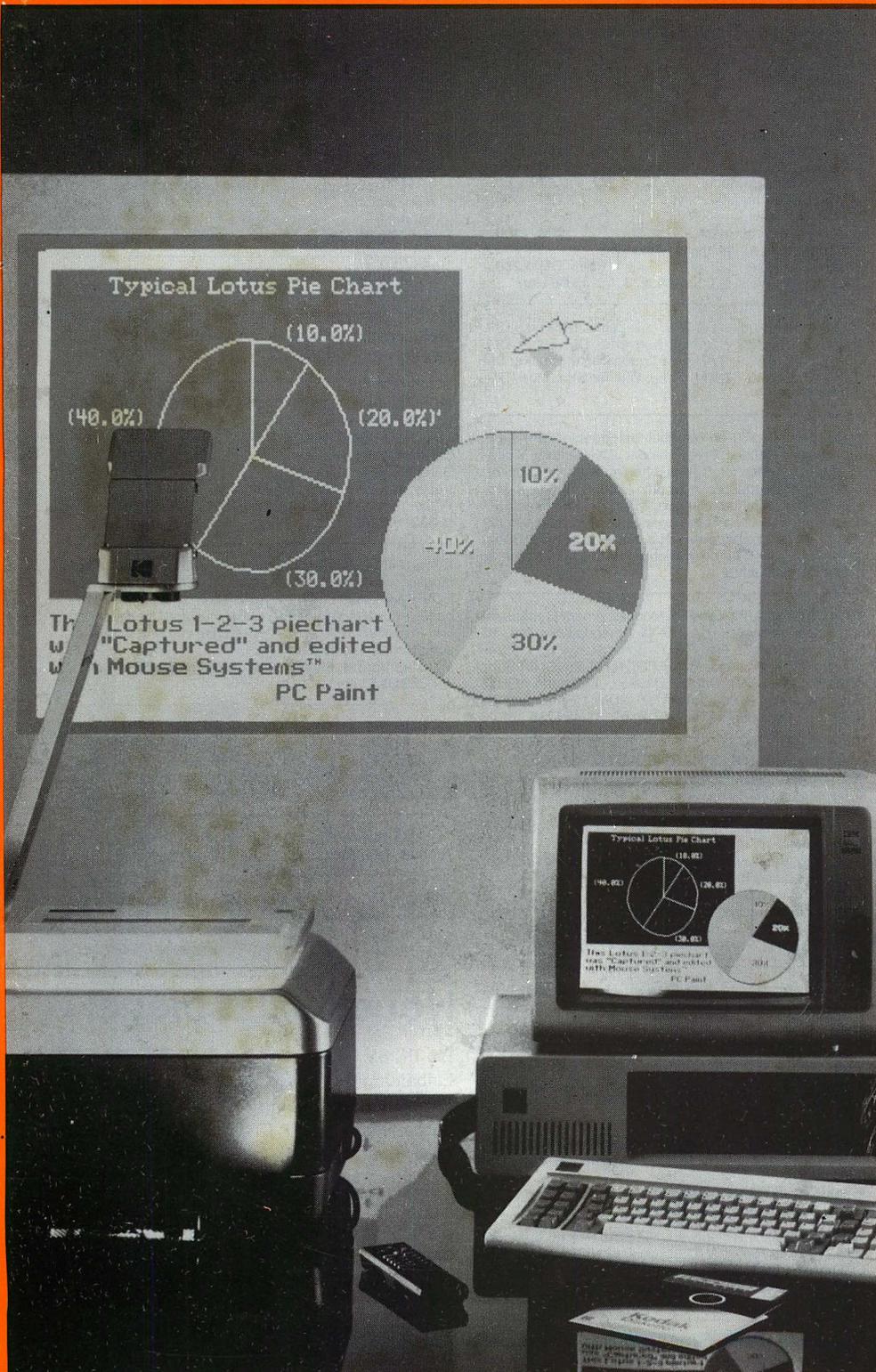


PRISMA

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

1987 Nr. 6

D 2856 E



Ganz nach dem Motto "Alle wollen etwas sehen!" funktioniert das Kodak Datashow System. Damit wird es zum Kinderspiel, ein vollständiges Auditorium an den Aktionen auf einem PC-Monitor teilhaben zu lassen. Den praktischen Nutzeffekt dieser Einrichtung schildert ein Erfahrungsbericht in diesem Heft.

Clubnachrichten

Clubbörse
Reise nach USA

Grundlagen

Die Mikroprozessor-Story

MS-DOS

Kodak Datashow System
Befehle und Programme
Zenith Z-183 Laptop

Serie 70

Datum & Zeit
GEOSTAT
HP-71 Speed Up
TIMER
Copy Out
Programmlisting
Formatted Input
DCMLX
Kalender
Eigenwerte und Eigenvektoren

Serie 40

Druckerakkus erneuern
RAMBOX
Mathematische Genauigkeit
Barcodes des HP-41
Tabellen-Editor
Astrofotografie
Steinersatz
Dreiecksberechnungen
File Header
COMPLEX
Funktionswerte von Polynomen
Tief- und Hochpaßfilter

Verkaufe Revox Tuner B261 und Revox Verstärker B251. 1 1/2 Jahre alt, Top-Zustand, VB 3600,-DM oder einzeln je 1900,-DM
Helmut Schaefer, Tel. (06172) 79741

HP41: Kartenleser, 250,- DM, Cassettenlaufwerk 500,-, HP-IL-Modul 180,-DM, Paccscreen V2.0 700,-DM, HP71 32KMo 250,- DM, Steinhauer, Tel. 069/2653288 von 8-15 Uhr.

- HP 75C DM 1.500,-
- Disc-Laufwerk 9114A DM 800,-
- Monitor 9" DM 200,-
- Monitor 12" DM 200,-
- Video Interface 82163B DM 100,-
- Video Interface 80 Zeichen DM 200,-
- Grabau Video Controller DM 400,-
- Text Formatter-Modul DM 100,-
- VisiCalc-Modul DM 200,-
- 8 K-Erweiterung DM 100,-
- IL-Interface für EPSON-Drucker DM 150,-
Alle Teile wie neu, Preise VB, Paketpreis - 20%.
Hugo J. Bugl
Petersweiher 51, 6300 Gießen, ☎ 0641/45414

Zu verkaufen: **HP-IL Digitalcassette-Drive** (HP82161A) original verpackt mit fünf Kassetten, für DM 600,-
Werner Meschede, Sorpestr. 4, 5788 Siedlingshausen

HP-75 Software Modul: Text Formatter für 130,- DM; HP-75 Programmsammlungen: Statistics und I/O Utilities für je 40,- DM und integrierbarer Thermodrucker für **HP-150 A/B** mit 6 Rollen schwarzem Papier für 550,- DM zu verkaufen. Toni Lerchenfeld, (CCD 1209)
☎ 08323/7323 oder 0043/5222 86669.

Verkaufe: Barcodeleser 180,- DM, 6-fach Netzgerät 150,- DM, Statik-, Stahl-, Massiv- und Holzbau- sowie weitere Programme für HP 97/41 D.I.T. Weckmann, ☎ 0234/471761 o. 04742/658

Verkaufe für HP41: Plotter-Modul, Math I-Modul, Akku 82033 A (neu) Michael Bering, Gartenstraße 9, 6929 Angelbachtal, ☎ 07265/7392

Verkaufe für HP-41: Synthetische Programmierung (Wickes) 15,- DM, Calculator Tips & Toutines (Dearing) 20,- DM; HP-41 Ic-Specifications (PFC) 18,- DM; Info gegen Freiumschiag
Suche: HP-IL Interface Specifications (90017) und The HP-IL Integrated Circuit (90016) als Original oder Kopie. Dirk Schüssler, Hüserstr. 4, 5760 Arnsberg 2

Schaltkreisanalyse-ROM f. 120,-DM; IDS I+II f. 160,-DM
Stephan Trzeciak, ☎ 04721/51688

Verkaufe wegen Systemumstellung: HP71B incl. IL-Modul DM 1.000,-, HP-82161 A Kassettenlaufwerk incl. 10 Kassetten DM 750,-, HP-82162 A IL-Thermodrucker DM 550,-, HP 2225 B IL-Thinkjet (neuwertig) DM 950,-
Volker Lang (130), ☎ 07138-5349 ab 18 Uhr.

Verkaufe (da doppelt) HP 71 B 1 1/2 Jahre 1A Zustand mit **HP-IL-Modul 82401, Mathe-Rom 5061-7226, Akkus** mit original **Zubehör** (Tasche, Schablone, engl. Handbücher und Kurzanleitungen) zusammen (1350,- DM) oder auch einzeln. Ulrich Bunse, 4630 Bochum,
☎ 0234/705052

Verkaufe folgende Prisma-Hefte, Jahrgang 1984 komplett Jahrgang 1985 1-3 und 5-8, Jahrgang 1986 komplett, Jahrgang 1987 1-5, Preis pro Heft 4,00 DM, Christoph Strüder, Schmittstr. 23, 5358 Bad Münstereifel - Soller

Verkaufe Grabau-Video-Interface GR7 IL (800,- DM)
Rainer Maier 3H71, ☎ 0511/517702

Original **UCSD-Pascal** für HP86/87 nur 250,-DM, ISFO-Diassembler auf Anfrage. I. Seelmann, Horstthof 12, 4250 Boltrop 2, ☎ 02045/82524

Wer hat Kenntnisse, mit mir im nächsten Frühjahr im westlichen Ruhrgebiet eine interessante **Programmierung von Robotern** für eine Montageaufgabe durchzuführen? Wer kann mir hierfür mit Informationen auch über die Vernetzung mit PC's weiterhelfen? Lutz Lebach, Botzelaer 34, B-4680 Gemmenich, Belgien

Wegen Systemwechsel gegen Gebot: **HP 75 C**, Epson FX 80 + IL-Schnittstelle eingebaut, IL-Kassettenlaufwerk, IL-Video-Interface. Hans Krissler, Haldestraße 7, 7901 Lonsee, ☎ 07336/5225

Suche X-Memory Module für HP 41
Mathias Seltam, ☎ 06151/316205

75C + 8K Erweiterung, Visicalc, i/o, text-formatter, Graphic-Solutions, 6 Card-Holder mit div. Magnetkarten, nur zusammen: DM 2000,-, CT-Ladegerät mit 3 Anschl. DM 150,-, Lucke, ☎ 040/401241, mo-fr. 9-17 Uhr

Biete an: **U-Logger W41** mit IL-Schnittstelle, Festpreis 1300,- DM wie neu! ☎ ab 19 Uhr 08104/7396

Verkaufe: Osborne I DQD, 4 Laufwerke 2 x 180 KB, 2 x 800 KB (extern), 1 Druckerkabel, 1 Kabel zur Tastaturverlängerung, Standardsoftware, Megadisk-CP/M, Handbücher, VHB 2400,- DM. Jürgen Staiger, 6800 Mannheim, ☎ 0621/106135 ab 19,00 Uhr.

Suche: IL-Drucker 82 162, Port-Extender
☎ 06181/20481 bis 19 Uhr

Defekter HP41CV zu verkaufen (Einschalttaste gebrochen) Preis Verhandlungssache CCD 2209, P.Weis ☎ 030/4653874 (ab 17.00 Uhr)

Zubehör für HP41CV wg. Systemwechsel zu verkaufen:
Kartenleser DM 250,-
Barcode-Leser DM 200,-
Time-Modul DM 100,-
XT-Modul DM 100,-
CCD-Modul DM 200,-
Mathe-Modul DM 50,-
Statistik-Modul DM 50,-
Finanz-Modul DM 50,-
Maschine Design-Modul DM 50,-
ca. 100 CCD-Magnetkarten DM 100,-
Prisma alle Hefte
C. Wickes, Best of Prisma DM 100,-
Alle Zubehörteile mit Original-Anleitung
Gesamtabnahme Preis DM 1.200,-
CCD 2209 P. Weis, ☎ 030/4653874, (ab 17.00 Uhr)

Für HP41: IL-Drucker DM 500,-, IL-Laufw. DM 650,-DM, IL Modul DM 150,- DM, IL-Konverter DM 200,- DM, I/O-Modul DM 50,-, 2 Akkus + Ladegerät für Drucker DM 100,-, Prisma 82-87 DM 150,-, Bücher Stck. 20,- DM, ☎ 0202/702618

Verkaufe wegen 70-er-Wechsel: Umgebauter HP-41-CX mit Akku und Netzteil, Port-Extender, 2X-Memory, Video-Interface, HP-Monitor, IL-Kassettenlaufwerk, IL-Drucker, IL-Modul, Barcode-Leser, Ersatz-Akku, 9 Software-Module, 7 Programmsammlungen, 7 Bücher, 11 original HP-Kassetten, 14 Rollen Thermopapier, 2 Modultaschen, 50 blanko Tastaturschablonen. Nur komplett gegen Gebot unter ☎ 07265/7392 oder Michael Bering, Gartenstr. 9, D-6929 Angelbachtal

HP-41 STANDARD/GAMES Modul in einem Gehäuse!!! m. Handbüchern, 65 DM; HP-41 Barcodeleser Zub. 10 DM; TI 58/59 Buch (kein 08/15 Schmöker, etwas für Insider), 20 DM; Anwenderprg. EPSON HX-20 (CHIP Sonderheft), 14,- DM; Daten, Dateien, Disketten (CHIP Buch, nagelneu u. ungelassen), 20,- DM; CHIP Jahrg. 86 kompl., 35 DM; alle Preise VHB, ☎ 07131/484738

Gesucht: „USER'S LIBRARY SOLUTIONS“-book.
● Surveying (00041-90141)
● Calendars (00041-90145)
● Cardiac/Pulmonary (00041-90097)
● Chemistry (00041-90102)
● Optometry I (General) (00041-90143)
● Heating, Ventilating and Air Conditioning (00041-90140)
● Test Statistics (00041-90082)
● Real Estate (00041-90136)
● Home Construction Estimating (00041-90096)
● Business Statistics/Marketing/Sales (00041-90094)
Wolfgang Knell, Prozeptionsweg 29, 4720 Bochum

Verkaufe: Advanced Pac Screen V1.1A für 650,- DM inklusive Handbuch und Kurzanleitung.
☎ 0228/379326 abends

Verkaufe: IL-Cassettenlaufwerk HP-82161A (10 Betriebsstunden) mit 2 Cassetten für 750,-DM.
Wolfram Winter, Wuppertal, ☎ 0202/741825

Verkaufe: 2 Stück, Externes Ladegerät mit Akku (HP-82037) passend für 82161/82162/HP-97 à 80,- DM NEU!! (wegen Systemaufgabe)
Peter Behrend, ☎ 0421/505117

Verkaufe: Kassettenlaufwerk HP 82161A incl. IL-Modul HP 82160A für 750,-DM, Drucker HP 82143A für 350,-DM CCD-Modul für 150,-DM,
Otto Lütkemeyer, ☎ 02161/41598

Verkaufe HP Thermodrucker 82143A plus Thermopapier 82175A plus Ladegerät für 400,-VHS.
W. Müller, ☎ 0221/402355

Verkaufe: schweren Herzens (wg. akutem Geldmangel) meinen 41er: Thermodrucker mit Graphikumbau 400,-DM Kartenleser (16) 300,- DM; Mathe I 50,- DM; Doppelmodule: QM/XF 220,- DM; CCD/XM 320,- DM - alles in bestem Zustand.
Thomas Bauer, ☎ 06151/63798 (öfter probieren)

Verkaufe: CT4100 Portextender (7 Ports, Batteriegepuffert) 110,- DM, Single mm 30,- DM, Akku für HP41 C+ (lange unbenutzt) 30,-
Suche: Schneider CPC-User zwecks Erfahrungs- und Programmaustausch. HP IL/Centronics Schnittstelle oder Bauanleitung einer solchen.
G. Radons, Kolbenzeil 18, 6900 Heidelberg

Verkaufe Kassettenlaufwerk HP-82161 A. Kaum gebraucht, völlig in Ordnung für nur 550,- DM (incl. 5 Kassetten). Jochen Haas, Leverkusen 3, ☎ 02171/46802

Neues Modul für den HP-71? Gibt es nicht!
Wir bieten Ihnen jedoch neue Software für den HP-71 im Umfang eines Modules an. Sie erhalten Programme zur Kurvenanpassung mit anschließender Graphik sowie vielen Utilities für nur 200,- DM (war sehr viel Arbeit). Info, Disketten bei Ulrich Schifferings, Leverkusen 1,
☎ 02173/41387

Zu verkaufen:
- Detaillierte Anleitung zum Bau von Doppelmodulen (6 Seiten DIN A 4) für 5,- DM.
- 15 Rollen schwarz- und 6 Rollen bladruckendes Thermopapier für 30,- DM plus Portokosten per Nachfrage
Stefan Fegert, (1937), Marxstr. 35 4230 Hattingen

HP41-Prgrms gesucht für Heizkostenabrechnung
Verkaufe **David Assmbl.** DM 100,-; ☎ 05123/8600

Wer kann im Münchner Raum beim HP-Thinkjet Hilfestellung leisten? ☎ 08122/14571

Zu verkaufen: DIGITAL-CASSETTE-DRIVE HP 82161 A auf 6 HP-Kassetten, DM 450,-
TECHNICAL AND ENGINEERING DICTIONARY Band 1 Deutsch-Englisch
Band 2 Englisch-Deutsch
Verlag McGraw-Hill Brandsteiner 1972 DM 80,-
Walter Schneider, Mommensenstraße 3, 5000 Köln 41,
☎ 0221/431898

Klima-Fachingenieur/in gesucht ab sofort oder baldmöglichst, mit Berufserfahrung oder auch Jung-Ingenieur. EDV-Kenntnisse und Bereitschaft zum selbständigen Arbeiten erforderlich für Planung, Beratung und Überwachung von Projekten im In- und Ausland. Bewerbung mit handschriftlichem Lebenslauf.
Ingenieurbüro R. Siegmund, Schlesienring 30 b, D-6368 Bad Vilbel 2

Verkaufe HP 41 CX + Magnetkartenleser + Barcodeleser + Akku + Ladegerät + 2 X-Memory-Module + Standart-Modul + Mathematik-Modul + 180 Magnetkarten + Literatur (Jarett, Dearing) nur komplett für 1200,-
Rüdiger Normann, ☎ 0541/387592

HP-71 B DM 750,- und HP-IL Modul DM 200,-
Thomas Kern, Lindenstr. 17, 5419 Großmairischeld

Verkaufe: HP-41 Kartenleser 250,- DM, Barcode-Lese-stift 150,- DM, X-Memory Modul 90,- DM, HP 62162 A IL-Thermodrucker 420,- DM, Frank Dickmann, Baumgartenweg 17, 6301 Pohlheim-Holzheim, ☎ 06004/2924

Neuer Satzspiegel

Die PRISMA-Redaktion bittet die Autoren um Einhaltung des neuen Satzspiegels für handgezeichnete Formeln, Abbildungen, Skizzen oder Zeichnungen. Dadurch wird viel Zeit und Arbeit gespart - und selbstverständlich auch der Platz im PRISMA besser genutzt.

Zwei Spaltenmaße stehen zur Auswahl: entweder 57 mm oder 87,5 mm. Aber bitte immer nur für ein Maß entscheiden.

Bei ganzseitigen Abbildungen gelten als maximale Abmessungen: 180 x 260 mm (Breite x Höhe).

CCD - quo vadis?*

* lateinisch: "wohin gehst du?"

Nimmt man diese Frage wörtlich, ja dann gibt es nur eine einzige Antwort: "... nach Corvallis in Oregon (USA)!" Dort findet nämlich im August 1988 die nächste Hand-Held-Conference statt. Der CCD will seinen Mitgliedern und den Mitgliedern anderer europäischer Computerclubs eine günstige - jedoch nicht billige - Reise nach USA anbieten, um neben anderen Dingen an eben dieser Konferenz teilzunehmen. Näheres dazu ist auf Seite 5 in diesem Heft nachzulesen.

Weniger wörtlich bezieht sich die einleitend gestellte Frage jedoch auf Inhalte und Aktivitäten innerhalb unseres Computerclubs. Da ist zunächst einmal ab 1988 eine geänderte und verbesserte Erscheinungsweise unserer Clubzeitschrift PRISMA herauszustellen. Die wenig befriedigende Unregelmäßigkeit in der letzten Zeit ist weder für die Redaktion tragbar und schon gar nicht ist sie unseren Lesern zuzumuten. Ab sofort erscheint PRISMA in zweimonatlicher Regelmäßigkeit jeweils in der Mitte eines geraden Monats. Das hat endlich den

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

Redaktion:
Hans Jürgen Hübner
Klaus Kaiser
Martin Meyer
Henry Schimmer
Dieter Wolf

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 ausschnittsweise - nur mit schriftlicher Genehmigung des CCD. Eine irgendwie geartete Gewährleistung kann nicht übernommen werden.

Vorteil zur Folge, daß wir einen klaren Redaktionsschluß definieren können - nämlich jeweils den letzten Tag eines ungeraden Monats. Das gilt selbstverständlich für alle Artikel, die Clubbörse und auch für kommerzielle Anzeigen. Dieser Schritt ist einfach notwendig, um die deutlich verbesserte Qualität von PRISMA mit einem regelmäßigen Erscheinen zu verbinden. Nach sorgfältiger Überlegung ist diese Entscheidung von Redaktion und Vorstand gemeinsam getroffen worden. Dabei war und ist immer zu berücksichtigen, daß alle redaktionellen Aktivitäten und Aufgaben ehrenamtlich und nach Feierabend geleistet werden.

Ein neuer Inhalt im PRISMA ist bereits in diesem Heft mit der neuen Rubrik "Grundlagen" eingeführt. Für unseren Club, der sich intensiv mit Computern und allem was dazugehört befaßt, stellt diese Rubrik eine weitere Bereicherung dar. In ihr werden in loser Reihenfolge - d.h. wenn wir entsprechendes Material eingesandt bekommen - Artikel erscheinen, die sich mit den Grundlagen der Computerei auseinandersetzen. Clubmitglied Dr. Ralf Kern aus Karlsruhe macht in dieser PRISMA-Ausgabe mit seiner "Mikroprozessor-Story" den Anfang auf Seite 7.

Alte Erinnerungen werden wach beim Lesen der Abkürzung UPLE (User's Program Library Europe), dem "Markenzeichen" der Europäischen Anwender Programmbibliothek von Hewlett-Packard. Nachdem der deutschsprachige Teil nun schon seit längerem dem CCD übergeben wurde, befinden sich die Programme zur Zeit in der Sichtung. Es sind nämlich weitaus mehr Programme, als im letzten von der UPLE erhältlichen Katalog verzeichnet waren. Nachdem alle Programme erfaßt und danach ausgewertet sein werden, bereichern auch diese das Angebot, daß der CCD seinen HP-41 Anwendern bieten kann. Eine interessante Perspektive für das Jahr 1988.

Ein weiteres Hurra - endlich ist SIE da! Die Regionalgruppe München ist von aktiven CCD-Mitgliedern in's Leben gerufen worden. Man will sich regelmäßig treffen und sowohl Clubmitgliedern als auch Nicht-Mitgliedern Unterstützung bei Hard- und Softwareproblemen bieten. Der Ansprechpartner ist Victor Lecoq, dessen Anschrift natürlich sofort in die Rubrik "Wichtige Clubadressen" aufgenommen wurde.

CCD - quo vadis?
Eine positive Entwicklung!

Frohe Weihnachten und ein
gutes Neues Jahr

Alf-Norman Tietze

Nachsendedienst

Wegen Familien-Nachwuchses - ein überaus freudiges Ereignis - muß Claudia Kiefer ihre Tätigkeit für den PRISMA-Nachsendedienst leider aufgeben. Wir bedanken uns bei Dir - liebe Claudia - für Deine Mitarbeit und wünschen Dir und Deiner Familie eine gute Zukunft.

Die ab sofort gültige neue Anschrift lautet:

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

Inhalt

Clubnachrichten	
Clubbörse	2
CCD - quo vadis?	3
Leserbrief	4
Hand-Held-Conference und USA-Reise 1988	5
Regionalgruppen	6
Grundlagen	
Die Mikroprozessor-Story	7
MS-DOS	
Kodak Datashow System	13
Befehle & Programme	15
Erfahrungsbericht Zenith Z-183	17
Serie 70	
Datum & Zeit setzen	18
GEOSTAT	19
HP-71 Speed Up	20
TIMER	21
Copy Out	21
Bug im LIST41	22
PRGMLIST	24
Formatted Input	26
... und noch ein LEX-File: DCMLX	28
Kalender	30
Eigenwerte und Eigenvektoren	32
Serie 40	
Druckerakkus erneuern	33
HP-41? - HP-41!	34
Betriebssystemerweiterung für die RAMBOX	34
Mathematische Genauigkeit	35
Barcodes des HP-41	38
Tabellen-Editor	41
Astrofotografie: Schwarzschildeffekt	43
Erläuterungen zu XSPLINE	44
Anmerkungen zu TR3 und TR4	45
Bauingenieurwesen: Steinersatz	46
Der Druckfehlerteufel	49
Dreiecksberechnungen	50
File Header	54
COMPLEX	55
Korrektur zu XSPLINE	56
Funktionswerte von Polynomen	61
Tief- und Hochpaßfilter	62
Barcodes	65
Clubadressen	70

Offener Brief

Liebe Freunde,
in letzter Zeit wurde viel über die Zukunft des CCD geschrieben. Diese Zukunft wird m.E. aber ganz erheblich von der Vergangenheit und Gegenwart bestimmt. Vielleicht sind meine Gedanken hierzu nicht ganz falsch:

Ein Club wie der CCD wird immer aus einer Minderheit bestehen, die Sachverstand hat und einer Mehrheit, die von diesem Sachverstand profitieren will und dafür Ihren Beitrag bezahlt. Dieser Beitrag nämlich ermöglicht es erst, daß der Club sich bemerkbar machen kann (s. Prisma, Messestände etc.). Es ist deshalb schädlich, diese Mehrheit anzuklagen oder zu vernachlässigen. Wer sich also (freiwillig) der Minderheit angeschlossen hat, der hat auch Verantwortung. Beispiele:

Wird in einem Prisma angekündigt, daß in der nächsten Ausgabe die finanz. Situation behandelt wird, so muß dies auch geschehen – oder es muß ein Grund angegeben werden, warum dies nicht geschah.

Wird über die Mitgliederversammlung berichtet, so darf es sich nicht um eine Meisterleistung handeln, mit wenigen Worten möglichst nichts zu sagen. Wird über ein geplantes Projekt berichtet, so muß auch über den Fortgang oder den Tod des Projektes etwas gesagt werden.

Hat jemand eine Aufgabe übernommen, so muß er dieser auch nachkommen (z.B. Clubadressen: Grabau-GR7: Wann hören die Mitglieder mal etwas über Möglichkeiten oder geplante Erweiterungen etc? Wenn es nichts zur berichten gibt, braucht man auch keine eigene „Institution“ hierfür),

Mitgliederschwund entsteht dadurch, daß Mitglieder enttäuscht sind; d.h. sie kommen zu der Ansicht, daß der Aufwand (Beitrag) nicht mehr dem Nutzen (Information) entspricht. Dies kann sogar der Fall sein, ob-

wohl viel „Information“ (viele und dicke Prisma) geboten wird, diese aber nur für jeweils wenige Mitglieder interessant ist (zu spezielle Programme). Notwendig sind m.E. Informationen, mit denen möglichst viele Mitglieder etwas anfangen können, das sind nützliche Routinen für die betreuten Rechner. So gibt es z.B. für die Rechner der Serie 80 Bin-Routinen, die auch für den 71er interessant wären (HGL\$, HMS(), HMSS(), MDJ(),MDJ\$()), XREFL, XREFV, MAT A-definierter Teil von MAT B (fehlt im 71 MAT-ROM), schnelle Sortierung eines \$-data-files usw.) Eine ganz kühne Vorstellung wäre z.B. eine Art Datenbank-Verwaltung. Vielleicht sollte man die Mitglieder einmal fragen, was sie an solchen „Hilfen“ interessiert.

Das alles geht natürlich nur, wenn es Leute gibt, die so etwas verwirklichen können. Gibt es diese nicht, so ist alles Jammern sinnlos, der CCD hätte keine Zukunft. Dies kann man auch nicht dadurch beheben, daß man neue Rechner einbezieht, deren Behandlung unter den gleichen Mankos leidet.

Wenn jemand jetzt glaubt, dies sei ein Schreiben gegen die Aktiven im CCD, so hat er leider nichts verstanden. Dieser Artikel soll die absolute Notwendigkeit der aktiven Spezialisten dokumentieren. Natürlich ist jedes Mitglied aufgefordert, nach Kräften etwas beizutragen. Erforderlich für die Cluberhaltung sind jedoch die „Rosinen-Produzenten“. Zum Glück haben wir einige davon, die sehr wach sind, und bei denen ich mich herzlichst bedanken möchte. Wenn einige andere, die vielleicht etwas schlummerten, aufgewacht sind, so haben diese Zeilen mehr erreicht, als ich zu hoffen wagte.

Ein anderer Punkt sind die Preise, die sicher vielen HP-Anhängern die Freude verderben. Daß ein HP-71 teurer ist, als ein Sharp 1600, das leuchtet mir ja gerade noch ein.

Völlig unklar ist mir jedoch, warum man ein 128 kB-RAM für den Sharp für DM 346.50 erhält, ein 128 kB-RAM für den 71B jedoch DM 1.859,- kostet. Da sollen wir doch wohl übers Ohr gehauen werden, RAM ist RAM – oder nicht? Bitte nicht den Einwand von der Stückzahl, bei dem Preis kommt der Verkauf natürlich auf keine Stückzahlen. Hier sollte der CCD versuchen, wirklich eine billigere Quelle zu finden (eventuell zusammen, mit den entsprechenden Clubs anderer Länder!).

Leider gibt HP selbst auch nur wenig Grund zur Freude. Auf der diesjährigen CEBIT fand ich am HP-Stand nicht einen einzigen loop-fähigen Taschenrechner (41, 71, 75). Das erklärt vieles. (Vielleicht sollte sich der CCD doch auf Sharp-Rechner umstellen!) Als man den HP-71 schuf konnte man sich schon eine Weiterentwicklung vorstellen, zwei Anzeigepositionen sind für spätere Anwendungen reserviert. Offensichtlich hat man bei HP leider vergessen, was man damit wollte. Bei einem solchen Marketing (s. CEBIT) verkauft man natürlich nichts, was wieder eine hervorragende Begründung abgibt, die Serien 40 und 70 langsam sterben zu lassen. Dabei gibt es noch Möglichkeiten. Ich z.B. würde ein 20 x 80 Zeichen LCD-Display begrüßen (etwa so groß und flach wie max. der HP-75), das es ermöglicht, Programme, die zu Hause über den Monitor laufen, auch unterwegs komfortabel zu benutzen (Display als sep. Gerät über die Loop anzusprechen). In dem Display-Gehäuse wären vielleicht noch Steckplätze für RAM-Bausteine unterzubringen, die einen Ausbau als RAM-Disc erlauben. Aber sicher kostet das dann wieder soviel, daß es sich niemand kauft. Was der CCD gegen die Unlust von HP tun kann, das weiß ich leider auch nicht.

Heinz Speckert
Am Graßgang 26
6370 Oberursel

Anmerkung der Redaktion

Clubmitglied Heinz Speckert hat völlig recht, wenn er der Meinung ist, daß angekündigte Themen auch wirklich in einer der Folge-nummern von PRISMA erscheinen sollten. Die PRISMA-Redaktion hat aus den "Pleiten" früherer Hefte jedoch gelernt und kündigt nichts mehr an, was nicht in die Tat umgesetzt werden kann.

Die Aufgabe von Holger v. Stillfried (Grabau GR7) wurde hier allerdings grundlegend mißverstanden. Die meisten unserer aktiven Clubmitglieder, die eine Aufgabe übernommen haben, sind hauptsächlich als Ansprechpartner innerhalb spezieller Fachgebiete oder bestimmter Problemkreise zu betrachten - und nicht als Artikelschreiber! Holger v. Stillfried ist gewissermaßen die "Schnittstelle" zwischen Anwender und der Firma Grabau, wenn es um Schwierigkeiten und Probleme bei der Anwendung des GR7 Interfaces geht.

Alf-Norman Tietze

Trainer für Computerkurse gesucht

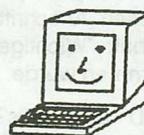
Gesucht werden Leute, die Erfahrung einerseits mit MS-DOS Rechnern haben, eventuell auch mit verschiedenen gängigen Softwarepaketen wie Lotus 1-2-3, Word und Tabellenkalkulation;

zusätzlich sollte man noch mindestens 4 Wochen Zeit haben, um diese Kenntnisse in den Ferienclubs Aldiana (NUR Touristik) an den Mann oder die Frau zu bringen.

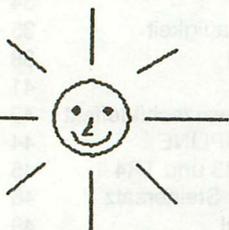
Dabei ist wegen der natürlich erwachsenen Schüler ein Alter von mindestens 22 Jahren Voraussetzung, pädagogische Fähigkeiten sollte man aber auch mitbringen, Hacker sind da wohl fehl am Platze.

Geboten werden wie schon erwähnt mindestens 4 Wochen frei Kost und Logis in einem der Ferienclubs von „Club Aldiana“, freien Hin- und Rückflug sowie ein, so beschrieben, leistungsbezogenes Entgelt.

Alle, die jetzt noch nicht abgeschreckt sind, können jetzt den Prospekt in der Redaktion bei mir anfordern, beachtet bitte die Altersgrenze!



CCD-Redaktion
Martin Meyer (1000)
Postfach 11 04 11
D-6000 Frankfurt/Main



Hand – Held – Conference 1988

Eine Clubreise in die USA

Auf der diesjährigen Hand-Held-Conference in Kopenhagen wurde unter anderem beschlossen, daß die nächste Konferenz im Sommer 1988 in den USA, in Corvallis/Oregon, mit europäischer Beteiligung stattfinden soll.

Verlockend an der Sache ist, daß die Konferenz von der Firma CMT Corvallis Microtechnology Inc., in Corvallis/Oregon durchgeführt wird und daß der CCD als der Partner auf dem Gebiet der Hand-Helds durch den Präsidenten der Gesellschaft – David Lin – offiziell eingeladen worden ist.

Nach einem Vorstandsbeschuß soll den Mitgliedern des CCD und anderer europäischer Clubs die Teilnahme im Rahmen einer großen europäischen Delegation an dieser Veranstaltung im August 1988 ermöglicht werden.

Der CCD kann allerdings nicht als Veranstalter fungieren. Deshalb wurde ein geeigneter Veranstalter eingeschaltet, der nachweislich Erfahrung in der Durchführung solcher Vorhaben für Verbände und anderer Organisationen hat. Dieser Veranstalter ist die Firma ZM Travel Service, Hartmann Reisen GmbH, 4300 Essen, die den Ingenieuren aus dem VBI unter unseren Mitgliedern als leistungsfähiges Unternehmen dieser Art bekannt sein dürfte, weil ZM unter anderem für den VBI Gruppenreisen zu internationalen Ingenieur-Kongressen in aller Welt seit Jahren veranstaltet.

ZM hat für den CCD verschiedene Alternativen dieser Clubreise vorgeplant. Der Reisevorschlag beruht auf der Idee, die Teilnahme an der Konferenz entweder im Rahmen eines einwöchigen Kurztrips durchzuführen oder in einer zweiwöchigen Kanada/USA-Reise einzubinden, damit die Teilnehmer auch die Möglichkeit haben, neben dem Thema Computer in einer der schönsten Landschaften der Welt einige Tage zu entspannen; aber auch die Gelegenheit zu haben, im „Mutterland“ der Hand-Helds und PC's Informationen aus

erster Hand zu bekommen und die besten Einkaufsmöglichkeiten für neue Geräte und Zubehör wahrnehmen zu können. Dieses Vorhaben ist somit nicht nur für unsere 41er- und 71er-Anwender interessant, sondern auch für die CP/M- und MS-DOS-Gruppe; somit für jedes Club-Mitglied!

Die Konferenz findet vom 4. bis 6. August 1988 in Corvallis statt. Die Anreise dazu soll einige Tage vorher stattfinden und erfolgt für alle Teilnehmer am Sonntag 31.07.88 von Europa nach Vancouver in Kanada. Dort werden zwei Tage Aufenthalt sein, so daß alle Teilnehmer rechtzeitig zur Konferenz am 03.08.88 in Corvallis eintreffen. Die Konferenz beginnt am Donnerstag, 04.08. morgens und endet am Samstag, 06.08. abends. Am Sonntag 07.08.88 reisen alle Teilnehmer nach San Francisco, wo sich dann die Gruppe am Dienstag 09.08.88, teilt; denn die Teilnehmer des Kurztrips – Reisevorschlag A – fliegen dann zurück nach Europa und die Teilnehmer am touristischen Programm – Reisevorschlag B – fahren von da über den High-Way Number One auf einer der schönsten Küstenstraßen entlang der Pazifikküste mit Stops in Carmel und Monterey nach Santa Barbara, um dort noch einige Tage am Pazifik-Strand zu verbringen, und weiter nach Los Angeles von wo die Rückreise am Sonntag, 14.08.88 beginnt, so daß diese Gruppe am Montag, 15.08.88, wieder in Europa ist.

Die Unterbringung erfolgt für alle Teilnehmer im Doppelzimmer mit Bad/Dusche und WC ohne Verpflegung (gefrühstückt wird für ein paar Dollars im Coffee-Shop nebenan und das ganz großartig!) für die Gruppe B in Mittel-Klasse-Hotels inclusive Frühstück.

Neben der Konferenz und den touristischen Attraktionen werden noch Besuche bei interessanten Computerfirmen vorgesehen. Daß die Firmen HP und CMT in Corvallis besucht werden, steht bereits heute schon fest. Für Begleitpersonen und Reisetilnehmer, die nicht an der Hand-Held-Conference teilnehmen, wird ein attraktives Rahmenprogramm geboten, wenn diese Tage nicht zur persönlichen Verfügung stehen sollen.

Die Kosten betragen einschließlich Linienflug mit der Deutschen Lufthansa für den Reisevorschlag A DM 3.150,00 und für den Vorschlag B DM 4.895,00 jeweils pro Person. Der Einzelzimmerzuschlag liegt bei ca. DM 600,00 für die Preisgruppe A und bei DM 1.000,00 für B, da es in Amerika keine Einzelzimmer als solche gibt. Dafür werden immer Doppelzimmer zur einzelnen Benutzung überlassen.

Die vorstehenden Reisepreise basieren auf einer Mindestteilnehmerzahl von 25 Personen. Es wurde zum zur Zeit gültigen Umrechnungskurs errechnet. Preisänderungen nach oben oder unten können sich aus Kursschwankungen und natürlich aus der tatsächlichen Teilnehmerzahl ergeben. Bei einer Gruppe von beispielsweise 50 bis 100 Teilnehmern sieht die Kalkulation wesentlich günstiger aus.

Ab- und Ankunftsflughäfen werden in Europa je nach Zusammensetzung der Reisegruppe Amsterdam, Düsseldorf oder Frankfurt sein. Es ist geplant, den übrigen europäischen Clubs, die in Kopenhagen beteiligt waren, die Teilnahme an unserer Veranstaltung anzubieten. In diesem Fall wird als Abflug- und Ankunftsbasis Amsterdam gewählt, wenn aus diesen Clubs Teilnehmer zu berücksichtigen sind, anderenfalls Düsseldorf oder Frankfurt. Gegen geringe Mehrkosten, die fast die Parkgebühren für 14 Tage an den Flughäfen ausmachen, können Zubringerflüge gebucht werden.

Es sollten Viele von dieser interessanten und sicherlich auch einmaligen Gelegenheit einer Studienreise zum Thema Computer in die USA, die zudem auch noch steuerlich geltend gemacht werden kann, Gebrauch machen und mit dem anhängenden Coupon vom Veranstalter ein Angebot anfordern und sich dann anmelden; denn Anmelde-schluß ist der 1. März 1988!

Erich H. Klee
Ruhrallee 8
4300 Essen 1

ZM-Travel Service
– Frau Karin Wolniczak –
Hartmann Reisen GmbH
Rolandstraße 9

4300 Essen 1

Absender:

Name

Vorname

Straße

PLZ/Ort

Telefon

Hardware · Software
Servicestation
Beratung · Zubehör

OSBORNE
Management by Computer.

WORDLORD · Textverarbeitung · CAD-Anwendungen · Komplettsysteme

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

Branchenlösung für Klein- u. Mittelbetriebe

Postfach 1220 · 4133 Neukirchen-Vluyn

Telefon 0 28 45 / 3 22 94

 **Sonderpreise für CCD-Mitglieder**

Regionalgruppe Karlsruhe

In Karlsruhe treffen sich CCD-Mitglieder regelmäßig an jedem zweiten Mittwoch eines Monats. Die Termine des ersten Quartals 1988 liegen wie folgt:

13.01.88 / 10.02.88 / 09.03.88

Regionalgruppe Rhein-Main

Auch im Rhein-Main-Gebiet haben CCD-Mitglieder die Möglichkeit regelmäßig Erfahrungsaustausch und persönliche Kontakte zu pflegen. Der Treffpunkt ist jeden letzten Samstag eines Monats um 15:00 Uhr in Raunheim (bei Rüssels-

heim, nahe Rhein-Main-Flughafen). Die Termine für 1988 liegen wie folgt:

jeweils um 15:00 Uhr

30.01.88 / 27.02.88 / 26.03.88

30.04.88 / 28.05.88 / 25.06.88

30.07.88 / 27.08.88 / 24.09.88

29.10.88 / 26.11.88

Dezember entfällt

Ich bitte um ein Angebot für die Clubreise des CCD zur

Hand-Held-Conference '88
vom 31. Juli bis 14. August 1988

Neben mir wird/werden noch Begleitperson/en teilnehmen. Mir ist bekannt, daß der vorläufige Reisepreis DM 3.150,00 für den Reisevorschlag A und DM 4.895,00 für den Reisevorschlag B jeweils pro Person beträgt und daß mir der verbindliche Preis mit dem Angebot des Veranstalters mitgeteilt wird.

Ort/Datum

Unterschrift

Die Mikroprozessor-Story

Vom 4-Bit-Prozessor zum Supermicro

Vorwort

Prisma soll inhaltlich ein neues Gesicht bekommen, und dazu soll ihm auch der folgende Beitrag verhelfen. Er ist Ende 1985 in der Zeitschrift „micro“ erschienen und daher in manchen Punkten nicht mehr ganz aktuell (was jedoch keine Rückschlüsse auf die neue Linie bei Prisma erlaubt!), aber vor allem die historischen Betrachtungen machen ihn auch jetzt noch lesenswert. Darüber hinaus ist es im nachhinein ganz interessant zu verfolgen, was aus den damaligen Erwartungen und Prognosen geworden ist, welche einstmal lautstark angekündigten Typen schon in Vergessenheit geraten und welche derzeit aktuellen Entwicklungen man damals noch gar nicht geahnt hat. Ich habe vor, den Artikel in einem Nachtrag auf den heutigen Stand der Mikroprozesstechnik nachzuführen. Erst aber einmal zurück ins Jahr 1985:

Die ersten 32-Bit-Micros sind auf dem Markt und leisten spielend so viel wie Minicomputer oder gar Großrechner, wie die Hersteller vollmundig versprechen. Dabei gibt es Mikroprozessoren erst seit knapp 15 Jahren. Der folgende Artikel informiert darüber, wie es mit den Mikroprozessoren begonnen hat und welchen Lauf ihre Entwicklung nahm.

Die Mikroprozessor-Revolution begann im Jahr 1971, als der Halbleiterhersteller Intel den ursprünglich im Auftrag der japanischen Firma Busicom entwickelten Mikroprozessor 4004 auf den Markt brachte und damit eine kaum noch überschaubare technische Entwicklung einleitete. Was war eigentlich die revolutionäre Neuerung an diesem Konzept?

Vor Erfindung des Mikroprozessors mußte in der Digitaltechnik praktisch für jeden neuen Anwendungsfall eine eigene integrierte Schaltung entwickelt werden – eine zeitraubende und kostenintensive Angelegenheit. Mit dem Mikroprozessor wurde es möglich, denselben Chip in grundverschiedenen Funktionen einzusetzen – man braucht ihn nur anders zu programmieren. Das bringt hohe Stückzahlen, damit niedrige Bauteilpreise; vor allem aber kann man für eine neue Anwendung die nötigen Mikroprozessor-Programme mit einem Bruchteil des Zeit- und Kostenaufwands entwickeln, der zum Entwurf einer integrierten Schaltung vergleichbaren Funktionsumfangs nötig wäre.

Der 4004 war noch ein 4-Bit-Prozessor mit einem 4-Bit-Addierer, 16 4-Bit-Registern und einem Stack auf dem Chip. Er konnte 46 Befehle ausführen und 4096 Bytes adressieren – eine im Vergleich zu modernen Prozessoren fast lächerlich niedrige Leistungsfähigkeit. Doch schon Anfang 1972 schob Intel mit dem 8008 den ersten 8-Bit-Mikroprozessor nach.

Die Konkurrenz schloß natürlich nicht, sondern erkannte auch bald die Möglichkeiten, die in den Mikroprozessoren stecken, und so kamen andere Halbleiterhersteller mit eigenen Mikroprozessor-Chips auf den Markt. Hier seien nur der 8-Bit-Typ 6800 von Mo-

torola und die TMS-1000-Familie von Texas Instruments genannt, die ebenso wie Intels 8080 1974 eingeführt wurden. In Tabelle 1 sind einige weitere Meilensteine der Mikroprozessorentwicklung bis heute aufgeführt.

4-Bit-Prozessoren – noch nicht passe

Interessant dabei ist, daß die TMS-1000-Familie, mit der Texas Instruments zum führenden Hersteller von 4-Bit-Prozessoren wurde, erst im Jahr 1980 mit 44 Millionen Stück ihr Produktionsmaximum erreichte. Zum Vergleich: Von der 8086-Familie von Intel, die derzeit bei modernen Personal Computern den weitaus größten Marktanteil hält, wurden 1983 nur gut 6 Millionen Stück abgesetzt. Das beweist, daß auch 4-Bit-Prozessoren noch längst nicht tot sind – bei Stückpreisen von unter 1 US-\$ eignen sie sich immer noch vorzüglich zum Einsatz in Taschenrechnern, Videospiele, Haushaltsgeräten und einfachen Steuerungen. Sie brauchen dabei die Konkurrenz der besseren, aber auch teureren Typen nicht zu fürchten.

Als etwas „exotische“ Besonderheit sei in diesem Zusammenhang auch der 1-Bit-Prozessor 14500 von Motorola mit einem Satz von 16 Befehlen erwähnt, der für einfachste Steuerungsaufgaben (z.B. An- und Ausschalten einer Heizung) eingesetzt werden kann.

Nicht nur die Wortbreite in Bit ist für den Einsatzbereich eines Prozessors entscheidend, sondern auch Kriterien wie z.B. Geschwindigkeit, Größe des Befehlssatzes, Adressierungsarten, benötigte (und angebotene) Peripherie-Chips und nicht zuletzt auch so „weltliche“ Dinge wie die Versorgungsspannung. Obwohl der 8080 hier einige Nachteile etwa gegenüber dem 6800 hatte (drei Spannungen gegen nur eine, weniger Auswahl an Peripherie-Chips), setzte er sich als ausgesprochener Industrie-Standard durch.

Die Entwicklung schritt rasant vorwärts – ebenfalls 1974 begann schon die 16-Bit-Ära mit dem PACE von National Semiconductor. Dennoch dominierten vorerst die 4- und 8-Bit-Typen, und in diesem Bereich tat sich auch noch einiges.

1976: Höhepunkt der 8-Bit-Entwicklung

Schon zwei Jahre später traten zwei neue erfolgreiche Konkurrenten auf den Plan. Ironie des Schicksals – beide Gesellschaften gingen auf abtrünnige Entwicklungsingenieure zurück. Federico Faggin sprang bei Intel ab, gründete Zilog, wo der Z 80 entwickelt wurde, der zum 8080 aufwärtskompatibel ist. William Mensch Jr., ehemals bei Motorola, baute MOS Technology (kurz Mostek, inzwischen von United Technologies geschluckt) auf, wo sich unter seiner Leitung der Entwurf des 6502 an der Architektur des 6800 von Motorola orientierte. Beide Mikroprozessoren, der Z 80 und der 6502, wurden zu den Spitzenreitern des 8-Bit-Marktes bei Personal Computern.

Während der 6502 vor allem von Apple, Atari und Commodore eingesetzt wurde, gewann der Z 80 seine Popularität auf einer Vielzahl von Computermarken vorwiegend in Zusammenarbeit mit dem Betriebssystem CP/M. Die enge Verbindung von CP/M mit dem Z 80 ist eigentlich paradox, denn dieses Betriebssystem wurde ursprünglich für den 8080 geschrieben, und dementsprechend macht es z.B. vom erweiterten Register- und Befehlssatz des Z 80 gar keinen Gebrauch. Auch die CP/M-Anwendungssoftware darf sich, wenn sie uneingeschränkt CP/M-kompatibel sein soll, genaugenommen nur auf die Möglichkeiten des 8080 stützen – und das trotz der Tatsache, daß es fast gar keine CP/M-Rechner mit dem 8080 gibt.

Trotz des Aufkommens modernerer 16-Bit-Prozessoren haben Z 80 und 6502 immer noch große Marktanteile (die Jahres-Pro-

Erscheinungsjahr	Typ	Hersteller	Wortbreite (Bit)
1971	4004	Intel	4
1972	8008	Intel	8
	PPS-4	Rockwell	4
1974	8080	Intel	8
	TMS 1000	Texas Instruments	4
	6800	Motorola	8
1974	PACE	National Semicond.	16
1976	6502	Mostek	8
	Z 80	Zilog	8
1978	TMS 9900	Texas Instruments	16
	8086	Intel	16
1979	Z 8000	Zilog	16
1980	MC68000	Motorola	16
1981	iAPX 432	Intel	32
1982	WE 32000	AT&T	32
	o. Namen	Hewlett Packard	32
1984	80286	Intel	16
	NS 32016	National Semicond.	16
1985	NS 32032	National Semicond.	32
	68020	Motorola	32

Tabelle 1: Erscheinungsjahre einiger wichtiger Mikroprozessoren

duktionszahlen liegen noch bei über 10 Millionen Stück). Sie werden daher zu Recht als „Arbeitspferde“ des 8-Bit-Marktes bezeichnet. Ebenso wie die 4-Bit-Typen werden auch die 8-Bit-Mikroprozessoren noch längere Zeit nicht aus ihren Sockeln zu verdrängen sein, da sie sehr billig geworden sind und sich auf sie viele ausgereifte Programme und Schaltungen stützen.

Darüber hinaus muß man beim Einsatz älterer, scheinbar schon überholter Prozessoren beachten, daß diese oftmals mit recht hohen Arbeitsgeschwindigkeiten aufwarten können. Mit dem Ausreifen des Produktionsprozesses sind die Hersteller in der Lage, den Chip schrittweise zu verkleinern und dabei immer bessere Leistungsparameter hineinzupacken. Kleinere Abmessungen vermindern aber auch die Leitungskapazitäten innerhalb der Schaltung und erlauben daher größere Verarbeitungsgeschwindigkeiten. Damit kann ein ausgereifter älterer Typ unter Umständen mehr leisten als die erste Marktversion eines ganz neuen Prozessors. So kam z.B. der Z 80 mit einer Taktfrequenz von 1 MHz auf den Markt; inzwischen kann man ihn – zumindest in kleinen Stückzahlen – mit 10 MHz bekommen.

Allerdings sollte man nicht glauben, daß man in einem vorhandenen Computer einfach den Mikroprozessor gegen eine schnellere Version austauschen könnte. Ohne einen höherfrequenten Quarz im Taktgenerator läuft der neue Prozessor so langsam wie der Alte. Hat man auch den Quarz ersetzt, so bleibt doch noch der Speicher als entscheidender Hemmschuh bestehen, und gerade die Aufrüstung des Speichers mit schnelleren ICs kann ziemlich ins Geld laufen.

Z 80 und 6502 – zwei ungleiche Architekturen

Es lohnt sich, die Architektur (d.h. den internen Aufbau) von Z 80 und 6502 zu vergleichen, denn darin unterscheiden sie sich ganz erheblich. Der Z 80 beruht als Weiterentwicklung des 8080 auf dessen eher konventioneller Architektur (s. Abb. 1). Bei ihm ist zu den acht 8-Bit-Registern des 8080, von denen sechs zu drei 16-Bit-Registern

zusammengeschaltet werden können, ein kompletter zweiter, zum ersten identischer Registersatz hinzugefügt worden, dazu noch zwei 16-Bit-Index-Register. Hinzu kommen der Befehlszähler und der Kellerzeiger („stack pointer“). Darüber hinaus haben die Zilog-Ingenieure den Befehlsatz des 8080 erheblich erweitert und außerdem die Interrupt-Eigenschaften verbessert.

Ganz anders sehen die Dinge beim 6502 aus. Betrachtet man sein Registerschema (Abb. 2), so fällt auf, daß er über nur vier 8-Bit-Register, Kellerzeiger und den Befehlszähler verfügt. Dennoch darf man sich nicht täuschen und glauben, daß der 6502 daher weniger leistungsfähig wäre als der Z 80. Der Befehlsatz des 6502 erlaubt es mittels besonderer Adressierungsarten, alle vorhandenen Operationen auch mit den Bytes Nr. 0 bis 255 im RAM durchzuführen, also in der sogenannten Grundseite („zero page“). Man kann sich damit den 6502 so vorstellen, als hätte er 256 Register! Allerdings dauert der Zugriff auf ein solches „RAM-Register“ länger als auf ein internes CPU-Register.



Abbildung 2: Das Registermodell des 6502. Die Register X und Y dienen als Indexregister. Der Keller liegt immer zwischen hexadezimal 100 und 1FF, so daß ein 8-Bit-Zeiger ausreicht.

Der 6502 macht diesen Geschwindigkeitsverlust mit zwei Maßnahmen wieder wett: Zum einen hat er getrennte Adreß- und Datenleitungen, während der Z 80 dieselben

Leitungen im sogenannten „Multiplex-Verfahren“ zeitlich abwechselnd für den Austausch von Adreß- und Dateninformationen benutzt. Zum anderen aber benutzt der 6502 als einziger 8-Bit-Typ eine sehr fortschrittliche Methode, die sich in fast allen neueren 16- und 32-Bit-Prozessoren wiederfindet: das sogenannte Befehls-„Pipelining“ oder Fließband-Verfahren, bei dem während der Abarbeitung eines Befehls schon der nächste zur Verarbeitung vorbereitet wird. Daher gilt eifß 6502 mit einer Taktfrequenz von 1 MHz als ebenso schnell wie ein Z 80 mit 4 MHz.

Die Firmen Zilog und Mostek konnten ihren Ersterfolg nicht wiederholen. Commodore kaufte seinen Lieferanten Mostek auf, um keine Nachschubprobleme mit dem 6502 zu bekommen; inzwischen ist Mostek beim Konzern United Technologies gelandet. Daher konnte sich die 6502-Linie nicht so recht geradlinig weiterentwickeln. Zwar gibt es einige Firmen (Rockwell, Synertek, Western Design Center), die sozusagen als „Erben“ von Mostek Nachfolgemodelle vorstellten (z.B. R65C02, 65C802, 65C816, 6510), von denen aber keiner die Verbreitung des 6502 erreichen konnte, obwohl diese Typen sämtlich zum 6502 aufwärtskompatibel sind.

Zilog geriet zwar unter die Fittiche des ölmultis Exxon, konnte aber ebensowenig wie Mostek mit dem 16-Bit-Typ Z 8000 den Erfolg des Erstlings wiederholen. Wie beim 6502 gibt es auch hier Versuche verschiedener Halbleiter-Hersteller, Verbesserungen des Z 80 auf den Markt zu bringen, etwa den Z 800 von Zilog selbst mit vergrößertem Adreßraum, leistungsfähigerem Befehlsatz und höherer Geschwindigkeit, oder der NSC 800 von National Semiconductor, der denselben Befehlsatz wie der Z 80 hat, aber elektrisch mit dem 8085 kompatibel ist, so daß der ohne weitere Änderungen in 8085-Schaltungen eingesetzt werden kann. Besonderes Interesse verdient auch die Entwicklung des Mikroprozessors S83 von American Microsystems Inc. (AMI) in Zusammenarbeit mit Zilog und Digital Research, von woher CP/M stammt. Der S83 vereinigt das Betriebssystem CP/M mit den Prozessorfunktionen des Z 80 auf einer einzigen integrierten Schaltung.

Bei einer Aufstellung der 8-Bit-Prozessoren darf der 6809 von Motorola nicht fehlen. Er bildet das obere Ende der 6800-Familie und wird auch auf einigen Mikrocomputern wie dem Eurocom II oder dem Dragon eingesetzt. Er gilt als etwa viermal so schnell wie ein 6800 mit der gleichen Taktfrequenz. Er besitzt zwei 8-Bit-Akkumulatoren, die bei Bedarf zu einem 16-Bit-Akku zusammengesetzt werden können. (s. Abb. 3). Damit kann er 16-Bit-Arithmetik durchführen und kennt – im Gegensatz etwa zu Z 80 oder 6502 – sogar einen Maschinenbefehl für die Multiplikation. Mit Hilfe der zusätzlichen Speicherverwaltungseinheit 6829 kann er statt der normalen 64 KByte bis zu 2 MByte adressieren. Mit diesen Eigenschaften steht der 6809 dicht an der Grenze zur Leistungsfähigkeit der 16-Bit-Prozessoren, die ihm allerdings größte Konkurrenz machen. Er ist leider zu spät auf den Markt gekommen, um sich in der Lücke zwischen 8 und 16 Bit seinen Eigenschaften entsprechend behaupten zu können.

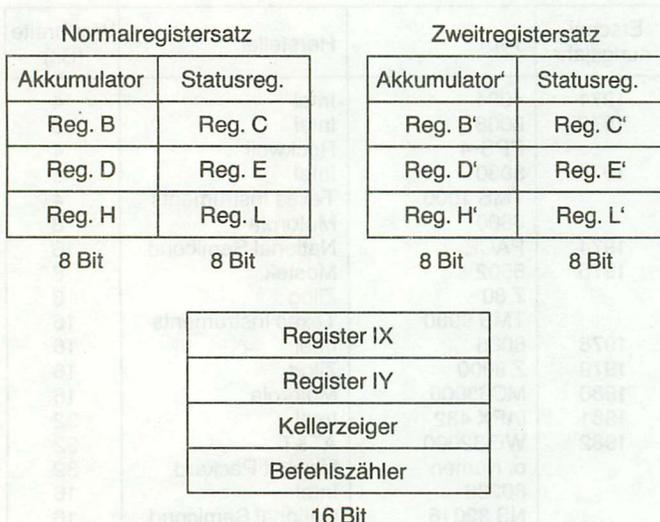


Abbildung 1: Das Registermodell des Z 80. Beim 8080 entfallen der Zweitregistersatz und die Indexregister IX und IY.

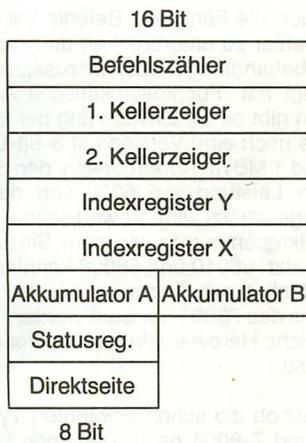


Abbildung 3:
Das Registermodell des 6809. Der erste Kellerzeiger ist für Systemzwecke reserviert, der zweite steht Anwendungsprogrammen zur Verfügung und kann auch als zusätzliches Indexregister genutzt werden. Im Direktseitenregister wird die Speicherseite festgelegt, die Funktion der Grundseite von 0 bis hexadezimal FF des 6502 übernimmt.

Von 8 zu 16 Bit

Wie schon erwähnt, gab es schon 1974, also drei Jahre nach dem ersten Mikroprozessor überhaupt, das erste 16-Bit-Modell. 1976 zog Texas Instruments mit dem TMS 9900 nach, der in den Heimcomputern TI 99/2 und TI 99/4 eingesetzt wurde. Der TMS 9900 benutzt übrigens ein ähnliches „RAM-Register“-Konzept wie der 6502, kann aber nur 64 KByte adressieren. Zum TMS 9900 gibt es seit 1982 verbesserte und kompatible Nachfolger der Serie TMS 99000 mit auf 384 KByte vergrößertem Adreßraum.

Den Durchbruch im 16-Bit-Markt schaffte erst 1978 der 8086 von Intel, zusammen mit seinem „kleineren Bruder“ 8088. Schlüssel für den Erfolg des 8086/8088 war die Tatsache, daß sich 8080- und Z 80-Software relativ leicht auf den 8086 portieren läßt und für ihn bald mit MS-DOS ein aus CP/M hervorgegangenes Betriebssystem zur Verfügung stand, so daß schnell eine Fülle von Anwendungsprogrammen angeboten werden konnte, die aus vorhandenen 8080-CP/M-Programmen entstanden waren.

Diesen Vorsprung haben die drei anderen führenden 16-Bit-Konkurrenten Z 8000 von Zilog (Markteinführung 1979), MC 68000 von Motorola (1980) und NS 32016 von National Semiconductor (1982), die als leistungsfähiger als der 8086 gelten, nicht mehr aufholen können. Bemerkenswert dabei ist, daß selbst der TI PC, der Personal Computer von Texas Instruments, nicht etwa mit einem der hauseigenen 16-Bit-Prozessoren TMS 9900 oder TMS 99000, sondern mit dem Konkurrenzprodukt 8088 ausgestattet wurde, – ein Tribut an das Markterfordernis der IBM-Kompatibilität.

Der 8088 – ein 16-Bit-Prozessor?

An dieser Stelle sei die alte Streitfrage aufgewärmt, ob der 8088 nun ein 8- oder 16-Bit-

Prozessor ist. Hardware-Leute haben darauf eine klare Antwort: Entscheidend für die Einordnung eines Prozessors ist die Breite des Datenbusses, und die beträgt beim 8088 eben 8 Bit. Konsequenterweise führt sogar die Firma Intel den 8088 in ihren Unterlagen als 8-Bit-Prozessor.

Vom Standpunkt der Software aus muß man aber bedenken, daß sich der 8088 nur durch eine längere Programmlaufzeit von seinem „größten Bruder“ 8086, einem echten 16-Bit-Typ, unterscheidet. Dies ist dadurch bedingt, daß er für Speicherzugriffe auf 16-Bit-Worte vier Taktzyklen mehr braucht als der 8086 und daß seine Befehls-„Pipeline“ eine Kapazität von nur 4 statt 6 Bytes hat. Deutlicher gesagt: Es gibt kein Programm, das (ohne Zeitmessungen) entscheiden kann, ob es auf einem 8088 oder 8086 abläuft. Insofern hat es schon einen Sinn, auch den 8088 bei den 16-Bit-Typen einzureihen.

Gerechterweise sollte man dann aber auch den MC 68000 und den NS 32016 unter 32 Bit einordnen, da beide eine interne 32-Bit-Architektur aufweisen, wobei allerdings der 68000 sich sehr wohl softwaremäßig von seinem 32-Bit-Bruder 68020 unterscheiden läßt. Da alle drei Prozessoren zusammen mit dem Z 8000 eine Generation repräsentieren, seien sie hier der Einfachheit halber gemeinsam abgehandelt.

Konstruktionen mit einem 16- oder 32-Bit-Prozessor mit 8-Bit-Datenbus, also dem 8088, 68008 oder 32008, haben neben geringeren Hardware-Kosten die inzwischen jedoch nicht mehr ins Gewicht fallen, durchaus noch einen vernünftigen Grund: Bei zeichenorientierten Anwendungen, wie sie typischerweise in der Textverarbeitung auftreten, wird ein Prozessor mit breiterem Datenbus gar nicht richtig ausgenutzt, da ja ein Zeichen im ASCII-Code durch ein Byte repräsentiert und als solches verarbeitet wird.

In diesem Zusammenhang mag es auch recht interessant sein, die Beweggründe kennenzulernen, die Chuck Peddle und Bobby Taylor, die „Väter“ des Sirius 1 der Firma Victor, dazu geführt haben, den 8088 auf diesem Mikrocomputer einzusetzen. Sie hatten ursprünglich die Prozessoren MC 68000, 8088 und die Kombinationen 6502/6502, 6502/6809 sowie 6502/Z 80 in die engere Wahl gezogen. Die Motorola-Prozessoren 68000 und 6809 (zusammen mit dem 6502) schieden als erste mangels einer breiten Basis an Anwendungssoftware aus. Ihre Entscheidung zugunsten des 8088, bei der sie noch nichts vom IBM PC wußten, gründete sich vor allem darauf, daß das Softwarehaus Microsoft in der Lage war, für den 8088 Programmiersprachen, Entwicklungswerkzeuge und Anwendungssoftware anzubieten. Außerdem legten sie Wert darauf, daß beim 8088 mit einem Adreßraum von 1 MByte die beim 6502 recht einengende Grenze von 64 KByte beseitigt ist.

Als sich dann IBM für den Einsatz des 8088 auf dem IBM PC entschied (vermutlich aus ähnlichen Gründen wie Peddle und Taylor), war das Rennen für die Intel-Familie gelaufen. Nach einer Marktuntersuchung über den Zeitraum vom ersten Quartal 1983 bis zum zweiten Quartal 1984 hatte sie 83% Marktanteil, während Motorola als stärkster Mitbewerber mit der 68000-Familie nur auf

11,5% kam. Allerdings konnte der MC 68000 seinen Anteil bei industriellen Anwendungen (also nicht in Personal Computern) und sehr leistungsfähigen Kleinrechnern mit UNIX als Betriebssystem stärken.

Die Konkurrenten des 8086

Um den Startvorsprung des 8086 aufholen zu können, mußten seine Konkurrenten mehr können und bessere Eigenschaften bieten. Vergleicht man die Architekturen des 8086 und seiner Konkurrenten, so fällt auf, daß an der Wiege des 8086 die Forderung nach Kompatibilität zum 8080 stand, während die Hauptkonkurrenten sich am Aufbau und Befehlssatz der Zentraleinheiten erfolgreicher Minicomputer orientierten, vor allem an der der DEC-Rechner PDP-11 und VAX, und zwar in der Reihenfolge Z 8000, MC 68000, NS 32000 in steigendem Maße. Dafür haben sie auf die Kompatibilität zu ihren jeweiligen Vorläufern verzichtet.

Als erste zog die Firma Zilog mit einem 16-Bit-Typ, dem Z 8000, nach, der sowohl vom Zeitpunkt der Markteinführung als auch von der Leistungsfähigkeit her zwischen dem 8086 und dem 68000 liegt. Dadurch geriet er zwischen die Fronten und konnte sich am Markt gegen die anderen beiden nicht so recht durchsetzen. Seine Anhänger schätzen an ihm vor allem die hohe Befehlsausführungsgeschwindigkeit, die zwischen der der DEC-Minicomputer PDP-11/45 und PDP-11/70 liegen soll, und eine Reihe leistungsfähiger Spezialbefehle zur Zeichenkettenverarbeitung, die die Zilog-Ingenieure dem Befehlssatz der IBM-Großcomputer der Serie 370 abgeschaut haben.

Die große Geschwindigkeit des Z 8000 rührt daher, daß seine Befehlsausführungslogik „fest verdrahtet“ und nicht, wie bei den meisten anderen neueren Mikroprozessoren, mikroprogrammiert ist. Bei einem mikroprogrammierten Prozessor wird jede Instruktion des Maschinenbefehlssatzes aus noch einfacheren Mikrobefehlen zusammengesetzt. Damit erreicht man eine höhere Flexibilität beim Prozessor-Entwurf (insbesondere von Nachfolgetypen), büßt aber auf der anderen Seite Schnelligkeit ein.

Unter dem Sammelbegriff „Z 8000“ reiht man genaugenommen die Typen Z 8001 bis Z 8004 ein, die sich durch die Art und Weise des Speicherzugriffs und der adressierbaren Speichergöße (384 KByte beim Z 8002 bzw. Z 8004 und 48 MByte in 64 KByte-Segmenten beim Z 8001 bzw. Z 8003) unterscheiden. Der Z 8000 kennt ebenso wie der 68000, aber im Gegensatz zum 8086, zwei Betriebsarten, wodurch sich besonders wirksame Schutzmechanismen für Betriebssysteme implementieren lassen. Im „privilegierten“ Systembetrieb sind mehr Befehle (insbesondere Ein/Ausgabe- und Halteinstruktionen) erlaubt als im Normalbetrieb, und es wird ein eigener Kellerzeiger verwendet.

Der Dritte im Bunde der führenden 16-Bit-Modelle ist der MC 68000 von Motorola. Er zeichnet sich vor allem dadurch aus, daß er einen „linearen“, d. h. unsegmentierten Adreßbereich von 16 MByte und lauter 32-Bit-Register besitzt, so daß er auch manchmal als 32-Bit-Mikroprozessor eingeordnet wird, obwohl er mit der Außenwelt nur über

einen 16-Bit-Datenbus verkehrt. Motorola hatte es gut verstanden, die 32-Bit-Architektur des 68000 herauszustreichen und den Eindruck zu erwecken, als sei es eine Kleinigkeit, 16 zusätzliche Datenleitungen aus dem 68000 herauszuführen, um ihn zu einem echten 32-Bit-Prozessor auszubauen. Immerhin hat es noch bis Ende 1984 gedauert, bis der 32-Bit-Nachfolger 68020 in Silizium gegossene Realität wurde.

Der Registersatz des 68000 gliedert sich in 8 Datenregister, die auch byte- oder wortweise angesprochen werden können, und in 7 Adreßregister sowie zwei Kellerregister und den Befehlszähler. Der 68000 hat zusätzlich zur arithmetisch-logischen Einheit mit 16 Bit Breite zwei weitere 16-Bit-Rechenwerke für simultane Adreßberechnungen bei einem Speicherzugriff. Darüber hinaus ist der 68000 in zwei Ebenen mikroprogrammiert, d. h., jeder Mikrobefehl wird seinerseits in eine Folge von „Nanobefehlen“ aufgelöst.

Obleich die Firma National Semiconductor 1974 mit dem „PACE“ den ersten 16-Bit-Prozessor auf den Markt brachte, konnte sie sich damit und auch mit einigen anderen Typen zunächst nicht sonderlich durchsetzen. Relativ spät, erst 1982, betrat National Semiconductor mit dem NS 16032 (später umbenannt in NS 32016) nochmals die 16-Bit-Bühne. Da auch der 32016 intern eine 32-Bit-Architektur besitzt, wird von ihm weiter unten bei den 32-Bit-Prozessoren noch genauer die Rede sein.

Erwähnung verdient an dieser Stelle auch noch das CMOS-Paar 65SC802 / 65SC816 von Western Design Center, das in der Tradition der erfolgreichen 8-Bit-Mikroprozessors 6502 steht, zu dem beide Prozessoren aufwärtskompatibel sind. Der 65SC802 kann unmittelbar gegen den 6502 ausgetauscht werden, während der 65SC816 einen vergrößerten Adreßraum von 16 MBytes bietet. Beide verfügen über einen erweiterten Befehlsatz und eine interne 16-Bit-Architektur, aber nach außen nur über einen 8-Bit-Datenbus.

Die meisten Mikroprozessoren arbeiten nicht mit konventionellen sogenannten „bipolaren“ Transistoren, sondern mit Feldefekt-Transistoren in MOS-Technik. Man unterscheidet zwischen MOS-ICs mit positiven (langsameren) und negativen Ladungsträgern, also PMOS und NMOS. Die NMOS-Technik ist mittlerweile auf Spitzenleistung hingezüchtet und wird dann auch als HMOS-Technik (mit vielerlei firmeneigenen Namensvarianten) bezeichnet. Ganz modern ist die CMOS-Technik, die PMOS- und NMOS-Transistoren auf einem Chip vereint und deren integrierte Schaltungen sich durch sehr geringen Stromverbrauch und höhere Geschwindigkeiten, aber geringere Packungsdichten auszeichnen. Aus letzterem Grunde sind bisher fast alle 32-Bit-Prozessoren noch in NMOS-Technik gefertigt; zweifellos gehört jedoch der CMOS-Technik die Zukunft. Integrierte Schaltungen mit bipolaren Transistoren sind übrigens teilweise erheblich schneller, „fressen“ aber auch mehr Strom als vergleichbare MOS-Chips.

Eine weitere Tendenz, die sich auch an der 32-Bit-Front fortsetzt, besteht darin, ganze

Minirechner-Zentraleinheiten auf einen Chip zu bringen und als Mikroprozessoren zu verkaufen. Der Vorteil solcher Konstruktionen liegt auf der Hand: Man hat für den neuen Mikroprozessor sofort ein reichhaltiges Software-Angebot aus dem Minicomputerbereich. Schon der TMS 9900 von Texas Instruments war an deren Minicomputer 990 angelehnt. Ebenso gibt es den Befehlsatz der Nova von Data General als mN602 derselben Firma und als 9445 von Fairchild, sowie den der Micronova als 9440 von Fairchild. Und DEC hat die CPU einer Basis-PDP-11 als Micro-T11 und die einer etwas leistungsfähigeren PDP-11/70 als Micro-J-11 in Silizium gegossen, wobei der J-11 sogar noch schneller ist als das Vorbild PDP-11070! Und im Vorgriff auf den 32-Bit-Bereich: Der Micro VAX 78032 von DEC packt fast eine ganze VAX-CPU auf einen Chip!

Entwicklungslinien der 16-Bit-Familien

Um den erreichten Marktvorsprung zu verteidigen, brachte Intel 1983 zur Verstärkung der 8086-Familie die Weiterentwicklungen 80 186 und 80 286 (= iAPX 186 bzw. 286) heraus. Beim 80 186, zu dem es auch noch eine 8-Bit-Ausführung 80188 gibt, wurde auf der Software-Seite der Befehlsatz des 8086 um 10 zusätzliche Maschinenbefehle erweitert. Hardwaremäßig bietet der 80186 eine Menge mehr als der 8086: höhere Geschwindigkeit und die Integration zahlreicher Funktionen auf dem Prozessorchip, für die beim 8086 zusätzlicher Schaltungsaufwand benötigt wird.

Der 80286 kennt zwar alle Befehle des 80186 und noch ein paar mehr, aber nicht dessen Hardware-Erweiterungen. Er zielt mit seiner auf dem Prozessorchip integrierten Verwaltung des virtuellen Speichers und einer Reihe von anderen Schutzmechanismen (z.B. privilegierte Systembetriebart wie beim Z 8000 und 68000) vor allem auf Mehrbenutzersysteme wie UNIX, die bisher klare Domäne des 68000 waren. Das bedeutet aber auch, daß der 80286 in Einzelsystemen wie dem IBM PC AT eigentlich völlig unterfordert ist.

Wird von einem Computer das Konzept des virtuellen Speichers verwendet, so müssen die von einem Programm verwendeten logischen Adressen auf die im Hauptspeicher vorhandenen physikalischen Adressen umgerechnet werden, wobei gleichzeitig die Zugriffsrechte überprüft werden können. Das Betriebssystem kann dann dem Programm einen viel größeren Speicher vorgaukeln, als tatsächlich vorhanden ist, indem es den Hauptspeicher auf einem Massenspeicher, z.B. einer Festplatte, „fortsetzt“.

Es gibt auch Bemühungen anderer Hersteller, Erweiterungen des 8086 anzubieten, so die CMOS-Version 80C86 der Firma Harris oder die V-Serie von NEC (ebenfalls in CMOS gefertigt), deren Befehlsatz den des 8086 umfaßt und deren Mitglieder (anders als der 8086) Maschinenprogramme des 8080 unmittelbar ausführen können.

Die Antwort von Motorola auf den 80286 war der MC 68010, dem man zu den Eigenschaften des 68000 ebenfalls Funktionen zur Verwaltung eines virtuellen Speichers

(genauer: die Fähigkeit, Befehle bei einem Adreßfehler zu unterbrechen und nach der Fehlerbehandlung wieder fortzusetzen) hinzugefügt hat. Für kostenkritische Anwendungen gibt es als Erweiterung der 68000-Familie noch eine Version mit 8-Bit-Datenbus und 1 MByte Adreßbereich, den 68008, dessen Leistung auf 60% von der des 68000 geschätzt wird. Er wird schon in einigen Mikrocomputern wie dem Sinclair QL eingesetzt. 68010 und 68008 konnten relativ einfach durch Änderung des Mikroprogramms des 68000 realisiert werden, da sie die gleiche Hardwarestruktur wie der 68000 aufweisen.

Zilog schob die schon erwähnten Typen Z 8003 und Z 8004 nach. Bei ihnen wurden einige Verbesserungen für den Betrieb mit virtuellem Speicher und mit Multiprozessor-systemen vorgenommen.

Es ist ein fast hoffnungsloses Unterfangen, eine vergleichende Wertung zwischen den führenden 16-Bit-Prozessoren zu versuchen. Zwar besteht die herrschende Meinung, daß die Leistungsfähigkeit in der Reihenfolge 8086, Z 8000, 68000 zunimmt, aber auf die zahlreichen Nachfolgetypen dieser Standardprozessoren läßt sich dieses Urteil nicht verallgemeinern. Vor allem der 80286 hat den Vorsprung des 68000 wieder aufgeholt. Geht man weiter ins Detail, so erkennt man, daß jeder Prozessor spezifische Vor- und Nachteile hat, die ihn für bestimmte Anwendungsgebiete prädestinieren und für andere eben nicht. Hinzu kommt, daß bei manchen Eigenschaften durchaus umstritten ist, ob sie als Vor- oder Nachteil anzusehen sind.

Betrachten wir als Beispiel die Speichersegmentierung (d. h. die Aufteilung des Gesamtadreßraums von insgesamt 1 MByte in maximal 64 KByte große Blöcke), wie sie von der 86-Linie von Intel und von Z 8001/8003 benutzt wird. Sie gestattet das problemlose Verschieben von Programmen im Speicher, was z.B. für einen Multiprogramm-Betrieb wichtig ist, während andererseits der Umgang mit großen Datenmengen, die nicht in ein Segment passen, reichlich erschwert wird. Unter dem Strich ergibt sich eher ein Nachteil der Speichersegmentierung. Betrachtet man aber kompliziertere Systeme mit virtuellem Speicher, so erleichtert das Segmentkonzept die Speicherverwaltung ganz erheblich. Aus diesem Grund ist der 80286 der erste und mit der brandneuen MicroVAX 78032 zusammen auch der bisher einzige Mikroprozessor, der eine komplette Speicherverwaltung auf dem Chip integriert hat und daher andere, ansonsten vergleichbare Systeme mit getrennter Speicherverwaltungseinheit deutlich an Geschwindigkeit übertrifft. Andererseits erlaubt das seitenorientierte Speicherkonzept, wie man es etwa in der NS-32000-Familie findet, eine erheblich flexiblere, aber auch aufwendigere Speicherorganisation.

Eine ähnlich umstrittene Frage ist die Ausstattung bestimmter Register des 8086 mit Sonderfunktionen. Sie sorgt für kleinere Befehlsängen und damit kompaktere Programme, da die an einer Operation beteiligten Register nicht immer angegeben werden müssen (sogenannte implizite Adressierung); andererseits erschweren solche Spezialitäten dem Assemblerprogrammieren

rer die Arbeit und erfordern häufiges umladen von Registern. Beim Arbeiten mit höheren Programmiersprachen wie BASIC, Pascal oder FORTAN fällt dies nicht so sehr ins Gewicht, da Compiler die Zuordnung von Registern ohnehin recht schematisch vornehmen und dabei spezielle Eigenheiten leicht berücksichtigen können.

Der Kampf um den 32-Bit-Markt

Die 16-Bit-Prozessoren hatten sich noch gar nicht richtig am Markt durchgesetzt, da begann schon der Kampf um den 32-Bit-Markt. Die Mikroprozessor-Hersteller hatten aus den Erfahrungen mit der Einführung der 16-Bit-Typen gelernt und versuchten nun, bei der neuen Prozessorgeneration Startvorteile zu erringen, indem sie frühzeitig die Spezifikationen der neuen Modelle bekanntgaben und Entwicklungswerkzeuge für Programme auf den neuen, noch gar nicht existenten Typen bereitstellten. Mit der tatsächlichen Realisation hingegen ließen sie ziemlich auf sich warten. Erst Anfang nächsten Jahres dürfte die 32-Bit-Runde komplett sein.

Charakteristisch für den 32-Bit-Markt ist die Erscheinung, daß sich um die neuen Mikroprozessoren ganze Familien von Zusatz-Chips gruppieren. Während in 8- und 16-Bit-Schaltungen CPUs, Koprozessoren und Peripheriebausteine verschiedener Hersteller in fast beliebiger Weise gemischt werden konnten, geht dies im 32-Bit-Bereich kaum noch, da sich die Schnittstellen zwischen den Chips, vor allem die Steuer-signale, zu sehr unterscheiden. Eine typische 32-Bit-Familie, hier dargestellt am Beispiel von National Semiconductor, besteht aus der CPU (NS 32032) als Hauptverarbeitungseinheit, meistens mit 8- und 16-Bit-Varianten (32008 bzw. 32016), dem Arithmetik-Prozessor (32081), der Speicherverwaltungseinheit oder MMU (32082, memory management unit), dem Zeitgeber (32201, timing control unit), dem Interrupt-Baustein (32202, interrupt controller) und der DMA-Steuerung (32203).

Eine Reihe von weiteren Eigenschaften lassen sich allgemein für die 32-Bit-Modelle feststellen: Adreßraum von 4 Gigabyte (dies entspricht 32 Adreßleitungen), umfangreiche Befehlssätze mit zahlreichen Adressierungsarten, „Pipelines“ zur Beschleunigung der Befehlsabarbeitung, Taktfrequenzen von 10 bis 18 MHz, Vorkehrungen für Multiprozessorbetrieb, meistens auch kleine Hilfsspeicher auf dem Prozessorchip (sogenannte „Caches“), die einen besonders schnellen Zugriff auf häufig benötigte Daten und Befehle erlauben.

Doch nun schön der Reihe nach. Bereits 1980 hatte IBM – eher ein Außenseiter auf dem Gebiet der Mikroprozessoren – versuchsweise den Befehlssatz einer IBM/370-Zentraleinheit auf einem Chip implementiert, wobei allerdings für einen kompletten Prozessor weitere ICs benötigt wurden. Diese Entwicklung wurde aber nie kommerziell verwertet.

Die nächste 32-Bit-Konstruktion gab es 1981, nämlich den iAPX 432 von Intel, dem allerdings kein nennenswerter Erfolg vergönnt war. Dies hat im wesentlichen zwei Gründe: Erstens hat der iAPX 432 eine von

allen Vorgängern völlig verschiedene Architektur, er ist also zu nichts kompatibel. Damit ist Intel über dasselbe Software-Kompatibilitätsproblem gestolpert wie vormals die Konkurrenten auf dem 16-Bit-Markt. Zweitens hält der iAPX 432 in puncto Leistungsfähigkeit nicht, was man sich von der Wortbreite von 32 Bit versprochen hat – ein 80286 bietet etwa drei- bis viermal so viel Leistung wie die drei Chips, die den 432 ausmachen. Der Mißerfolg des iAPX 432 ist umso bedauerlicher, als Intel hier versucht hat, einige ganz neue und richtungsweisende Konzeptionen, die sich großenteils auch in der Programmiersprache Ada wiederfinden, in einem Mikroprozessor zu realisieren. Mit dem 80386 aus der konventionellen 86-Linie hat Intel im Gegensatz zu den Konkurrenten den Vorstoß auf den 32-Bit-Markt noch nicht geschafft.

Da der iAPX 432, ähnlich wie der IBM-Prozessor, aus mehreren Chips besteht, datiert man den eigentlichen Beginn der 32-Bit Ära auf das Jahr 1982 mit dem Erscheinen des WE 32000 von AT&T, der früher Bellmac-32A hieß, und der namenlosen 32-Bit-CPU von Hewlett Packard. Beide Mikroprozessoren werden aber nicht einzeln, sondern nur als Bestandteil der Rechner dieser Hersteller verkauft.

Der WE 32000, zu dem es inzwischen den WE 32100 als Nachfolger gibt, ist mit diesem zusammen bisher der einzige 32-Bit-Prozessor, der vollständig in CMOS-Technik gefertigt wird. Er wurde besonders auf die Bedürfnisse des Betriebssystems UNIX sowie der Sprache C hin optimiert, die ja ebenso wie der WE 32000 in den Bell Laboratories von AT&T entwickelt wurden.

HP – nicht nur bei Taschenrechnern vorn

Die CPU von Hewlett-Packard hat unter den Mikroprozessoren in MOS-Technik, die sich im Produktionsstadium befinden, eine Reihe bisher ungebrochener Rekorde aufgestellt: Sie weist die höchste Taktfrequenz (nämlich 18 MHz), mit 1 μ m die kleinste Strukturgröße und die größte Anzahl von Transistoren (450000) auf dem Chip auf. Keiner der Konkurrenten kommt über die Grenze von 200 000 Transistoren hinaus. Damit kann die HP-CPU nicht weniger als 39 32-Bit-Register bieten, von denen 28 für den Programmierer sichtbar sind (zum Vergleich: beim MC 68 020 sind 23 sichtbar, 15 beim NS 32032). Der reichhaltige Befehlsvorrat (230 Befehle gegenüber 221 beim iAPX 432, 169 beim WE 32000, 121 beim 80286 und 85 beim 68020) umfaßt einen kompletten Satz von Gleitkomma-Operationen, für die sonst ein Koprozessor benötigt wird, und ist stack-orientiert, d.h., arithmetische und logische Befehle wie z.B. Addition, Und und Oder wirken auf die obersten Wörter des Daten-Kellers, der auf dem Prozessor-Chip beginnt und im Speicher fortgesetzt wird.

Die ersten auf dem Markt frei erhältlichen 32-Bit-Mikroprozessoren waren der NS 32032 von National Semiconductor und der NCR/32 von NCR. Die Stärke des NCR/32 liegt in seiner Fähigkeit, praktisch beliebige andere Prozessoren mit großer Effizienz emulieren zu können. Er ist nämlich mikroprogrammierbar; jedoch kann man im Gegensatz zu anderen mikroprogrammierten

Mikroprozessoren die Mikroprogramme des NCR/32 und damit den gesamten Maschinenbefehlssatz nachträglich ändern, da sie in einem jederzeit überschreibbaren Steuerspeicher außerhalb des Prozessors abgelegt sind. Das bedeutet, daß man auf dem NCR/32 je nach Bedarf einmal ein Programm im IBM-/370-Maschinencode, danach ein 8080-Maschinenprogramm ablaufen lassen kann, indem man zwischen durch den Steuerspeicher umlädt. Damit kann ein und dieselbe Hardware völlig verschiedene Gesichter annehmen wie Weiland Dr. Jekyll und Mr. Hyde – eine Eigenschaft, die man bisher nur bei Großrechnern fand.

Von der Familie NS 32000 von National Semiconductor war schon mehrfach die Rede. Ihre Architektur ist besonders eng an die des erprobten 32-Bit-Minicomputers VAX von DEC angelehnt. Der Befehlssatz gilt als besonders „orthogonal“, also regelmäßig, d. h., jeder Befehl kann, soweit logisch möglich, mit jedem Datentyp und jeder Adressierungsart kombiniert werden. Weiteres hervorstechendes Merkmal ist die Möglichkeit, die Ausführung eines Befehls zu unterbrechen und später genau an der Unterbrechungsstelle wieder aufzusetzen, wenn bei der Adreßberechnung festgestellt wurde, daß der benötigte Speicherinhalt noch nicht von der Festplatte in den Hauptspeicher geladen worden ist. Während der Unterbrechung können die fehlenden Daten nachgeladen werden. Damit läßt sich besonders elegant eine seitenorientierte Speicherverwaltung realisieren. Dabei wird der Speicher in Seiten von 512 Bytes zerstückelt, in die Programme und Daten unzusammenhängend verteilt werden können.

Vom 32032 leiten sich die 8- und 16-Bit-Typen 32008 und 32016 ab, die im internen Aufbau untereinander identisch sind. Der Vorteil von dieser Differenzierung besteht darin, daß man für eine Hardwarevielfalt, die vom Heimcomputer mit der 8-Bit-Variante bis zum Super-Mikrocomputer mit 32-Bit-Typ, Gleitkommaprozessor usw. reicht, die Programmierwerkzeuge wie Übersetzer usw. und später die Anwendungssoftware nur einmal entwickeln muß. Während aber bei den Konkurrenten diese Familien mehr oder weniger „historisch“ gewachsen sind, hat man bei der Konzeption der Serie 32000 von National Semiconductor von vornherein „Familienplanung“ betrieben. Dennoch scheint es, daß 32008 und 32016 für ihre Märkte zu spät gekommen sind und daher – im Gegensatz zum 32032 – weniger gute Zukunftsaussichten haben.

A propos Planung: Als nächstes Kind der Familie wird der 32C432 entwickelt, der den Leistungsumfang der CPU 32032 mit dem der Speicherverwaltungseinheit 32082 und der Interruptsteuerung 32202 auf einem CMOS-Chip kombinieren soll. Ehrgeizige Ziele! übrigens hat Texas Instruments offensichtlich auf eine eigene 32-Bit-Entwicklung verzichtet und fertigt die Serie 32000 als Zweithersteller.

Hat der 68020 die Nase vorn?

Vermutlich wird jedoch Motorola auf dem 32-Bit-Marktsegment die Nase vorn haben, da dem populären 68000 die Aufwärtskompatibilität zum „großen Bruder“ 68020 ein-

gebaut ist. Damit kann sich der 68020 von vornherein auf eine breite Software-Basis stützen – ein Vorteil, der bekanntlich dem 8086 auf dem 16-Bit-Markt zum Durchbruch verholfen hatte; andererseits wird der 68020 seit Ende 1984 produziert und hat damit einen deutlichen zeitlichen Vorsprung vor den 32-Bit-Nachfolgern seiner Hauptkonkurrenten auf dem 16-Bit-Markt, nämlich dem 80386 und dem Z 80000. Von der Firma Altos wird schon ein Rechner mit dem 68020 als CPU und Xenix als Betriebssystem angeboten. Der 68020 kombiniert die größere Busbreite von 32 Bit mit allen Leistungen des 68010 sowie zahlreichen bedeutsamen Verbesserungen. So hat man u. a. den Befehlssatz vergrößert und die Adressierungsmöglichkeiten erweitert sowie besondere Kooperationsmöglichkeiten mit Mikroprozessoren geschaffen.

Wie bei den 16-Bit-Prozessoren wollen wir auch im 32-Bit-Bereich auf eine vergleichende Wertung lieber verzichten, zumal noch nicht viele praktische Erfahrungen vorliegen und zumal auch die Runde noch gar nicht komplett ist: Es fehlen die Prozessoren 80386 von Intel und Z 80000 von Zilog, die beide noch nicht gefertigt werden.

Im Vergleich zu anderen Herstellern hat sich Intel mit Vorausinformationen über den 80386 reichlich zurückgehalten. Immerhin läßt sich ungefähr folgendes sagen: Er wird aufwärtskompatibel zum 80286 sein, so daß sich dann eine einheitliche Linie vom 8086 (und mit Abstrichen sogar vom 8080) zum 80386 ziehen läßt – wieder ein Beispiel ausgeprägten „Familiensinns“! Damit wird der 80386 ebenso wie der 286 die Speicherverwaltung auf dem Chip haben, aber volle 4 Gigabytes statt bisher nur 16 Megabytes physikalisch adressieren können, virtuell sogar 64 Terabytes! Es wird erwartet, daß er etwa doppelt so viel Transistoren auf dem Chip haben und zwei- bis dreimal so schnell sein wird wie der 80286, sowie daß er bis Anfang nächsten Jahres erhältlich sein soll.

Für den anderen „Nachzügler“, den Z80000 von Zilog, wurde im Gegensatz zum 80386 schon lange und kräftig die Werbetrommel gerührt – offensichtlich will man bei möglichen Kunden die Entscheidung offenhalten, um nicht wegen zu später Markteinführung trotz eines vielleicht überlegenen Produkts wieder das Nachsehen zu haben – immerhin war der Z80000 schon für Mitte 1984 angekündigt. Er wird 16 Allzweck-Register zu je 32 Bit bieten, auf Maschinencode-Ebene aufwärtskompatibel zum Z8000 sein und mit der auf dem Chip integrierten Speicherkapazität den Glaubensstreit zwischen den Befürwortern segment- oder seitenorientierter Speicherorganisation auf seine Weise lösen: Er wird beides können.

Exote mit „RISC“

Als etwas „exotisch“ gilt der „Transputer“ T 414 (mit der 16-Bit-Version T 212) der englischen Firma Inmos, die eigens für diese Entwicklung vom britischen Staat gegründet worden ist. Er verdient großes Interesse, da er ähnlich wie der iAPX 432 eine völlig neuartige Architektur bekommt, auf deren Bedürfnisse hin man bei Inmos sogar eine eigene Hochsprache namens Occam maßgeschneidert hat, die man als

Alternative zu Ada zu verkaufen versucht. Dem T 414 wurde als erstem kommerziell erhältlichen Mikroprozessor eine „RISC“-Architektur (RISC = Reduced Instruction Set Computer) gegeben.

Die „RISC“-Philosophie bedeutet eine völlige Abwendung von dem bisherigen Bemühen, neuentwickelten Mikroprozessoren immer größere Befehlssätze mit immer leistungsfähigeren Befehlen und Adressierungsarten zu geben, wodurch die Ausführungszeiten der einzelnen Instruktionen auch immer länger werden. Stattdessen versucht man nach der Devise „Weniger ist mehr“, den Befehlssatz eines „RISC“-Mikroprozessors auf eine kleine Zahl sehr schneller Instruktionen zu begrenzen und überläßt es hochgezüchteten Compilern, diesen eingeschränkten Befehlssatz optimal zu nutzen. Dabei kommen Rechner heraus, die unter ansonsten vergleichbaren Bedingungen um ein Mehrfaches schneller arbeiten als solche mit konventioneller Architektur – außer im Bereich der Gleitkomma-Operationen, wo die Leistung der „RISC“-Prozessoren bisher deutlich zurückbleibt.

Darüber hinaus hat man bei Inmos Wert darauf gelegt, dem T 414 Eigenschaften mitzugeben, die ihn zum Einsatz in Mehrprozessorsystemen prädestinieren, wie z.B. vier sehr schnelle Ein-/Ausgabekanäle auf jedem Transputer für die Interprozessorkommunikation. Ergänzend dazu hat man die Sprache Occam auf besondere Nutzung und Unterstützung von Mehrprozessorsystemen hin entwickelt; man verspricht sich allgemein von solchen Systemen weitere Leistungssteigerungen über die technischen Grenzen hinaus, die der heutigen Mikroprozessor-Technik gesetzt sind.

Trendspekulationen

Wo liegen diese Grenzen? Da die Entwicklung unaufhaltsam weitergeht, kann man nur Aussagen zum Stand der Technik (hier sei z.B. an die oben erwähnten Daten der 32-Bit-CPU von Hewlett-Packard erinnert) oder bestenfalls zu Fortschritten machen, die zwar schon im Labor erreicht wurden, aber noch nicht zur Produktionsreife gediehen sind. Es gibt eine Reihe von Maßstäben, anhand derer man den Fortschritt bei den Mikroprozessoren verfolgen kann, etwa Taktfrequenz, Anzahl der Operationen pro Sekunde, Strukturgröße, Packungsdichte oder die Zahl der Transistoren, die auf einem Chip integriert worden sind. Das Wachstum des letztgenannten Maßstabes um einen Faktor von 200 läßt sich in der nebenstehenden Tabelle verfolgen.

Die Anzahl der Transistoren hängt von einer Reihe von Faktoren ab, vor allem von der Chipfläche und der erreichten Strukturgröße, damit mittelbar auch von der verwendeten Transistor-Technik. Mit NMOS hat Hewlett-Packard Größen von 1 µm erreicht, während der Spitzenwert für CMOS derzeit bei 1,75 µm (WE 32100) liegt. In absehbarer Zukunft stehen auch für CMOS-Mikroprozessoren Strukturgrößen von 1,25 µm vor der Tür (NS 32132, CMOS-Version des 32032, sowie 65SC816), und sogar 0,75 µm werden angestrebt.

Die Beschleunigung der Taktfrequenzen von 1 MHz etwa beim Standard-6502 bis zu den 18 MHz der HP-CPU beeindruckt ebenso. Dabei muß man im Auge behalten, daß bei den verschiedenen Versionen eines Prozessors die Taktfrequenz variieren kann. Auf den Zusammenhang zwischen vergrößerter Packungsdichte und höherer Verarbeitungsgeschwindigkeit wurde ja schon hingewiesen. Ab einer Taktfrequenz von etwa 10 MHz treten Probleme auf, die zwar jedem Hochfrequenztechniker geläufig sind, aber zur Beseitigung erheblichen zusätzlichen Schaltungsaufwand erfordern: übersprechen zwischen einzelnen Leitungen des Busses, Leitungsreflexionen und ähnliches. Die derzeit bei der MOS-Technik erreichbar scheinende Grenze von 25 MHz wird auch schon von einigen der neuen Mikroprozessoren wie dem Z 80000 angepeilt.

Wann kommt der 64-Bit-Mikroprozessor?

Kommt nach 32 Bit nun auch 64 Bit? So naheliegend diese Fortsetzung der Entwicklungslinien auch wäre, sie wird zumindest in absehbarer Zeit nicht, vielleicht auch nie, eintreten. Im Laufe der Entwicklung der 32-Bit-Prozessoren ist man auf unerwartete Probleme beim Testen gestoßen, die man in der herkömmlichen („diskreten“) Rechner-technik nicht kannte. Beim Test eines Mikroprozessors müssen die Zustände einiger hunderttausend Transistoren über nur ungefähr hundert Anschluß-Pins beobachtet werden. Selbst beim Einsatz spezieller Test-Prototypen sowie schärfster Testmethoden, wie z.B. Vergrößerung der Zahl der beobachtbaren Schaltelemente durch Multiplex-Belegung der Anschlußleitungen, läßt es sich offenbar nicht vermeiden, daß sich auf dem Chip abgeschlossene Bereiche bilden, deren Verhalten sich von außen nicht mehr kontrollieren läßt.

Solange man dies Problem nicht in den Griff bekommt, ist mit 32 Bit die Grenze für Mikroprozessoren erreicht. Abhilfe ist beispielsweise durch weitgehende Modularisierung

Erscheinungsjahr	Typ	Anzahl Transistoren
1971	4004	2300
1974	8080	5000
1978	8086	29000
1982	80286	130000
1982	WE 32000	146000
1982	HP (ohne Namen)	450000

Tabelle 2: Anzahl der Transistoren für einige ausgewählte Mikroprozessoren

beim Mikroprozessor-Entwurf denkbar, oder durch vollständige Software-Simulation der Konstruktions- und Testvorgänge bei der Entwicklung eines neuen Prozessors und die fehlerfreie Umsetzung der Simulationsergebnisse in Silizium. Vielleicht helfen auch besondere elektronenmikroskopische Methoden weiter, mit denen man heute schon den Ladungszustand eines Schaltungselements auf einem Chip sichtbar machen kann.

Da man vorerst jedoch auf diesem Weg nicht weiterkommt, versucht man, die Leistungsfähigkeit von Mikrocomputer-Systemen auf andere Weise zu steigern. Der Zusammenschluß einzelner Universalprozessoren zu Multiprozessorsystemen wurde schon erwähnt. Daneben entwickelt man Spezialprozessoren, die nicht selbständig arbeiten können, sondern den Hauptprozessor eines Rechners bei bestimmten zeitaufwendigen Aufgaben entlasten sollen, die man schneller mit maßgeschneiderter Hardware lösen kann.

Mehr Leistung mit Koprozessoren

Bekanntestes Beispiel hierfür sind die Arithmetik-Prozessoren, die mathematische Operationen von der Gleitkomma-Multiplikation bis hin zur Berechnung trigonometrischer Funktionen um ein Vielfaches schneller als normale Mikroprozessoren erledigen. Schon für den 8080 gibt es einen solchen Arithmetik-Prozessor, den 8231A, der aber ohne Bedeutung blieb. Sehr populär wurde der 8087 als Numerik-Koprozessor zum 8086/8088, entsprechend gibt es den 80287 zum 80286. Bei National Semiconductor ergänzt der NS 32081 (früher 16081) die Serie 32000, und Motorola hat seit neuestem zum 68020 den MC 68881 auf den Markt gebracht.

Übrigens hat der 8087 eine interne Datenbreite von 80 Bit, was aber den oben angestellten Überlegungen zum 64-Bit-Prozessor nicht widerspricht, da der 8087 einen sehr spezialisierten Befehlssatz und daher einen einfacheren internen Aufbau als ein Universalprozessor besitzt. Außerdem ist der 8087 eng an seinen Hauptprozessor gebunden, an dessen Datenbus der 8087 laufend „mithört“, ob nicht ein für ihn bestimmter Befehl kommt. Andererseits übernimmt der Hauptprozessor z. B. sämtliche Adreßberechnungen für ihn, während wiederum der 8087 die Speicherzugriffe selbständig durchführt und auch selbst die für ihn bestimmten Befehle erkennt. Daher kann der 8087 auch nur zusammen mit einem 8086 oder 8088 eingesetzt werden.

Im Gegensatz dazu läuft die Kommunikation zwischen dem 32081 und seinem Hauptprozessor 32016 ganz anders ab. Der Hauptprozessor gibt hier die für den 32081 bestimmten Befehle und Daten an diesen weiter und erwartet dessen Antworten, bevor er sie weiterverarbeitet oder abspeichert. Zwar ist dieses Verfahren zeitaufwendiger als bei der Kopplung von 8086 und 8087, aber diese Schnittstelle läßt sich auch mit anderen Prozessoren realisieren. In der Tat wurde der 32081 auch schon mit dem MC 68000 von Motorola kombiniert, da es von Motorola keinen Arithmetik-Koprozessor für den 68000 gibt.

Manchmal werden auch die schon erwähnten Speicherverwaltungseinheiten zu den Koprozessoren gerechnet. Sie haben unter anderem die Aufgabe, bei Verwendung des Prinzips des virtuellen Speichers die logischen Adressen, die ein Programm etwa für seine Variablen benutzt, auf die physikalischen Adressen, auf denen das Programm oder seine Daten tatsächlich im Speicher abgelegt wurden, umzurechnen. Außerdem verhindern sie Adreßbereichsüberschreitungen, mit denen fehlerfreie Programme fremde Daten oder das Betriebssystem zerstören könnten.

Bei Verwendung des Prinzips des virtuellen Speichers brauchen die von einem Programm verwendeten logischen Adressen nicht mit den tatsächlich angesprochenen physikalischen Adressen im Speicher übereinzustimmen. Es ist dann Aufgabe der Speicherverwaltung, die entweder in einem eigenen Baustein oder – wie beim 80286 – auf dem Prozessorchip realisiert ist, die logischen Adressen in physikalische umzusetzen. Diese Vorgehensweise bietet zwei Vorteile: Zum einen braucht sich der Programmierer nicht darum zu kümmern, in welchem Teil des Speichers sein Programm später abläuft, was für Multiprogrammsysteme wichtig ist. Zum anderen kann das Betriebssystem den Programmen einen viel größeren Speicher vorgaukeln, als tatsächlich vorhanden ist, indem es den Hauptspeicher auf einem Massenspeicher, z.B. einer Festplatte, „fortsetzt“. Wenn nun ein Programm eine logische Adresse anspricht, die nicht sofort in eine physikalische übersetzt werden kann, so bedeutet dies, daß der gewünschte Speicherinhalt auf dem Massenspeicher liegt. Das Betriebssystem muß nun den benötigten Speicherabschnitt von der Festplatte in den Hauptspeicher laden; dann kann das unterbrochene Programm seine Arbeit fortsetzen.

Daneben gibt es noch Graphikprozessoren, die z.B. selbständig Linien verschiedener Formen auf einen Bildschirm zeichnen können, Textprozessoren, bei denen die wichtigsten Operationen eines Editors in der Hardware realisiert sind, Ein/Ausgabeprozessoren und vielerlei andere Peripherie-Prozessoren, also programmierbare Peripherie-Chips zur Entlastung des Zentralprozessors. Intel hat mit den Betriebssystem-Prozessoren 80130 und 80150 die Betriebssysteme iRMX und CP/M-86 „in Silizium gegossen“. Recht ausgefallen sind der Koprozessor PF 474, der auf Stringvergleiche spezialisiert ist, und ein noch im Entwicklungsstadium befindlicher Spezialprozessor namens SUM für logische Ableitungen, wie sie im Bereich der künstlichen Intelligenz durchgeführt werden.

Als „letzten Schrei“ gibt es individuell entwickelte Koprozessoren, die den Befehlssatz vorhandener Standard-Mikroprozessoren um Spezialbefehle für bestimmte Anwendungen (z.B. Matrix-, besondere Ein/Ausgabeoperationen oder Datenübertragungstätigkeiten) ergänzen. Sie werden in niedrigen Stückzahlen auf der Basis programmierbarer „Gate Arrays“ (Gatterfelder) gefertigt. Solche Gatterfelder enthalten eine zunächst funktionslose Menge von Logik-Elementen, denen erst in einem zusätzlichen Herstellungsgang eine individuelle Funktionsweise aufgeprägt wird.

Bedeutet solche „Individual-Prozessoren“ nicht einen Rückschritt gegenüber der ursprünglichen Idee des Mikroprozessors, der ja seinen Erfolg, wie wir eingangs erwähnt hatten, der Eigenschaft verdankt, in ganz unterschiedlichen Anwendungen eingesetzt werden zu können und daher über hohe Stückzahlen niedrige Preise zu erreichen? Sicher nicht, denn softwaregestützte Entwicklungsverfahren erlauben heutzutage ebenso wie modernere Produktionsmethoden den wirtschaftlichen Einsatz dieser „Individual-Prozessoren“ auch in geringen Stückzahlen. Dennoch – an dieser Stelle schließt sich der Kreis vom Ausgangspunkt des Mikroprozessors als Massenprodukt bis hin zu seiner Verfeinerung als individuellem Produkt.

Bleibt nur noch nachzutragen, daß in diesem Artikel bei weitem nicht sämtliche existierenden Mikroprozessoren behandelt werden konnten, sondern nur die wichtigsten Produktfamilien und interessantesten Entwicklungslinien.

Dr. Ralf Kern (1100)
Ettliger Straße 9
D-7500 Karlsruhe 1

Kodak Datashow System

Kommunikation wird leichter

A. Anwendungsgebiete für Bildschirmprojektion

Einigen Dutzend Studentinnen und Studenten bemüht sich ein Dozent, die Bedienung eines PC, das Betriebssystem und Programmierkenntnisse nahezubringen. Das Bemühen des Dozenten leidet jedoch unter einem starken Handicap: seine Tastatureingabe und die Reaktion des Systems werden zwar auf seinem Bildschirm angezeigt, doch außer ihm kann das niemand lesen. Es hilft auch kaum, daß der Dozent zusätzlich zu dem 12 Zoll Bildschirm noch einen Fernseher mit einer 69-cm Bildröhre angeschlossen hat; zwar haben einige Studenten Feldstecher mit in die Vorlesung gebracht, durch die sie angestrengt auf den Fernsehschirm schauen, doch ein gleichzeitiges Mitschreiben ist ihnen dann kaum noch möglich, außerdem ermüden ihre Augen sehr schnell. Der Dozent sieht ein, er muß zurück zur weniger praxisnahen Vermittlung über vorbereitete Folien, die jedoch nicht das geeigneteste Mittel für eine interaktive Kommunikation bilden, und zurück zur Verwendung der guten alten Tafel. Eine unbefriedigende Situation.

Heutzutage lernen Studenten aller Fachbereiche das Programmieren an leistungsfähigen Personalcomputern. Diese sind für die Hochschulen in großer Anzahl u.a. aufgrund eines Computer-Investitions-Programms (CIP) der Bundesregierung und der Landesregierungen angeschafft worden. Vor und parallel zu den praktischen Übungen an den Geräten soll das dazu notwendige Wissen möglichst praxisnah vermittelt werden. Um aus den geschilderten unbefriedigenden Situationen herauszukommen haben vereinzelt Fachbereiche einen Vorlesungs-

raum mit teuren Video-Projektionsanlagen ausgestattet. Doch nun ist man an diesen ganz bestimmten Raum gebunden, und das ist auch nachteilig, wie die folgende Situation zeigt:

Eine Gruppe von Studenten sitzt zusammen mit einer Bibliothekarin in irgendeinem Nebenraum der Hochschulbibliothek, gruppiert um einen PC, auf dem gerade zu Demonstrationszwecken der Ablauf einer Literaturrecherche über Telekommunikation zu Datenbanken gezeigt wird. Auch hier wieder: Die auf dem Bildschirm erscheinenden Texte sind für kaum jemand zu lesen, die Gruppe wird unruhig, und die den PC bedienende Bibliothekarin wiederum aufgrund der Unruhe nervös, neigt zu Fehlbedienungen. Unbefriedigend.

Eine Lösung ist die Billig-Verbindung von Overhead-Folienprojektor mit einem durchsichtigen LCD-Bildschirm, der an den PC angeschlossen ist. Diese flache LCD-Projektionsplatte wird auf einem konventionellen Projektor aufgelegt, und schon kann eine Real-Time-Präsentation beginnen. Vorbereitete oder spontane Bildschirm-Aktionen werden sofort an die Wand projiziert. Ich habe mir Geräte verschiedener Hersteller angeschaut und schließlich das Datashow-System von Kodak bestellt.

Die Flüssigkristall-Anzeige (LCD-Display oder auch LCD-Projektionsplatte) des Datashow-Systems bietet die Graphik-Auflösung der CGA-Karte, also 640 x 200 Punkte, allerdings nur in zwei Farben (weiß und tiefblau). Das System wirkt als Monitor, die LCD-Projektionsplatte zeigt dasselbe Bild an, das auf dem Computerbildschirm erscheint, und ermöglicht die Projektion digital gespeicherter Texte oder Grafiken. So lassen sich beliebige Programme, natürlich auch die Bedienung von Textverarbeitungsprogrammen und der Ablauf von Kalkulationsprogrammen von einem größeren Publikum demonstrieren.

B. Bedienung

1. LCD-Projektionsplatte

Die LCD-Projektionsplatte ist 300 mm x 330 mm x 55 mm groß und hat ein Gewicht von 2 kg mit Netzteil. LCD ist Glas und zerbrechlich. Das gesamte Datashow System wird mit einem (ordentlichen kleinen Reise-) Koffer geliefert, in das in Schaumgummiformen die einzelnen Teile sicher für einen Transport verpackt werden können. LCD ist auch hitzeempfindlich: Ein Kühlgebläse ist an die Platte fest angebaut, das zusammen mit einer auf der unteren Fläche aufgebrachten Wärmeschutzschicht für eine hohe Lebensdauer des LCD-Display sorgen wird.

Die Pixel-Auflösung beträgt 640 x 200, das Pixel ist hell oder dunkelblau. Die Farben grün, braun, cyan und weiß werden weiß, die Farben schwarz, rot, blau und violett werden schwarz projiziert.

Kontrast und Helligkeit können über eine Infrarot-Fernbedienung leicht angepaßt werden. Die Helligkeit der LCD-Projektionsplatte muß hin und wieder nachgeregelt werden, da ein LCD auf Temperaturänderungen reagiert; kein Problem mit der Fernbedienung. Der Kontrast ist beim Einsatz auf einem normalen Projektor ausreichend

groß, lediglich bei starker Sonneneinstrahlung muß für eine teilweise Abdunkelung des Raumes gesorgt werden. Die Fernbedienung erlaubt auch die Bildumkehr (Negativbild) und das Einblenden eines elektronischen Zeigers.

2. Hardware-Anschluß

Das Datashow System kann mit dem IBM PC, XT und AT oder einem 100%-ig IBM PC-kompatiblen Computer mit Farbgrafik-Adapter CGA oder erweitertem Farbgrafik-Adapter EGA (aber nur im CGA Modus) verwendet werden. Beim Anschließen der LCD-Projektionsplatte über ein Schnittstellenkabel bleibt der PC unverändert. Die LCD-Projektionsplatte wird über ein eigenes kleines Netzgerät mit Strom versorgt.

Der Farbgrafik-Adapter des PC verfügt über einen 9-poligen Stecker für das RGB-Signal (rot, grün, blau), an das normalerweise das Monitorkabel führt. Dieses wird abgezogen und mit einem Stecker des Schnittstellenkabels verbunden, ein zweiter Stecker für das RGB-Signal am PC verbunden. So bleibt der Monitor voll funktionstüchtig. Offensichtlich ist das nicht selbstverständlich, denn bei einem getesteten Gerät eines anderen Herstellers konnte der normale Bildschirmmonitor nicht gleichzeitig mit der LCD-Projektionsplatte benutzt werden.

Das Schnittstellenkabel enthält auch noch einen Stecker zum Anschließen an eine V.24-Schnittstelle des Rechners. Wird dieser Anschluß hergestellt, ist eine Steuerung der Folge von vorbereiteten Bildern mittels des mitgelieferten Showmaker-Programms möglich.

3. Showmaker Software

Im Vorlesungsbetrieb wird in der Regel ein Programm auf dem PC erstellt, verändert und zum Ablauf gebracht, über die LCD-Projektionsplatte wird eine Kopie des auf dem PC-Bildschirm ablaufenden Geschehens für alle Zuhörer angezeigt.

Darüber hinaus kann mit der mitgelieferten Software und etwas vorbereitender Arbeit eine Bildershow durchgeführt werden. Die Showmaker-Software erlaubt die Erfassung von auf einem Computerbildschirm anzeigbaren Bildern (SMCAPTOR.EXE), ihre Zusammenstellung zu einer Bildfolge (SMEDITOR.EXE) und die Steuerung der Präsentation (SMPLAYER.EXE).

SMCAPTOR.EXE ist ein speicherresidentes Programm, das auf dem Bildschirm angezeigte Texte oder Grafiken erfaßt und ein Duplikat des Bildschirminhalts in komprimierter Form auf Diskette abspeichert. Da dieses Programm speicherresident ist, kann der Inhalt von praktisch jedem Anwendungsprogramm, das auf dem PC läuft, erfaßt und dann später vorgeführt werden.

Mit dem Programm SMEDITOR.EXE lassen sich die erfaßten Bilder in eine gewünschte Reihenfolge bringen; jedes Bild kann mit einem Namen versehen werden und mit einer individuellen Projektionszeit verknüpft werden. Einfache Effekte wie schnelles oder langsames Blinken (abwechselnde Anzeige eines Bildes und des nächsten Bildes) und Invertierung (Farbumkehr) der Darstellung sind möglich.

Nach diesen Vorbereitungen kann mit dem Programm SMPLAYER.EXE die Vorführung über die PC-Tastatur oder über die Infrarot-Fernbedienung gesteuert werden, falls man keinen automatischen Ablauf möchte. So kann der automatische Ablauf gestoppt und unabhängig von der vorbereiteten Reihenfolge ein beliebiges Bild ausgewählt werden. Sehr hübsch ist auch die Möglichkeit, den Bildschirm vertikal zu teilen und zwei Bilder (in halber Breite) nebeneinander anzuzeigen.

Mit diesen drei Programmen läßt sich leicht eine ordentliche Bildershow zusammenstellen und zum Ablauf bringen.

Die Zahl der möglichen Effekte ist allerdings bei weitem nicht so groß wie die, die andere Programme bieten. So erlaubt das von Microsoft erstellte Programmpaket SHOWPARTNER (reine Software, kein LCD-Display), das ebenso aus drei Programmen mit den gleichen Funktionen wie beim Datashow-Paket zusammengesetzt ist, eine regelrechte Animation der Bilder: Bilder oder Bildteile können verschoben werden, ein Bild zu einem nächsten übergeblendet werden, das sind zwei der besonders wirkungsvollen Effekte. Selbstverständlich kann der Ablauf dieses mit viel raffinierten Effekten ausgestatteten Programms von Microsoft über die LCD-Projektionsplatte von Kodak ablaufen, dabei wird die Steuerung des Programms natürlich nicht mehr über die Fernbedienung, sondern wie sonst beim Einsatz des SHOWPARTNER-Programms nur über die Tastatur durchgeführt.

Typischerweise enthalten derartige Showprogramme auch ein Programm zur Demonstration der eigenen Leistung und zum Erlernen der Bedienung, dies gilt auch für das Showmaker-Programm. Es ist bei meiner Version in englisch gehalten, wie auch die Hilfsmenüs, die man in jedem Programm aufrufen kann.

Die Handbücher sind in deutsch abgefaßt. Es gibt eine Bedienungsanleitung für das Datashow-System und eine Anleitung für die Präsentationssoftware Showmaker, beides ausreichend ausführlich und gefällig.

C. Wertung

Ich habe lange auf ein System wie das Datashow-System gewartet. Die in früheren Jahren auf der Hannover Messe gezeigten ersten LCD-Projektoren waren zu kontrastarm, zu teuer und zeigten oft nur einen Ausschnitt des PC-Bildschirms. Heute gibt es eine Reihe von Systemen, zwischen denen man wählen kann. Es gibt inzwischen auch Geräte mit einer Auflösung von 640 x 400 Punkten und mit Graustufenungen, um die Farbe zu emulieren. Für DM 3680 (im Herbst 1987) bietet Kodak jedoch ein preiswertes und problemlos zu bedienendes System an, daß hoffentlich zudem aufgrund seines angebauten Lüfters auch beim häufigen Einsatz eine lange Zeit funktionstüchtig bleibt.

Wolfgang Fritz
Kronenstraße 34
7500 Karlsruhe

Funktion:

CED (Commandline Editor) ist ein speicherresidentes Utility, das die DOS-Oberfläche um eine Reihe von Funktionen zur Befehlseditierung erweitert. Außerdem bietet es die Möglichkeit Befehlskürzel anzulegen.

Format:

CED [-Bn1,n2,n3,n4,n5,n6] [-FDateiname]
 CED PCALL Befehl
 CED SYN *Synonym Befehl*
 CED USERSYN *Synonym Befehl*
 CED IGNORE Befehl
 CED CLEAR Funktion [[FROM] Name]
 CED CHAINCH Zeichen
 CED IGNORECH Zeichen
 CED RARAMCH Zeichen
 CED STATCH Zeichen

oder

Typ:

Intern Extern

Hinweise:

Nach dem erstmaligen Aufruf von CED sind die folgenden Funktionen verfügbar:

Alle eingegebenen Befehle werden in einem Stapel hinterlegt. Mit der Cursor-auf Taste kann man sich die letzten Befehle wieder in die Kommandozeile holen und editieren.

Zusätzlich gibt es noch eine Reihe weiterer Funktionen.

CED PCALL Befehl ruft den Befehl mit den zuletzt verwendeten Parametern auf.

CED SYN *Synonym Befehl* legt ein *Synonym* für den Befehl an.

CED USERSYN *Synonym Befehl* legt ein *Synonym* für den Befehl eines Programmes an. Diese Funktion arbeitet nur mit verträglichen Programmen (z.B. EDLIN, DEBUG, etc.).

CED IGNORE Befehl unterbindet die Ausführung des Befehls. Damit kann man bestimmte Befehle einfach abstellen.

CED CLEAR Funktion [[FROM] Name] hebt die Wirkung eines SYN, USERSYN oder IGNORE Befehls wieder auf.

CED CHAINCH Zeichen ändert die Einstellung für das Zeichen zur Trennung von Befehlen (^).

CED IGNORECH Zeichen ändert die Einstellung für das Zeichen zur direkten Bearbeitung von Befehlen durch DOS (^N).

CED PARAMCH Zeichen ändert die Einstellung für das Zeichen zur Kennzeichnung von Parametern (%).

CED STATCH Zeichen ändert die Einstellung für das Zeichen zum Anzeigen des Status (^T).

Funktion:

Der Befehl PKARC dient dazu mehrere Dateien in einer Archivdatei zusammenzufassen. Die Dateien werden dabei mit modernsten Verfahren komprimiert, um so wenig Diskettenplatz wie möglich zu verbrauchen.

Format:

[c:][Pfad]PKARC [-ocf][-nct] Optionen [G<Passwort>
 Archiv [[c:][Pfad] Dateiname [.Erw][...]]

Typ:

Intern Extern

Hinweise:

Folgende Parameter können angegeben werden:

[c:][Pfad] vor PKARC zur Angabe des Laufwerks und Pfads mit der PKARC-Befehlsdatei.

[-ocf] zum Erzeugen von Archiven im Kompatibilitätsmodus. Diese Archive können von ARC Version 5.12 und höher gelesen werden. [-nct] darf nicht zusammen mit [-ocf] angegeben werden.

[-nct] zum Erzeugen von Archiven im neuen Format. Diese Archive können nicht mit ARC 5.12 gelesen werden. Der [-nct] Parameter ist standardmäßig eingestellt und braucht deshalb nicht angegeben werden. [-nct] darf nicht zusammen mit [-ocf] angegeben werden.

Es muß eine Option angegeben werden:

- A - Zum Hinzufügen von Dateien in ein Archiv
- D - Zum Löschen von Dateien aus einem Archiv
- F - Zum erneuten Archivieren einer bereits im Archiv vorhandenen Datei. Die vorhandene Datei wird dabei durch die neue ersetzt.
- M - Zum Übertragen von Dateien in ein Archiv Funktion wie A, nur daß die Ursprungsdateien gelöscht werden
- U - Identisch mit F
- V - Inhaltsverzeichnis eines Archives
- L - Anzeigen der Software-Lizenz
- C - Dateikommentare benutzen
- X - Archivkommentar benutzen

Die Optionen A, F, M und U können von C und/oder X gefolgt werden. Die Option V kann von C gefolgt werden.

G<Passwort> verwendet ein Passwort zum Archivieren. Das Archiv ist dann ohne dieses Passwort nicht mehr zugänglich.

Archiv zur Angabe der Archivnamens im Format [c:][Pfad] Dateiname [.Erw] [c:][Pfad] Dateiname [.Erw]
 Dateien. Es können mehrere Dateien angegeben werden.
 * und ? sind erlaubt.

Quelle:

PKARC Version 3.5, 27-04-87, Copyright © 1986, 1987 PKWARE Inc.

Funktion:

Der Befehl PKXARC dient dazu Dateien aus einem Archiv in ihre Ursprungsform auszullesen.
Desweiteren können die Dateien auch auf dem Bildschirm oder Drucker ausgegeben werden. Das Inhaltsverzeichnis eines Archives kann man sich selbstverständlich auch ausgeben lassen.

Format:

[*d:*][*Pfad*][PKXARC [*Optionen*][-*G*<*Passwort*>]
Archiv [*d:*][*Pfad*] [*Dateiname* [*Erw*][...]]

Typ:

Intern Extern

Hinweise:

Folgende Parameter können angegeben werden:

[*d:*][*Pfad*] vor PKXARC zur Angabe des Laufwerks und Pfades mit der PKXARC-Befehlsdatei.

Es kann eine *Option* angegeben werden:

- R - Vorhandene Dateien ersetzen
- V - Inhaltsverzeichnis des Archives
- C - Datei auf dem Bildschirm ausgeben
- P - Datei auf dem Drucker ausgeben
- T - überprüft, ob das Archiv so kompakt wie möglich ist
- L - Zeigt die Softwarelizenz an
- E - Liest Dateien in ihre Ursprungsform aus
- X - wie E

-G<*Passwort*> - ermöglicht den Zugriff auf Archive, die mit einem Passwort versehen worden sind

Archiv zur Angabe des Archivnamens im Format [*d:*][*Pfad*] *Dateiname* [*Erw*]
[*d:*][*Pfad*] zur Angabe des Laufwerks/Pfades in den die Dateien ausgelesen werden sollen

[*Dateiname* [*Erw*]] nach Archiv zur Angabe der zu archivierenden Dateien.
Es können mehrere Dateien angegeben werden. * und ? sind erlaubt.

Quelle:

PKARC Version 3.5, 27-04-87, Copyright © 1986, 1987 PKWARE Inc.

Beim erstmaligen Aufruf von CED können folgende Parameter angegeben werden:

CED [-*Bn1,n2,n3,n4,n5,n6*] [-*FDateiname*]

Mit -B kann man die Größe der Puffer einstellen.

- n1* - Befehlsstapel (DOS)
- n2* - Befehlsstapel (Programme)
- n3* - Synonyme
- n4* - PCALL
- n5* - IGNORE
- n6* - USERSYN

Mit -F kann man eine Befehlsdatei angeben, die gleich nach dem Start von CED aufgerufen wird.

Bei Synonymen können auch mehrere Befehle angegeben werden. Dazu sind die Befehle durch ein ^ zu trennen (z.B. CED SYN X DIR /P ^ CLS).

Bei Synonymen können, wie bei Batchdateien, Parameter angegeben werden. Dazu müssen die späteren Parameter durch %n ersetzt werden. (z.B. CED SYN X COPY %1 %2).

Wird einem Befehl ^N vorangestellt, gibt CED ihn direkt an DOS weiter, ohne zu prüfen, ob es sich an ein Synonym handelt.

Der Status von CED kann mit ^T abgerufen werden.

Erfahrungsbericht Zenith Z-183 Laptop

Benutzer-Zielgruppe

Wer setzt sinnvollerweise einen tragbaren, mit sich zu führenden Computer ein? Wie bei jedem Einsatz eines Werkzeuges oder einer Maschine wird man vorher eine Kosten/Nutzen-Analyse auch in einem solchen Fall durchführen müssen. Sicherlich ist dieser Computer eine Fehlinvestition, wenn ausschließlich große und größere Datenbestände stationär verwaltet werden sollen, auch dann, wenn man unbedingt einen kleinen Computer auf dem Schreibtisch stehen haben möchte, den man schnell zuklappen kann, damit dem neugierigen Kollegen der Blick auf den Bildschirm verwehrt wird. Dem Z-183 fehlt es für die Bewältigung dieser Datenmengen an zwingend erforderlichen Voraussetzungen. Dafür bietet er aber bei zielgerichtetem Einsatz eine hervorragende Leistung.

Mögliche Benutzer

Einige Beispiele sollen erläutern, wer mit diesem Gerät hohen Nutzen erreichen kann. Für alle diejenigen, die laufend mit der Erfassung und Überarbeitung von Texten, wie die schreibende Zunft, zu tun haben, ist dieser Computer das Werkzeug schlechthin.

Ein Ingenieurteam erfaßt Daten mit sofortiger Auswertung. Auf der Baustelle können Konstruktionsänderungen in das gesamte Zahlenwerk des Projektes einfließen, gleichgültig, ob es sich dabei um die Statik, den Materialfluß, die Fertigstellungstermine oder beispielsweise die Kostenkontrolle handelt. Im Büro wird auf einfache Weise das geänderte Zahlenwerk in den stationären Computer übertragen.

Der Vermessungs-, der Kraftwerks-, der Steuerungsingenieur, Kartographen, Geowissenschaftler usw. usw. haben bei ihrer Arbeit mit einem Laptop den Computer "an der Hand".

Der Handelsvertreter hat beim Kundenbesuch sein komplettes Angebot mit allen Konditionen in Sekundenbruchteilen zur Hand. Das lästige und zeitraubende Blättern in Katalogen und Preislisten entfällt. Im Beratungsgespräch beim Kunden kann der Bauparkassen- und Versicherungsvertreter sämtliche Tarife seines Angebotes kundenspezifisch abrufen.

Der Immobilienmakler variiert für den Kunden die Finanzierung so lange, bis er die passende gefunden hat. Bei der Hereinnahme eines Verkaufsauftrages erfaßt er im Auftragstermin alle Objektdaten, bearbeitet und speichert sie entsprechend ab.

Der Steuerberater ruft im Mandanten-

gespräch außerhalb des Büros alle nötigen Informationen aus Buchhaltung, Bilanzen und betriebswirtschaftlichen Daten ab.

Der Kundenberater der Bank macht seinem Kunden quasi "on the spot" die nötigen finanztechnischen Vorschläge.

Die Reihe der Beispiele ließe sich beliebig fortsetzen.

Gerätevoraussetzungen

Mit den Abmessungen (BxTxH) von 34,3x35,5x8,7 und einem Gewicht von rund 10 Pfund ist er nicht zu groß und auch nicht zu schwer, um einfach zur Arbeit außerhalb des Büros am integrierten Handgriff mitgenommen zu werden. Die Batterieaufladung reicht wirklich lange, um ohne Steckdose irgendwo unterwegs mit diesem Gerät zu arbeiten. Die technischen Daten sind schnell aufgezählt. Der Z-183 hat einen 80C88 mit 8 Mhz getaktet als CPU, der sich per DIP-Schalter auf 4,77 Mhz umschalten läßt. Standardmäßig verfügt er über ein 640 KB großes RAM, ein 3,5-Zoll-Floppy-Laufwerk mit 720 KB und eine 10 MB-Festplatte.

Je eine serielle und parallele Schnittstelle, RGB- und Composite-Video-Ausgang für zusätzlichen Bildschirm, Anschluß für ein externes Keyboard und Bus-Stecker für externes Laufwerk sind vorhanden.

Das hintergrundbeleuchtete flache Supertwist-LCD stellt Kreise als Kreise dar, hat das gleiche Seitenverhältnis wie ein normaler Bildschirm und mißt in der Diagonale 10,5 Zoll. Sowohl Helligkeit als auch Kontrast lassen sich regeln. Die Auflösung beträgt im Grafik-Modus 640 x 200 Punkte (CGA-kompatibel); es werden 25 Zeilen 80 Zeichen breit im Textmodus gut lesbar dargestellt.

Batteriegepufferte Echtzeituhr und Kalender; auswechselbarer NiCad-Akku für bis zu 4 Stunden Betriebszeit (Herstellerangabe, Erfahrungswert ist besser); Betriebssystem MS-DOS 3.2.

Optionen und Zubehör

Co-Prozessor 8087, zusätzliches RAM von 1 MB unter EMS-Spezifikation und wird von den meisten Standardprogrammen unterstützt. Ein externes 5 1/4 Zoll-Floppy-Laufwerk als Subsystem mit eigener Stromversorgung und Verbindungskabel (Anschluß erfolgt über BUS-Stecker am Z-183), ein Anschlußkabel für die serielle Schnittstelle und die nötige Software zum Überspielen von Daten und Programmen vom PC zum Z-183 und umgekehrt sowie eine Tragetasche wie auch eine zusätzliche Batterie

mit Ladehalter sind als Zubehör vom Hersteller erhältlich.

Lieferumfang

Enthalten ist das Ladegerät, das mit einigen von Zenith geschaffenen zusätzlichen Erweiterung und Utilities ergänzte Betriebssystem MS-DOS 3.2 auf zwei 3,5-Zoll Disketten sowie zwei dazugehörige umfangreiche, deutsche Handbücher, die wie auch bei anderen Zenith-Computern von ausgezeichneter Qualität sind, und das original Referenz-Manual in englischer Sprache, das über den technischen Aufbau des Computers umfassend Auskunft gibt.

Co-Prozessor 8087, zusätzliches RAM von 1 MB unter EMS-Spezifikation, externes 5 1/4 Zoll-Laufwerk, Übertragungskabel zum Anschluß eines PC incl. Übertragungssoftware zum Überspielen von Daten und Programmen, Tragetasche und zusätzliche Batterie mit Ladehalter.

Erfahrungen

Zunächst einmal kam der Z-183 gut verpackt in einem geräumigen Karton per Zustelldienst an. Augenfällige Aufkleber sorgten dafür, daß vor dem ersten Betätigen des Ein-Schalters doch die ersten Seiten des Bediener-Handbuchs gelesen wurde. Danach ist das Installieren des Betriebssystems einfach, weil auf der zweiten Systemdiskette ein entsprechendes Batch-File aufgerufen wird, das dann die Installation automatisch vornimmt. Alles in allem ist es so einfach, daß auch ein unerfahrener Käufer diesen Computer zum Laufen bekommt. Danach muß dann lediglich die einzusetzende Software auf die Festplatte kopiert werden, sofern diese auf entsprechenden 3,5-Zoll Disketten vorliegt.

Wird der Z-183 als Zweitgerät zu einem vorhandenen PC eingesetzt, erfolgt das Kopieren der Software auf die Festplatte des Z-183 per Übertragungssoftware mit dem dazugehörigen Übertragungskabel. Es funktioniert reibungslos. Zum Test wurde Turbo-Pascal, Framework II, Multimate-Advantage und dBASE III+ einschließlich einiger dazugehöriger Programm-, Kalkulations-, Text-, Datenbank- und DFÜ-Dateien sowie passende nützliche Utilities (Norton-Commander, Norton-Utilities, Norton-Editor, Fastback und PC-Tools) auf die Festplatte übertragen.

Die Übertragung dieser Programme und Datenbestände, immerhin insgesamt 6,5 MB, nahm doch Zeit in Anspruch, da die Übertragung per Null-Modem bei rund 19.200 Baud maximaler Geschwindigkeit erfolgt, aber durch den Handshake-

GEOSTAT

HP-75 Anwender !

Programmbeschreibung

Dieses Programm wird im Herbst dieses Jahres besonders aktuell werden, wenn nämlich der erste direkt empfangbare Rundfunksatellit (DBS) gestartet wird. Wenn die Ariane mit ihrer Nutzlast nicht wieder ins Wasser fällt, kann jeder mit einer einfachen Antenne Satellitenfernsehen empfangen. Zum richtigen Ausrichten der Antenne hilft dieses Programm.

In Zeile 50 wird die geographische Breite des Empfangsortes eingegeben. In Zeile 60 wird die geographische Länge des Empfangsortes eingegeben. In den Zeilen 70 und 80 stehen Konstanten. In Zeile 110 wird der Azimut und in Zeile 130 die Elevation ausgerechnet. Zeile 140 berechnet die Entfernung Satellit-Empfangsantenne (slant range). Nach dem Starten des Programmes, mit RUN, fragt der Rechner die geographische Länge des Subsatellitenpunktes. Westliche Längen werden mit negativen Zahlen eingegeben. In der formatierten Ausgabe kommt dann Azimut, Elevation und slant range.

Azimut: Die Winkelabweichung einer beliebigen Richtung auf der Erdoberfläche gegenüber geographisch bzw. magnetisch Nord.

Elevation: Höhe eines Gestirns über dem Horizont.

Formeln:

Längen von Subsatellitenpunkten einiger Fernsehsatelliten INTELSAT VF6 : 18.5W, TDF 1A : 19W, Olympus : 19W, TV-SAT 19W, INTELSAT V : 34.5W.

Literaturhinweise:

Davidoff, Martin R.: THE SATELITE EXPERIMENTER'S HANDBOOK
 ISBN: 0-87259-004-6
 Kriebel, Henning: Satelliten-TV-Handbuch, ISBN: 3-7723-8591-5

Kontakte zum Erfahrungs- und Programm- austausch sind international gesucht von:

Harri Kalaja
 Otsolahdentie 7A6
 SF-02110 Espoo
 Finnland

Da der HP-75 nicht mehr produziert wird, aber trotzdem ein feiner Rechner ist, sind internationale Kontakte unter den Anwendern bestimmt nicht von Nachteil.

Harri ist Mitglied im finnischen Taschen-computerclub STaK und hat uns auf der Konferenz in Kopenhagen um einen Hinweis in unserer Zeitschrift gebeten.

$$\beta = \arccos(\cos \phi_1 \cdot \cos(\lambda_1 - \lambda_2)) = \text{Winkel zwischen Empfänger und Subsat.p.}$$

$$A = \arccos \frac{\tan(\phi_1)}{\tan(\beta)} = \text{Azimut}$$

$$E = \arctan \frac{R_1 \cdot \cos(\beta) - R}{R_1 \cdot \sin(\beta)} = \text{Elevation}$$

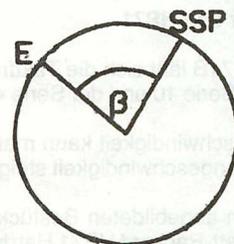
$$D = \sqrt{R_1^2 + R^2 - 2R_1 R \cdot \cos(\beta)} = \text{slant range}$$

ϕ_1, λ_1 = Breite, Länge des Empfangsortes

R = Erdradius

λ_2 = Länge des Subsatellitenpunktes

R_1 = Radius der Satellitenbahn



Formeln Thesen
 Heino Auer 2.8.1977

```

10 ! GEOSTAT
20 ! (c) Armin W. Thesen Bingerbrueck
30 REAL A,B,E,L,L1,L2,P,R,R1
40 DEGREES
50 L=49.9634 ! latitude des Empfängers S[-]
60 L1=7.8929 ! longitude des Empfängers W[-]
70 R=6371 ! Erdradius
80 R1=42164 ! Radius der Satellitenbahn
90 INPUT "SSP long. W[-]=";L2
100 B=ACOS(COS(L)*COS(L1-L2))
110 A=ACOS(-TAN(L)/TAN(B))
120 IF SGN(L1-L2)=1 THEN A=360-A
130 E=ATAN((R1*COS(B)-R)/R1/SIN(B))
140 P=SQR(R1^2+R^2-2*R*R1*COS(B))
150 DISP USING "15A,3D.D,A";"Azimut"      =";A;CHR$(1)
160 DISP USING "15A,3D.D,A";"Elevation"  =";E;CHR$(1)
170 DISP USING "15A,5D,AA";"slant range" =";P;"km"
180 END
    
```

Armin W. Thesen (3041)
 Im Schwalg 14
 6530 Bingerbrück

HP71 Speed up

SPEED UP für den HP71

Auch beim HP71B läßt sich die Taktfrequenz der CPU wie bei den Rechnern der Serie 10 und der Serie 41 erhöhen.

Die Rechengeschwindigkeit kann man so auf etwa 1,5 – 1,8fache normale Rechengeschwindigkeit steigern.

Auf dem unten abgebildeten Bestückungsplan, es ist ein Auszug aus dem Hewlett-Packard HP71 Hardware IDS, kann man die Lage der INPUT/OUTPUT-Oszillator Bausteine ersehen. Es sind C4, C5 und L1 mit den Größen 33pF, 33pF und 180mikroH.

Die Bauteile sind in SMD-Technik ausgeführt und auf die Platine aufgelötet. Man tauscht C4 und C5 durch kleine, keramische 10pF Kondensatoren aus. Die Spule kann ebensogut modifiziert werden, Richtwerte sind hier etwa 130-140mikroH, wobei jedoch kleine Festinduktivitäten schwer zu beschaffen sind.

Aber auch hier gilt, für den eigenen Rechner die optimale Kombination zu finden, indem man ein bißchen experimentiert.

Jedoch äußerste Vorsicht beim Löten! Auf ausreichende Erdung und kurze Lötzeiten achten.

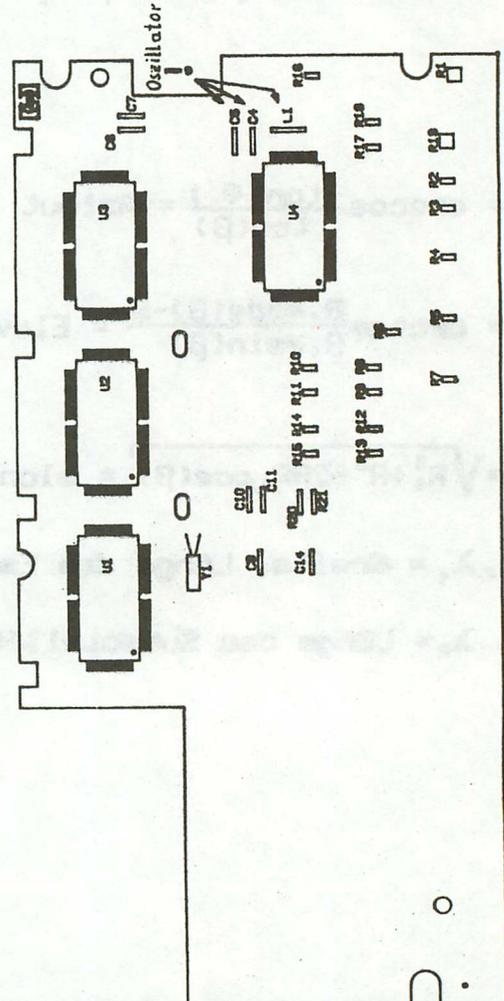
Man sollte ebenfalls auf die Leiterbahnfolie, die Ober- und Unterseite verbindet acht geben. Allzu häufiges Verbiegen und Verdrehen beschädigt sie leicht.

Für Rückfragen, Fragen und Hilfestellung stehe ich gern zur Verfügung.

Der Leser wende sich an meine unten angegebene Anschrift.

Happy Tuning!

Ralf Rosche
 Berner Heerweg 155b
 2000 Hamburg 72



TIMER

Weckprogramm mit Weckwiederholung

434 Bytes, HP-71,

Dieses kleine Programm ist eine sehr nützliche Hilfe, wenn man den HP-71 z.B. auf Reisen als Wecker benutzen möchte.

Die Bedienung ist einfach: RUN TIMER.

Als Weckzeit wird die momentane Systemzeit vorgegeben, damit man die direkte Kontrolle über die aktuelle HP-71 Uhrzeit hat. Das Datum wird nicht berücksichtigt. Ist die eingegebene Weckzeit später als die Systemzeit, dann wird der Alarm für denselben Tag gesetzt. Ist sie früher als die Systemzeit, dann gilt der Alarm für den nächsten Tag. Nach Eingabe der Alarmzeit schaltet sich der Rechner aus und sollte bis zum Alarm nicht mehr eingeschaltet werden. Durch das Einschalten wird der Alarm vorzeitig aktiviert.

Nachdem der Alarm aktiviert wurde, ist er durch Drücken jeder beliebigen Taste zu unterbrechen. Der Rechner bleibt aus und nach 5 Minuten (T=300 in Zeile 230) wird ein neuer Alarm ausgelöst. Nur durch Drücken der „/“-Taste wird das Programm beendet, so daß keine weitere „Weckwiederholung“ stattfindet.

In den Zeilen 190 bis 220 ist die Tonsequenz für das Wecksignal festgelegt. Die beiden Beep-Flags -2 (Beep An/Aus) und -25 (Laut/Leise) werden in Zeile 160 entsprechend eingestellt. Die ATTN-Taste wird mit dem POKE-Befehl in Zeile 180 „blockiert“, damit durch diese nicht das Programm versehentlich unterbrochen werden kann. In der Zeile 250 werden diese speziellen Einstellungen wieder unwirksam gemacht.

Viel Spaß beim Wecken
Alf-Norman Tietze

```

100 ! TIMER by A.-N.TIETZE / CCD e.V.
110 ! stand-alone
120 !
130 INPUT '? Time [hh.mm]: ',TIME#[1,2]#.' '&TIME#[4,5];T
140 T=IP(T)*3600+FP(T)*6000
150 IF T>TIME THEN T=T-TIME ELSE T=86400-TIME+T
160 B=FLAG(-2,0) @ L=FLAG(-25,1)
170 ON TIMER #1,T GOTO 180 @ BYE
180 OFF TIMER #1 @ POKE '2F441','F'
190 FOR I=1 TO 33
200 BEEP 1320,.1 @ BEEP 1320,.1 @ BEEP 1320,.1 @ WAIT .8
210 BEEP 880,.1 @ BEEP 880,.1 @ BEEP 880,.1 @ WAIT .8
220 BEEP 440,.1 @ BEEP 440,.1 @ BEEP 440,.1 @ WAIT 2
230 K#=KEY# @ IF K#='/' THEN 250 ELSE IF K##' ' THEN T=300 @ GOTO 170
240 NEXT I
250 POKE '2F441','0' @ B=FLAG(-2,B) @ L=FLAG(-25,L)
260 END

```

Copy Out

Filetransfer zwischen zwei HP-71

241 Bytes, HP-71, IL

Häufig kommt es vor, daß sich zwei HP-71 Anwender treffen und einen Programmaustausch machen wollen, aber kein Diskettenlaufwerk verfügbar ist. Mit dieser kleinen Routine läßt sich das nun auch ohne ein Laufwerk verwirklichen – direkt von HP-71 zu HP-71. Voraussetzung dafür ist, daß beide Rechner ein IL-Modul haben – und zwei IL-Kabel müssen verfügbar sein.

Start mit „RUN COPYOUT“ und schon wird man abgefragt, welches File zu übertragen ist. Auf dem zweiten HP-71 erscheint dann nach der Übertragung der Filename – die Übertragung hat „geklappt“.

Ein kleiner Tip am Rande: Man Sollte immer zwei IL-Kabel dabei haben, auch wenn man kein Laufwerk mitnehmen will.

Jürgen Menke
Köhlerweg 9
6233 Kelkheim

```

100 ! Prgm COPYOUT - Filetransfer zwischen zwei HP-71
110 ! by Jürgen Menke / CCD e.V
120 ! requires: HPIL
130 !
140 CONTROL OFF @ RESTORE IO @ CONTROL ON
150 INPUT "Send: ? ";A$
160 REMOTE :HP71
170 OUTPUT :HP71 ;"restoreio@a$=' '&A$&' '@COPYa$&':loop'@BEEP@a$"
180 COPY A$ TO :HP71
190 END

```



Unser Vorstand von links nach rechts: Dieter Wolf (Schatzmeister), Prof. Dr. Wolfgang Fritz (1. Vorsitzender), Werner Dworak (1. Beisitzer), Alf-Norman Tietze (2. Beisitzer) und Erich H. Klee (2. Vorsitzender).

Dieses Bild entstand während der Mittagspause der Vorstandssitzung am 4. Oktober 1987 in Frankfurt am Main. Mit von der Partie - jedoch nicht im Bild - waren noch die drei Mitglieder des Beirats: Stefan M. Schwall, Günther Schwarz und Alex Wolf. Vor- und Nachmittags wurden wichtige Themen behandelt. Übrigens: bei diesem Foto fällt auf, daß alle Vorstandsmitglieder eine Brille tragen. Ob das wohl von der vielen "Guckerei" in den PC-Bildschirm kommt?

Bug im LIST 41

Das Programm arbeitet fehlerhaft, wenn die Anzahl L der beschriebenen Records im file „HP41“ kleiner als 180 ist und bei ganzzahliger Division durch 3 einen Rest von 1 ergibt. Die Records, deren Nummer $(C+1) \cdot \text{CEIL}(L/3)$ ist, werden bei der Druckausgabe vergessen. Im Beispiel sind es die Zahlen 50 und 102 des HP-41-Programms.

Dieser Fehler tritt in meiner Fassung nicht mehr auf. Kernpunkt sind die Zeilen 165, 180 und 225. Die Variablen M und N werden nach jeder gedruckten Seite anhand der bereits bearbeiteten Records des file „HP41“ neu berechnet. Dies stellt sicher, daß der Platz auf dem Papier optimal genutzt wird, weil nun die Seiten in jedem Fall 3-spaltig bedruckt werden; und natürlich gehen keine Records mehr verloren.

Eine andere Änderung betrifft die Zeilen, die die Sonderzeichen des HP-41-Standardzeichensatzes in solche Zeichen umwandeln, die der ThinkJet auch drucken kann. Sie habe ich zu einem Block zusammengefaßt (Zeilen 75 . . . 95, Zeile 200 entfällt). Folgende Zeichen werden verarbeitet (Ähnlich macht es auch READ41 im HP-41 TranslatorPac.):

NULL byte	wird zu „*“
„Winkel“-Symbol	wird zu „<“ + „backspace“ + „,“
„Sigma“	wird zu „S“
„Ungleich“-Symbol	wird zu „Lattenzaun“
APPEND“ symbol	wird zu „geschweifte schließende Klammer“

Das Ergebnis dieser Zeichenmanipulationen ist aber nur dann befriedigend, wenn man seine Programme „ordentlich“ nach den Anweisungen in den Handbüchern von HP abfaßt. Programme, die mit synthetischen Programmzeilen gespickt sind, müssen weiterhin mit EDTEXT zur Druckreife gebracht werden. Durch die programmierte Zeichenumwandlung besteht allerdings Verwechslungsgefahr: Ein „S“, das in einer Textzeile steht, kann ursprünglich (im HP-41) ebenso gut ein „Sigma“ gewesen sein. Die Statikfunktionen sind jedoch eindeutig zu identifizieren. Sie werden als CLS, S+, S- und SREG ausgedruckt.

Einige Ein- und Ausgabezeilen habe ich geändert, um das Programm meiner Auffassung von Bedienungskomfort usw. anzupassen. Die eigentliche Funktion von LIST41 wird dadurch nicht berührt.

(Anmerkungen zu diesem Programm sollten nicht an mich gesandt werden, da mir kein HP-71 mehr zur Verfügung steht.)

Beispiel für ein fehlerhaft gelistetes HP-41 Programm

14:02	29.04	51 " "	103 XEQ 06
# steht fuer ESC		52 XEQ 07	104 STO 03
01*LBL "ELDI*"		53 ACA	105 4 E2
02 "#&k1S "		54 RCL 01	106 STO 09
03 ACA		55 RCL 01	107 RCL 08
		56 RCL 01	
		57 +	
46 -		58 " dH"	150
47 RCL 05		59 PROMPT	151 ACA
48 RCL 07		100 +	152 END
49 -		101 XEQ 06	

Programm LIST41 in korrigierter Fassung

```

1 ! (c) Oliver Lohkamp, CCD# 2622
2 ! (c) 02.07.1987 Sebastian von Borries, CCD# 2184
10 DESTROY ALL @ DELAY 0 @ IF FILESZR("HP41")<0 THEN CREATE TEXT HP41 @ GOTO 50
20 DISP "New program? (Y/N)"
30 L$=KEY$ @ IF NOT LEN(L$) THEN 30
35 L$=UPRC$(L$)
40 IF L$="N" THEN 120 ELSE IF L$="Y" THEN 50 ELSE 30
50 DISP @ ASSIGN #1 TO HP41 @ CONTROL OFF @ RESTORE IO
60 DISP "HP-41: MANIO, PRP 'xxx'"
70 ENTER :LOOP ;L$
75 X=POS(L$,CHR$(0)) @ IF X THEN L$[X,X]="*" @ GOTO 75
80 X=POS(L$,CHR$(13)) @ IF X THEN L$[X,X]="<"&CHR$(8)&" " @ GOTO 80
85 X=POS(L$,CHR$(28)) @ IF X THEN L$[X,X]="S" @ GOTO 85
90 X=POS(L$,CHR$(29)) @ IF X THEN L$[X,X]="#" @ GOTO 90
95 X=POS(L$,CHR$(127)) @ IF X THEN L$[X,X]="}" @ GOTO 95
100 PRINT #1;L$ @ DISP L$
110 IF L$[LEN(L$)-2]#"END" THEN 70
120 ASSIGN #1 TO * @ EDTEXT "HP41", ""
130 DELAY INF @ DISP "Now connect printer" @ OFF
140 DELAY 0 @ DISP "Printing..."
150 RESET HPIL @ ASSIGN #1 TO HP41 @ L=FILESZR("HP41")
160 PWIDTH 80 @ PRINT CHR$(27);"&172p1L"
165 N=CEIL((L-180*P)/3) @ M=MIN(60,N)
170 FOR Z=0 TO M-1 @ FOR C=0 TO 2
180 I=180*P+M*C+Z @ IF I>=L THEN 220
190 READ #1,I;L$
210 PRINT USING "#,25A";L$
220 NEXT C @ PRINT @ NEXT Z
225 IF M=60 AND N<>60 THEN PRINT CHR$(12) ELSE IF I>=3*(60*P+M-1) THEN 240
230 P=P+1 @ GOTO 165
240 PRINT @ DISP "Done: LIST41" @ END

```

Sebastian von Borries
Brömsenstraße 2a
2400 Lübeck 1

PRGM LIST

HP71 BASIC 1605 Bytes
Thinkjet

PRGMLIST

BASIC 1605 BYTES

page 1

```

1 | Print BASIC file on HP2225B ThinkJet printer in boldface mode.
2 |
3 | Author: Folke Danielsson
4 |       Bäckalyckevägen 40
5 |       SE-552 58 JÖNKÖPING - Sweden
6 |
10 CALL PRGMLIST
20 SUB VER(A$)
   @ A$="PRGMLIST Ver 1.0"
   @ DISP A$
30 SUB PRGMLIST
40 PURGE DMY
50 PWIDTH INF
   @ PRINT CHR$(27)&'&10L';
60 DIM B$[12],C$[18],D$[8],E$[1],G$[105],N$[9],L$[100],S$[100],T$[5],V$[8],X$[
   32]
70 INTEGER A,B,C,D,E,F,G,L,P
80 D$=DATE$
   @ E$=CHR$(13)
   @ T$=TIME$[1,5]
   @ X$=RPT$(' ',32)
90 INPUT 'Program name ? ':N$
   @ N$=UPRC$(N$)
   @ E=1
   @ F=LEN(N$)
   @ P=1
100 INPUT 'Ver ', "0.0";V$
   @ IF V$="0.0" THEN SFLAG 0 ELSE V$="Ver "&V$
110 G=POS(CAT$(E)[1,9], ' ')-1
120 IF N$=CAT$(E)[1,LEN(N$)] AND F=G THEN 140 ELSE E=E+1
130 IF LEN(CAT$(E))=0 THEN 90 ELSE 110
140 B$=CAT$(E)[12,23]
   @ C$=CHR$(27)&'&k1S'&CAT$(E)[1,8]&CHR$(27)&'&k0S'
150 GOSUB 510
   @ GOSUB 460
160 DISP 'Transforming...'
170 TRANSFORM N$ INTO TEXT DMY
   @ ASSIGN #1 TO DMY
   @ DISP 'Working...'
180 GOSUB 220
190 OFF ERROR
   @ GOSUB 520
   @ PRINT " ■";CHR$(8);"■";E$;CHR$(12);      ! ■ = CHR$(127)
200 ASSIGN #1 TO *
   @ PURGE DMY
210 DESTROY ALL
   @ DISP "Done"
   @ END
220 ON ERROR GOTO 190
230 READ #1:L$
240 OFF ERROR

```

PRGMLIST

page 2

```

250 IF L>52 THEN GOSUB 520          ! L>55 for A4 paper
    @ GOSUB 430

260 PRINT USING '#,4d,1a,4d,1a';VAL(L$(1,4));E$;VAL(L$(1,4));E$;
270 L$="    "&L$(5)
280 GOSUB 290
    @ GOTO 220
290 IF L$(6,6)="!" OR L$(6,8)="REM" THEN A=1
    @ B=0
    @ GOTO 310
300 A=1
    @ B=POS(L$,"e",A)
310 IF B=0 THEN B=LEN(L$) ELSE B=B-1
320 S$=L$(A,B)
    @ GOSUB 370
330 IF B=LEN(L$) THEN RETURN
340 A=B
    @ B=POS(L$,"e",A+2)
350 IF B=0 THEN B=LEN(L$) ELSE B=B-1
360 S$="    "&L$(A,B)
    @ GOSUB 370
    @ GOTO 330
370 G$=S$
380 D=0
390 C=D+1
    @ D=C+79
    @ PRINT G$(C,D);E$;G$(C,D)
400 L=L+1
410 IF D>=LEN(G$) THEN RETURN
420 G$(D,D)="    "
    @ GOTO 390
430 PRINT " >";CHR$(8);">";CHR$(12);
440 GOSUB 510
450 P=P+1
    @ GOTO 470
460 PRINT X$;B$;'BYTES';E$;X$;B$;'BYTES';E$;
470 PRINT USING '18a,57x,4a,3d,1a,18a,57x,4a,3d';C$;'page';P;E$;C$;'page';P
480 GOSUB 510
490 L=0
    @ IF NOT FLAG(0) THEN PRINT USING '73x,7a,1a,73x,7a';V$;E$;V$;
    @ L=L+1
500 PRINT
    @ RETURN
510 PRINT E$;RPT$("-",80);E$;RPT$("-",80)
    @ RETURN
520 FOR I=L TO 55          ! I=L TO 58 for A4 paper
    @ PRINT
    @ NEXT I
530 PRINT RPT$(" ",60);DATE$;"    ";T$;"    ";
    @ RETURN

```

For 12" European dimensioned fanfold paper insert line 55
55 PRINT CHR\$(27)&'&16d72p60F';

Formatted Input

HP-71 LEX-File FINPUT zur formatierten Eingabe in BASIC Programmen. Mit der freundlichen Genehmigung vom PPC-Paris zur Veröffentlichung freigegeben.

(übersetzt und überarbeitet von Henry Schimmer und Alf-Norman Tietze)

Trotz der enormen Leistungsfähigkeit des HP-71 sind einige seiner Möglichkeiten nur schlecht beschrieben oder umständlich anzuwenden. Er bietet z.B. geschützte Felder in seinen Eingabebefehlen, die jedoch im Benutzer-Handbuch (S. 273) leicht übersehen werden. Die Beschreibung wird von vielen Benutzern als undurchsichtig empfunden und die Anwendungsmöglichkeiten werden nicht deutlich.

Mit wenigen Worten zusammengefaßt, dienen die "geschützten" Felder dazu, bestimmte Zeichen in der Eingabeaufforderung des INPUT und LINPUT Befehls zu isolieren und vor dem Löschen bzw. Überschreiben zu schützen. Nehmen wir als Beispiel die Eingabe eines Datums. Der entsprechende Programmteil könnte mit den Möglichkeiten des HP71 so aussehen:

```
100 E$=CHR$(27)&"<" ! Cursor ausschalten
110 A$=CHR$(27)&">" ! Cursor einschalten
120 D$=A$&"tt"&E$&"."&A$&"mm"&E$&"."&A$&"jj"
130 DISP E$&"Datum: "&D$&E$; ! Eingabe-Aufforderung
140 INPUT "";I$ ! Eingabe des Datums
```

Anmerkungen:

- a) Obwohl Kommentarzeilen das Programm dokumentieren, ist die Funktionsweise nur schwer nachzuvollziehen.
- b) Die Eingabemaske wird nur langsam im Display aufgebaut.
- c) Das eingegebene Datum läßt sich zwar bei einer Fehleingabe mit der [ATTN] Taste löschen und der Cursor springt an den Anfang zurück, aber danach geht nichts mehr. Dann funktioniert nur noch eine Programmunterbrechung oder die [ENDLINE] Taste, mit welcher dann ein unkontrollierter Wert übernommen wird. Das ist natürlich ärgerlich.
- d) Außer dem Datum lassen sich mit dieser Methode trotzdem noch weitere Zeichen hinter dem Datum eingeben, da die restlichen Zeichen nicht geschützt sind. Abhilfe dagegen läßt sich nur schaffen, indem man auch alle restlichen Zeichen gegen Überschreiben schützt. Dazu muß

man rechnen: bei einer Maske mit 14 Zeichen bleiben im 96 Zeichen langen Display noch 82 übrig. Folgende Zeilen müssen in das obige Beispiel eingefügt bzw. ersetzt werden:

```
122 DIM S$(82)
124 S$=""
126 S$(82)="" ! füllt gesamten String mit Leerzeichen
128 WIDTH INF
130 DISP E$&"Datum: "&D$&E$;
```

Nach dem Datum kann man nun keine weiteren Zeichen mehr eingeben, aber dafür zeigt sich bei der Ausführung ein unerwünschtes Blinken in der Anzeige. Außerdem wird man jetzt mit einem neuen Problem konfrontiert. Durch Drücken der Cursorstasten [->] und [g]+[->] nach dem Datum verschiebt sich die Anzeige nach links bzw. bleibt eine relativ lange Zeit leer. Sehr benutzerfreundlich, nicht wahr?

Wie man leicht erkennt, ist die Anwendung von geschützten Feldern im Prinzip eine nützliche Sache - wenn sie nur entsprechend komfortabel und benutzerfreundlich wäre. Aus diesem Grunde kam uns die Idee, einen neuen Befehl in Assembler zu entwickeln, der diese Möglichkeiten nutzt und dabei die Arbeit des Programmierers erleichtert. Das Ergebnis haben wir **FINPUT** (Formatted Input) genannt.

Der Gebrauch von FINPUT

Der Schwerpunkt bei der Entwicklung dieses LEX-Files lag darin, eine einfache Benutzung für den Programmierer zu ermöglichen - ohne dabei auf komplexe Anwendungen verzichten zu müssen.

FINPUT in einer Zeile

In seiner einfachsten Form ist FINPUT eine Erweiterung des Befehls LINPUT, um geschützte Felder einfach zu verwalten. Bezogen auf unser obiges Beispiel 'genügen nun zwei Zeilen im BASIC-Programm zur Eingabe eines Datums mit geschützten Feldern.

```
100 DIM I$(6)
110 FINPUT I$,"Datum: tt.mm.jj","7P2UP2UP2UP",A
```

mit folgenden Parametern:

- Eingabevariable: I\$ zur Aufnahme für das Datum
- Aufforderungs-String: "Datum: tt.mm.jj". Es kann selbstverständlich auch eine Stringvariable übergeben werden.

- Protection-String: "7P2UP2UP2UP". Mit diesem String werden die geschützten und ungeschützten Felder definiert. Das geschieht ähnlich einem IMAGE-String, wobei "P" Protected (Geschützt) und "U" Unprotected (Ungeschützt) bedeutet. Eine Zahl vor dem jeweiligen Buchstaben bestimmt, wieviele Zeichen geschützt bzw. ungeschützt sein sollen. Das letzte "P" legt fest, daß auch der Rest der Anzeige geschützt ist.

- ATTN-Variable: (hier A) mit ihr kann kontrolliert werden, ob die Eingabe mit der [ATTN]-Taste beendet wurde. Sie enthält den Wert 0, wenn bei der Ausführung von FINPUT [ATTN] gedrückt wurde. Näheres zur Verwendung der ATTN-Variablen siehe unten.

Somit gestaltet sich jetzt die Eingabe mit geschützten Feldern fast zum "Kinderspiel". Die Verwendung von FINPUT wird schnell zur Gewohnheit werden. Aber das ist noch nicht alles, was dieser Befehl leisten kann.

Kontrolle der [ATTN]-Taste

In dem Befehl FINPUT dient die [ATTN]-Taste sowohl zur Wiederanzeige der Eingabeaufforderung als auch zum Beenden des Befehls (beim 2. Drücken). Dabei wird das Programm nicht unterbrochen. Mit Hilfe der ATTN-Variable läßt sich die [ATTN]-Taste sehr gut verwalten. Diese Variable hat den Wert 0 wenn ATTN gedrückt wurde, andernfalls ungleich Null. Ein einfacher Test genügt also zur Kontrolle:

```
IF NOT A THEN ... ! für den Fall, daß mit ATTN beendet wurde.
```

Verwaltung der Cursor-Tasten

Mit den Cursor-Tasten kann man sich nur noch auf den ungeschützten Feldern "bewegen".

Verwaltung der Eingabevariablen

Mit der Verwendung von FINPUT reduziert man das Risiko von "String Overflow", weil diese Funktion selbständig die Dimensionierung der Eingabevariablen überprüft und gegebenenfalls anpaßt. Eine zu kurz dimensionierte Variable wird automatisch "verlängert" - jedoch nur bei angegebenem Protection-String, weil aus diesem die benötigte Länge errechnet wird.

FINPUT ohne Protection-String

In vielen Fällen wird keine ausgefeilte Vorgabe mit geschützten und ungeschützten Zeichen benötigt, so daß der Protection-String nicht angegeben werden muß. Eine herkömmliche Anweisung zur Eingabe eines Filenamens lautet z.B.:

```
100 INPUT "Filename: ",F$
```

Da ein Filename im HP-71 jedoch maximal nur 8 Zeichen lang sein darf, empfiehlt sich:

```
100 DIM F$(8)
110 FINPUT F$,"Filename: ",A
120 IF NOT A THEN END
```

Einfacher geht es nicht. Die Eingabe ist hierbei trotzdem von FINPUT automatisch auf 8 Zeichen begrenzt. Der Protection-String ist daher optional. Ist er nicht vorhanden, so erstellt ihn FINPUT nach folgendem Muster:

```
STR$(LEN(string))&"PU"
```

Alle Zeichen der Eingabeaufforderung (string) sind geschützt, die nachfolgenden Zeichen sind ungeschützt bis zur maximalen Dimension der Eingabevariablen.

FINPUT über mehrere Zeilen

Eine bedeutende Eigenschaft von FINPUT ist die Verwaltung von Eingaben, die in herkömmlicher Weise über mehrere Zeilen erfolgen, in nur einer Eingabeaufforderung. Das folgende Beispiel zeigt die Eingabe von Datum und Zeit - beide mit geschützten Feldern - nach der herkömmlichen Methode:

```
100 DIM D$(6), DIM H$(6)
110 FINPUT D$,"Datum: tt.mm.jj","7P2UP2UP2UP",A
120 IF NOT A THEN END
130 ... Verarbeitung des Datums
...
200 FINPUT H$,"Zeit: hh:mm:ss","6P2UP2UP2UP",A
210 IF NOT A THEN END
220 ... Verarbeitung der Uhrzeit
...
```

Es gibt jedoch eine elegantere Lösung:

```
100 OPTION BASE 1
110 DIM I$(2)[6],M$(2),P$(2)
120 DATA Datum: tt.mm.jj,Zeit: hh:mm:ss
130 DATA 7P2UP2UP2UP,6P2UP2UP2UP
140 READ M$
150 READ P$
160 FINPUT I$,M$,P$,A
170 IF NOT A THEN END
180 ... Verarbeitung von Zeit & Datum
...
```

Diese Lösung ist wesentlich praktischer als die vorherige, weil beide Eingaben

mit einer Anweisung erledigt werden. Die Cursor-Tasten UP und DOWN dienen zum Wechsel von einer Eingabezeile in die andere. Die Taste [ENDLINE] wertet die jeweilige Zeile aus.

WICHTIG: Die Taste [RUN] wertet die aktuelle Eingabezeile aus und beendet den Befehl FINPUT. Die ATTN-Variable beinhaltet die Zeilennummer derjenigen Zeile, aus welcher FINPUT verlassen wurde. Der Wert Null zeigt immer einen Abbruch mit [ATTN] an.

Diese Anwendung von FINPUT ermöglicht es, ganze Files in nur einem einzigen Befehl "abzuarbeiten" - ohne, daß sich der Programmierer mit der Zeilenverwaltung belasten muß.

FINPUT für Menüs

Im "Tabellen-Modus" - wenn alle Zeichen geschützt sind - läßt sich FINPUT zur Auswahl eines Menüpunktes verwenden. Ein Tastendruck auf [ATTN] dient zur Auswahl der gewünschten Zeile. Ihre Nummer wird von der Funktion in der ATTN-Variablen zurückgegeben.

Zusammenfassung

Syntax:

```
FINPUT v1,exp,exp,attn
FINPUT v1,exp,v2
```

oder

```
FINPUT tab,tab,tab,attn
FINPUT tab,tab,v2
```

mit

v1, v2 = alphanumerische Variablen
exp = alphanumerischer Ausdruck
attn = numerische Variable
tab = Name von alphanumerischer Tabelle gleicher Dimension.

Die Eingabe-Aufforderung darf nur darstellbare Zeichen enthalten (also keine Steuerzeichen ASCII 0-32). Die Formatangabe im Protection-String darf nur die Buchstaben "U" und "P" als Groß- oder Kleinbuchstaben enthalten. Diesen kann - wenn nötig - eine Zahl als Faktor vorangestellt werden. Die einfachen Variablen können in Tabellen (bzw. Arrays) zusammengefaßt werden.

Bei der Verwendung von FINPUT haben folgende Tasten eine bestimmte Bedeutung:

[ATTN]

Wenn bereits Zeichen eingegeben wurden, wird der ursprüngliche Aufforderungsstring erneut angezeigt.

Nach zweimaligem Drücken wird die Funktion FINPUT abgebrochen.

[f] [OFF]

Direkter Abbruch von FINPUT.

[ENDLINE]

Die aktuelle Zeile wird ausgewertet und es wird mit der nächsten Programmzeile fortgefahren. Gibt es keine nachfolgende Programmzeile, dann wird FINPUT abgebrochen.

[RUN]

Auswertung der aktuellen Zeile und Abbruch von FINPUT. Gibt es keine nachfolgende Programmzeile, dann wird FINPUT abgebrochen.

CURSOR-Tasten

Wechsel der Zeile, ohne Auswertung der aktuellen Zeile. Bei nur einer einzigen Zeile wird der Aufforderungs-String erneut angezeigt.

Inhalt der Variablen nach Beendigung:

Die Variablen der "Eingabeaufforderung" und des "Formates" werden grundsätzlich nicht verändert.

Beim normalen Verlassen (mit [ENDLINE] oder [RUN]) beinhaltet die ATTN-Variable die Zeilennummer, in welcher FINPUT verlassen wurde. Diese Zahl liegt zwischen 1 und der Dimension der verwendeten Tabelle. Die Input-Variable enthält die eingegebenen Zeichen.

Beim Verlassen von FINPUT mit [ATTN] oder [f] [OFF] wird in der ATTN-Variablen der Wert "0" zurückgegeben. Die Input-Variable wird nicht verändert.

Alles in allem ein sehr leistungsfähiger Befehl, der viel Arbeit gemacht hat und sich hoffentlich in vielen Programmen wiederfinden wird.

Autoren:

Pierre David & Janick Tallandier
(PPC Paris)

Hinweis:

Ein Hexdump dieses LEX-Files würde etwa zwei Seiten füllen, weshalb wir auf einen Abdruck verzichten haben. Das Programm ist aber durch Einsenden eines Datenträgers bequem bei der HP-71 Bibliothek zu beziehen.

HP71B ... UND NOCH EIN LEX-FILE!

LEX-FILE DCMLX 140 Bytes
BASIC-File LEXCHK

Unter diesem Titel sollen ab nun in regelmäßiger Folge jeweils Lexfiles vorgestellt werden. Lange schon habe ich bedauert, daß die Veröffentlichung von HP-71B-Lexfiles in PRISMA zu kurz gekommen ist und habe daher versucht, etwas dagegen zu unternehmen. Das Ergebnis ist nun die vorliegende Artikelserie, die es zum Ziel hat, die vielen nützlichen Lexfiles, die es schon gibt, den HP-71B-Benutzern unter unseren Mitgliedern zugänglich zu machen. Dabei können alle vorgestellten Programme direkt aus dem Heft abgetippt werden, ein Assembler Modul ist nicht erforderlich.

Unter einigen HP-71B-Fans in meiner Bekanntschaft werden nämlich seit einiger Zeit eifrig Lexfiles getauscht und weitergegeben, die teilweise ganz tolle und ungeahnte Möglichkeiten bieten. Oft ist es aber nicht leicht, herauszufinden, von wem einzelne Lexfiles stammen und wer das Urheberrecht besitzt.

Teilweise ist es mir nun gelungen, die Genehmigung zur Veröffentlichung von bestimmten Lexfiles zu bekommen, teilweise ließ sich der Ursprung beim besten Willen nicht mehr zurückverfolgen. Diese Files veröffentlichte ich eventuell mit dem entsprechenden Hinweis und dem Aufruf, der Autor möge sich bitte melden, falls er die Veröffentlichung lesen sollte. Ich werde seinen Namen selbstverständlich nachträglich nennen. Viele der Files, die mir vorliegen, stammen von SIG (Saturn Interest Group), unserem französischen Schwesternverein und seinem Journal JPC. Andere kommen aus Amerika von Leuten im PPC und CHHU, einige aber auch von unabhängigen Autoren, die in keinem Club sind.

Hier nun erst einmal eine kurze Zusammenfassung zum Thema, was ist ein Lexfile? Es handelt sich dabei um ein Maschinenprogramm für den HP-71B, das von BASIC aus mit einem Schlüsselwort aufgerufen wird. Dieses neue Schlüsselwort stellt dann eine Erweiterung des BASIC-Wortschatzes dar, daher kommt auch der Name LEX-File (Language EXTension, engl. = Spracherweiterung). Schlüsselwörter können Befehle oder Funktionen sein. Die neuen Schlüsselwörter sind so lange verfügbar, wie sich der entsprechende Lexfile irgendwo im Memory des Rechners befindet (:MAIN oder :PORT, RAM, ROM oder EP-ROM). Entfernt man den Lexfile, so findet man anstelle der Schlüsselwörter im BASIC Programm entsprechende XWORD- oder XFN-Nummern. Dabei handelt es sich um die dezimale Darstellung der ID- und Tokennummer. Z.B. XWORD255024 würde bedeuten ID# Hex FF und Tokennummer Hex 18. Dazu gleich die Erklärung:

Jeder Lexfile muß eine ID# Nummer besitzen. Diese kann einen Wert zwischen 0 und 255 annehmen, oder hexadezimal: 00 - FF. In jedem Lexfile können bis zu 255 Schlüsselwörter (Token) stehen, die entsprechend nummeriert sind (Tokennummer). Man kann sich also ausrechnen, daß es eine ganze Menge möglicher Kombinationen gibt, dennoch aber muß achtgegeben werden, daß

nicht zwei gleiche ID# und Tokennummern zugleich im Speicher zu stehen kommen, sonst kann es zu unliebsamen Überraschungen kommen (schlimmstenfalls Systemabsturz). HP hat sich daher ein System einfallen lassen, nach dem ID# Nummern vergeben werden sollen. Zur Zeit sind die folgenden ID# Nummern reserviert:

ID# Hex	reserviert für
00-01	Mainframe (Betriebssystem)
02	Math-ROM
03	Curve fit
2F	FORTH / Assembler
52-54	Users Library
5C-5E	Scratch (zum Experimentieren)
5F-60	AMPI Statistics
61	Third party / Independent developers
A0	CHHU Interchange
E1	French Users Group
E2	CHHU ROM
F0	Text Editor
F5	Zenwand
F6	Datacomm
FE	Data Acquisition ROM
FF	HPIL ROM

Es wird sich allerdings zeigen, daß einige der veröffentlichten Lexfiles ID# Nummern besitzen, die in dieser Aufstellung nicht vorkommen. Hier haben sich die Autoren nicht an diesen Standard gehalten. Das ist nicht weiter schlimm, nur müssen wir als Benutzer jederzeit den Überblick behalten über die in unserem Rechner vorhandenen ID#- und Tokennummern. Dazu dann weiter unten ein Programm.

Der Vollständigkeit halber sei noch erwähnt, daß es auch Lexfiles gibt, die keine Schlüsselwörter besitzen, sondern nur durch sogenannte Polls und Messages in Erscheinung treten. Diese werden dann entweder durch irgendeinen Tastendruck, oder ein internes Ereignis, wie z.B. Auftreten eines Fehlers aktiviert.

Wie schreibt man eigentlich ein Lex-Programm? Zunächst einmal muß man die HP-71B Maschinensprache beherrschen. Außerdem benötigt man jede Menge Unterlagen, nämlich hauptsächlich die „IDS“ (Internal Design Specifications) von HP, mehrere dicke Bücher, in denen das „Innenleben“ des HP-71B genau dokumentiert ist. Besonders wichtig dabei sind die Einsprungadressen in Unterprogramme, die im Betriebssystem bereits vorhanden sind. Weiter braucht man dann einen Assembler (wie z.B. im FORTH/Assembler Modul), um den Assëmlercode in Maschinensprache umsetzen zu lassen.

All das ist aber nicht nötig, wenn wir uns darauf beschränken, bereits fertige Lexfiles zu veröffentlichen und für die Weitergabe aufzubereiten. Dazu dienen uns die Programme MAKEFILE und DUMPFILe aus PRISMA 7/86. Die Lexfiles werden als Listing in diesem Format veröffentlicht und können so mit Hilfe des Programms MAKEFILE abgetippt werden und sind dann sofort lauffähig im HP-71B vorhanden. Gegen Tippfehler schützt eine Checksumme in jeder Zeile. Natürlich werden alle Lexfiles sowie alle längeren BASIC-Programme aus dieser Artikelserie auch bei unserer Programmbibliothek, die von Henry Schimmer betrieben

wird, auf magnetischen Medien erhältlich sein. Auch die Programme MAKEFILE und DUMPFILe können dort bezogen werden.

Es wurde angeregt, zusätzlich zu den Dumps jeweils auch ein Assembler Sourcelisting zu den einzelnen Lexfiles zu veröffentlichen, um den an Maschinensprache Interessierten unter uns Stoff zum Studium zu geben. Leider scheitert das daran, daß ich nur zu ganz wenigen Lexfiles ein solches Sourcelisting besitze. Diese Artikelserie wendet sich auch hauptsächlich an die BASIC-Programmierer unter uns, die mit dem Angebotenen in Zukunft leichter und eleganter arbeiten können. Dazu zähle auch ich mich. Das Feld der Assemblerprogrammierung möchte ich lieber den Autoren überlassen, die mehr davon verstehen.

Abgesehen davon besteht aber die Möglichkeit, jeden Lexfile mit Hilfe eines entsprechenden Programms zu disassemblieren. Einen solchen Disassembler besitze ich auch und werde ihn ebenfalls zusammen mit einer kurzen Anleitung der Programmbibliothek HP-71 zukommen lassen.

Als ersten Lexfile möchte ich heute vorstellen:

DCMLX

Dieser Lexfile beschäftigt sich mit dem Aufbereiten von Informationen, die in einzelnen Speicherzellen des HP-71B vorhanden sind und normalerweise mit der Funktion PEEK\$ herausgeholt werden. Das Programm stammt von Andreas Schultze. Niels Nöhren hat es mir freundlicherweise zukommen lassen und so meine Sammlung bereichert.

Die Größe des Files ist 140 Bytes. 3 Token sind vorhanden.

Filename:	DCMLX
Filesize:	140 Bytes
ID# Hex:	90

Word	Token	Char
HEXADD\$	01	F
MEM\$	02	F
REV\$	03	F

HEXADD\$ (a,b) ist eine Stringfunktion, wobei die Parameter a und b Strings oder Stringausdrücke sein müssen, die eine Hexadezimalzahl darstellen (also nur Ziffern und die Buchstaben A-F). Als Ergebnis wird die Summe (Addition) der beiden Parameter geliefert, ebenfalls im Hex-Format.

MEM\$ (a, n) ist auch eine Stringfunktion. a ist ein String, der eine hexadezimale Adresse im Computer enthalten muß (max. 5-stellig) und n eine Zahl (oder numerischer Ausdruck). MEM\$ funktioniert ähnlich wie PEEK\$, nur daß statt einer hexadezimalen Darstellung die Speicherinhalte als ASCII-Werte interpretiert werden und die entsprechenden Buchstaben ausgegeben werden. n gibt dabei nicht die Anzahl der auszugebenden Nibbles, sondern die Zahl der Buchstaben (Bytes) an. Da alle Buchstaben im HP-71B intern in vertauschter Darstellung gespeichert werden (z.B. Hex 14 statt 41 für „A“), tauscht MEM\$ den Code für uns

zurück und bringt uns gleich die richtigen Buchstaben.

Beispiel: Angenommen, Sie haben ein File namens „TEST“ im Speicher. Würde man nun PEEK\$(Addr\$(„TEST“),8) ausführen, so bekäme man unverständliche „45543545“, den reversen Hex-Code von „TEST“. Die Funktion MEM\$(Addr\$(„TEST“),4) gibt aber genau das, was gemeint ist, nämlich „TEST“.

Die Stringfunktion REV\$(a) dreht den String a um. Aus „ABCDE“ wird z.B. „EDCBA“. Diese Funktion wiederum ist wichtig, da der HP-71B die Eigenschaft hat, bestimmte Größen intern verkehrt herum abzulegen. So ist zum Beispiel die Filegröße jedes Files in Nibbles im Nibble Nr. 32 bis 36 des Fileheaders abgespeichert. Um diese abzufragen tippen wir einfach: REV\$(PEEK\$(HEXADD\$(ADDR\$(„TEST“),„20“),5)) und bekommen die Hex-Darstellung der Filegröße (ohne Header). Falls „TEST“ ein leerer BASIC-File sein sollte also „00011“.

In PRISMA 7/86 befindet sich übrigens als Testprogramm zu den MAKEFILE/DUMPF-FILE Programmen eine andere Version der REV\$-Funktion. Sie funktioniert genauso, nur sollte man nur eine der beiden Versionen verwenden, da es sonst vom Betriebssystem her zu Verwechslungen kommen kann.

Als Beispielprogramm, in dem die Funktionen von DCMLX häufig verwendet werden, möchte ich hier noch das BASIC Programm LEXCHK vorstellen. Dieses Programm ist gleichzeitig sehr nützlich, wenn es darum geht, die ID# und Tokennummern festzustellen von den gerade im Speicher befind-

lichen Lexfiles. Dadurch wird es uns ermöglicht, über diese Nummern Buch zu führen und Doppelbelegungen zu vermeiden, auch wenn Sie noch kein eigenes Lexfile in Verwendung haben sollten, können Sie dieses Programm sicher dazu verwenden, die Lexfiles in etwa vorhandenen Erweiterungsmodulen zu überprüfen, wie etwa im HPIL-Modul, im Text-Editor, Kartenleser, oder was Sie sonst noch an Erweiterungen benutzen.

Vor dem Starten von LEXCHK sollte mit PRINTER IS . . . ein Drucker zugewiesen sein. Das Programm wird mit RUN gestartet und fragt zunächst nach dem Port, der überprüft werden soll. Hier kann entweder 0 für :MAIN eingegeben werden, oder die gewünschte Portnummer. Danach geht alles automatisch. Auf dem Drucker wird eine Liste aller mit diesem Port vorkommenden Lexfiles ausgegeben zusammen mit ihren Namen, ID# Nummern, Filegrößen, Tokennummern und Schlüsselwörtern. Und zwar genau in dem Format wie oben, wo die Daten des Lexfiles DCMLX angegeben sind. Genau so werde ich auch bei den zukünftig zu veröffentlichenden Lexfiles dieses Format einhalten. Falls Sie dieses Programm ohne Drucker verwenden wollen, ersetzen Sie alle PRINT-Anweisungen durch DISP und starten es im DELAY 8 Modus. Nach Anzeige jeder Zeile ist dann SPC oder irgendeine andere Taste zu drücken.

Zum Anzeigeformat von LEXCHK noch einige Anmerkungen:

1) Die Filegröße entspricht der tatsächlichen Anzahl von Bytes, die der File im Speicher belegt, inclusive aller Bytes im Header. Die Zahl ist also nicht identisch mit der im CAT angezeigten Größe.

2.) Der Buchstabe (Nibble) unter Char beinhaltet einige Informationen zu dem Schlüsselwort. Unter anderem ist daran zu erkennen, ob das Wort programmierbar ist oder nicht, und ob es sich um eine Funktion oder einen Befehl handelt. Bei unseren Beispielen kommt hauptsächlich D und F vor, wobei D einen Befehl und F eine Funktion bezeichnet. Ist Char kleiner als 8, so ist das Schlüsselwort nicht programmierbar.

Hexdump-Listing: DCMLEX
Zum Abtippen benötigen Sie ein Hex-Ladeprogramm, wie z.B. MAKEFILE aus Prisma 7/86!

DCMLX	L	ID#90	140	Bytes		
	0123	4567	89AB	CDEF	ck	
000:	4434	D4C4	8502	0202	94	
001:	802E	0000	2190	9078	BF	
002:	7F00	0091	0300	0000	18	
003:	F920	0000	0000	0000	AE	
004:	0FA0	00F1	10E4	000F	01	
005:	C103	3000	FD84	5485	07	
006:	1444	4442	107D	454D	39	
007:	4422	0725	5465	4230	4A	
008:	1FF4	118F	E83B	18DC	D1	
009:	32F0	8422	8F78	5C18	D2	
00A:	FBC6	3117	F100	8F5A	26	
00B:	5C18	F064	A110	9D41	0D	
00C:	3011	OCC4	3114	E8F4	0C	
00D:	0581	1616	DEF8	508F	F7	
00E:	7B18	18D8	2860	4422	1D	
00F:	8F78	5C18	F5A5	C1D9	6B	
010:	1088	F5A5	C111	0C08	3A	
011:	D428	C1F			81	

HP71B BASIC Programm: LEXCHK

```

5 ON ERROR GOTO 320
10 STD @ INPUT 'PORT(?) ', '0';P
15 IF P THEN P$=':PORT('&STR$(P)&')' ELSE P$=':MAIN'
20 FOR I=1 TO 999 @ IF CAT$(I,P$)=' ' THEN I=INF @ GOTO 300
30 IF CAT$(I,P$) > 'a' THEN 300
40 A$=ADDR$(CAT$(I,P$)) @ T$=PEEK$(HEXADD$(A$, '10'), 4)
50 IF T$# '802E' AND T$# 'FF00' THEN 300
60 PRINT 'Filename: ', CAT$(I,P$) [1,8]
70 A$=HEXADD$(A$, '20')
80 S=HTD(REV$(PEEK$(A$, 5)))+32 @ S=CEIL(S/2)
90 PRINT 'Filesize: ';S; ' Bytes'
100 A$=HEXADD$(A$, '5') @ I$=REV$(PEEK$(A$, 2))
110 PRINT 'ID# Hex: ', I$
120 PRINT
130 PRINT 'Word';TAB(11); 'Token Char'
140 PRINT
150 A$=HEXADD$(A$, '2') @ T0=HTD(REV$(PEEK$(A$, 2)))
160 A$=HEXADD$(A$, '2') @ T9=HTD(REV$(PEEK$(A$, 2)))
170 A$=HEXADD$(A$, '7') @ IF PEEK$(A$, 1)='0' THEN A$=HEXADD$(A$, '4F')
180 A$=HEXADD$(A$, '1') @ T$=HEXADD$(HEXADD$(A$, REV$(PEEK$(A$, 4))), 'FFFF')
190 A$=HEXADD$(A$, 'D')
200 FOR T=T0 TO T9
210 X$=HEXADD$(T$, REV$(PEEK$(A$, 3)))
220 PRINT MEM$(HEXADD$(X$, '1'), (HTD(PEEK$(X$, 1))+1)/2);TAB(11);DTH$(T) [4];
230 A$=HEXADD$(A$, '8')
240 PRINT TAB(17);PEEK$(A$, 1)
250 A$=HEXADD$(A$, '1')
260 NEXT T
270 PRINT
280 PRINT
290 PRINT
300 NEXT I
310 PUT '#43' @ END
320 IF ERRL=40 AND ERRN=61 THEN 300
330 BEEP @ DISP ERRM$;ERRL @ PAUSE @ END
    
```

Michael Fiedler
Friedrichstraße 17
6070 Langen

KALENDER

Nachfolgendes Programm druckt einen Kalender für ein beliebiges Jahr. Es ist nicht auf meinem eigenen Mist gewachsen - sondern eine 71'er-Modifikation eines Programms, das irgendwann einmal veröffentlicht wurde. Ich hoffe, der eine oder andere hat Verwendung dafür. Der Aufruf erfolgt mit CALL KALENDER.

KALENDER 815 Bytes
 ~~~~~

```

100 SUB KALENDER @ STD @ DIM M$(12),L(12),A(12),W$(7)
110 INPUT "Jahreszahl: ";J1
120 PRINT @ PRINT TAB(26);'Kalender für';J1
130 PRINT TAB(26);'===== ' @ PRINT @ PRINT
140 DATA "Montag","Dienstag","Mittwoch","Donnerstag","Freitag"
150 DATA "Samstag","Sonntag"
160 DATA "JANUAR",31,"FEBRUAR",28,"MÄRZ",31,"APRIL",30
170 DATA "MAI",31,"JUNI",30,"JULI",31,"AUGUST",31
180 DATA "SEPTEMBER",30,"OKTOBER",31,"NOVEMBER",30,"DEZEMBER",31
190 A(1)=365*J1+IP((J1-1)/4)+5
200 A(1)=A(1)-IP(A(1)/7)*7+1
210 FOR K=1 TO 7
220 READ W$(K) @ NEXT K
230 FOR J=1 TO 12
240 READ M$(J),L(J) @ NEXT J
250 IF IP(J1/4)=J1/4 THEN L(2)=29
260 FOR J=2 TO 12
270 A(J)=L(J-1)-28+A(J-1)
280 IF A(J)>7 THEN A(J)=A(J)-7
290 NEXT J
300 FOR J=1 TO 11 STEP 2
310 PRINT TAB(20);M$(J);TAB(48);M$(J+1)
320 FOR K=1 TO 7
330 PRINT TAB(3);W$(K);
340 Y=15
350 IF K<A(J) THEN Y=Y+4 @ T=K-A(J)+8 @ GOTO 370
360 T=K-A(J)+1
370 PRINT TAB(Y);T;
380 Y=Y+4 @ T=T+7
390 IF T<=L(J) THEN 370
400 J=J+1
410 IF Y>53 THEN J=J-2 @ GOTO 430
420 Y=43 @ GOTO 350
430 PRINT @ NEXT K
440 PRINT
450 NEXT J @ PRINT CHR$(12)

```

Manfred Hammer (2743)  
 Oranienstraße 42  
 6200 Wiesbaden  
 Tel. 06121/375294

Kalender für 1988  
=====

|            |                  |    |    |    |    |                 |    |    |    |    |    |
|------------|------------------|----|----|----|----|-----------------|----|----|----|----|----|
|            | <b>JANUAR</b>    |    |    |    |    | <b>FEBRUAR</b>  |    |    |    |    |    |
| Montag     |                  | 4  | 11 | 18 | 25 | 1               | 8  | 15 | 22 | 29 |    |
| Dienstag   |                  | 5  | 12 | 19 | 26 | 2               | 9  | 16 | 23 |    |    |
| Mittwoch   |                  | 6  | 13 | 20 | 27 | 3               | 10 | 17 | 24 |    |    |
| Donnerstag |                  | 7  | 14 | 21 | 28 | 4               | 11 | 18 | 25 |    |    |
| Freitag    | 1                | 8  | 15 | 22 | 29 | 5               | 12 | 19 | 26 |    |    |
| Samstag    | 2                | 9  | 16 | 23 | 30 | 6               | 13 | 20 | 27 |    |    |
| Sonntag    | 3                | 10 | 17 | 24 | 31 | 7               | 14 | 21 | 28 |    |    |
|            | <b>MÄRZ</b>      |    |    |    |    | <b>APRIL</b>    |    |    |    |    |    |
| Montag     |                  | 7  | 14 | 21 | 28 | 4               | 11 | 18 | 25 |    |    |
| Dienstag   | 1                | 8  | 15 | 22 | 29 | 5               | 12 | 19 | 26 |    |    |
| Mittwoch   | 2                | 9  | 16 | 23 | 30 | 6               | 13 | 20 | 27 |    |    |
| Donnerstag | 3                | 10 | 17 | 24 | 31 | 7               | 14 | 21 | 28 |    |    |
| Freitag    | 4                | 11 | 18 | 25 |    | 1               | 8  | 15 | 22 | 29 |    |
| Samstag    | 5                | 12 | 19 | 26 |    | 2               | 9  | 16 | 23 | 30 |    |
| Sonntag    | 6                | 13 | 20 | 27 |    | 3               | 10 | 17 | 24 |    |    |
|            | <b>MAI</b>       |    |    |    |    | <b>JUNI</b>     |    |    |    |    |    |
| Montag     |                  | 2  | 9  | 16 | 23 | 30              | 6  | 13 | 20 | 27 |    |
| Dienstag   |                  | 3  | 10 | 17 | 24 | 31              | 7  | 14 | 21 | 28 |    |
| Mittwoch   |                  | 4  | 11 | 18 | 25 |                 | 1  | 8  | 15 | 22 | 29 |
| Donnerstag |                  | 5  | 12 | 19 | 26 |                 | 2  | 9  | 16 | 23 | 30 |
| Freitag    |                  | 6  | 13 | 20 | 27 |                 | 3  | 10 | 17 | 24 |    |
| Samstag    |                  | 7  | 14 | 21 | 28 |                 | 4  | 11 | 18 | 25 |    |
| Sonntag    | 1                | 8  | 15 | 22 | 29 |                 | 5  | 12 | 19 | 26 |    |
|            | <b>JULI</b>      |    |    |    |    | <b>AUGUST</b>   |    |    |    |    |    |
| Montag     |                  | 4  | 11 | 18 | 25 | 1               | 8  | 15 | 22 | 29 |    |
| Dienstag   |                  | 5  | 12 | 19 | 26 | 2               | 9  | 16 | 23 | 30 |    |
| Mittwoch   |                  | 6  | 13 | 20 | 27 | 3               | 10 | 17 | 24 | 31 |    |
| Donnerstag |                  | 7  | 14 | 21 | 28 | 4               | 11 | 18 | 25 |    |    |
| Freitag    | 1                | 8  | 15 | 22 | 29 | 5               | 12 | 19 | 26 |    |    |
| Samstag    | 2                | 9  | 16 | 23 | 30 | 6               | 13 | 20 | 27 |    |    |
| Sonntag    | 3                | 10 | 17 | 24 | 31 | 7               | 14 | 21 | 28 |    |    |
|            | <b>SEPTEMBER</b> |    |    |    |    | <b>OKTOBER</b>  |    |    |    |    |    |
| Montag     |                  | 5  | 12 | 19 | 26 | 3               | 10 | 17 | 24 | 31 |    |
| Dienstag   |                  | 6  | 13 | 20 | 27 | 4               | 11 | 18 | 25 |    |    |
| Mittwoch   |                  | 7  | 14 | 21 | 28 | 5               | 12 | 19 | 26 |    |    |
| Donnerstag | 1                | 8  | 15 | 22 | 29 | 6               | 13 | 20 | 27 |    |    |
| Freitag    | 2                | 9  | 16 | 23 | 30 | 7               | 14 | 21 | 28 |    |    |
| Samstag    | 3                | 10 | 17 | 24 |    | 1               | 8  | 15 | 22 | 29 |    |
| Sonntag    | 4                | 11 | 18 | 25 |    | 2               | 9  | 16 | 23 | 30 |    |
|            | <b>NOVEMBER</b>  |    |    |    |    | <b>DEZEMBER</b> |    |    |    |    |    |
| Montag     |                  | 7  | 14 | 21 | 28 | 5               | 12 | 19 | 26 |    |    |
| Dienstag   | 1                | 8  | 15 | 22 | 29 | 6               | 13 | 20 | 27 |    |    |
| Mittwoch   | 2                | 9  | 16 | 23 | 30 | 7               | 14 | 21 | 28 |    |    |
| Donnerstag | 3                | 10 | 17 | 24 |    | 1               | 8  | 15 | 22 | 29 |    |
| Freitag    | 4                | 11 | 18 | 25 |    | 2               | 9  | 16 | 23 | 30 |    |
| Samstag    | 5                | 12 | 19 | 26 |    | 3               | 10 | 17 | 24 | 31 |    |
| Sonntag    | 6                | 13 | 20 | 27 |    | 4               | 11 | 18 | 25 |    |    |

## Mathematik

## Eigenwerte und Eigenvektoren

```

100 DESTROY ALL
110 INPUT "DIM: ";N
120 OPTION BASE 0 @ DIM D(N),P(N),C(N,N)
130 OPTION BASE 1 @ DIM A(N,N),B(N,N) @ COMPLEX E(N),Z(N,N)
140 MAT INPUT A
150 M=INT(-N/2)
160 FOR T=M TO M+N
170 MAT B=IDN @ MAT B=(T)*B @ MAT B=A-B
180 X=1 @ FOR I=N TO 0 STEP -1 @ C(R,I)=X @ X=X*T @ NEXT I
190 D(R)=DET(B) @ R=R+1
200 NEXT T
210 MAT P=SYS(C,D)
220 MAT E=PROOT(P) @ DELAY INF,1/6 @ MAT DISP E
230 K$=KEY$ @ INPUT "EIGENVEKTOR NR. ", "0":V
240 IF NOT V THEN DELAY 0 @ END
250 DESTROY A0,A1 @ COMPLEX B(N,N),Y,C(N) @ DIM A0(N),A1(N) @ G=ABS(E(V))*0.00001
260 F=0 @ MAT B=IDN @ MAT B=(E(V))*B @ MAT B=A-B
270 FOR K=1 TO N @ FOR I=1 TO N
280 IF NOT (A1(I)>0 OR ABS(B(I,K))<G) THEN 300
290 NEXT I @ GOTO 370
300 A0(K)=I @ A1(I)=K @ F=F+1
310 FOR L=1 TO N @ C(L)=B(I,L) @ NEXT L @ MAT C=(1/C(K))*C
320 FOR J=1 TO N
330 Y=B(J,K)
340 FOR L=1 TO N
350 B(J,L)=B(J,L)-C(L)*Y @ NEXT L @ NEXT J
360 FOR L=1 TO N @ B(I,L)=C(L) @ NEXT L
370 NEXT K
380 R=N-F @ COMPLEX Z(R,N)
390 V=0 @ FOR K=1 TO N @ I=A0(K)
400 IF I<1 THEN V=V+1 @ Z(V,K)=1 @ GOTO 450
410 W=0 @ FOR L=1 TO N
420 IF A0(L)>0 THEN 440
430 W=W+1 @ Z(W,K)=-B(I,L)
440 NEXT L
450 NEXT K
460 MAT DISP Z; @ GOTO 230

```

Heiko Schmale  
Steinmeisterstr. 8  
4980 Bünde

# Druckerakkus erneuern

Ihr kennt den Ärger sicher auch: Der alte Druckerakku ist hin, und ein neuer von HP geht nicht unter 90 DM über die Ladentheke. In der folgenden Bauanleitung zeige ich Euch, wie man mit etwas Eigenarbeit und nur 32 DM (je nach Akku-Preis) zu einem neuen Akkupaket kommt.

Zunächst muß das Kunststoffgehäuse des alten Akkupaketes aufgebrochen werden. Dazu ritzt man die Nähte an den Stirnseiten (Pfeile in Bild 1) mit einem Teppichbodenmesser ein, und hebt die Gehäusehälften mit einem Schraubenzieher an den Längsseiten auf. Wer keine Krankenversicherung hat, sollte diese Methode allerdings vergessen, das Akkupaket in einen Schraubstock vorsichtig einspannen, und die in Bild 1 gezeigten Stirnnähte mit einer Laubsäge öffnen. Jetzt lassen sich die beiden kurzen und der lange HP-Akku herausnehmen. Die Anschlußlötfahnen (Bild 2, Nr. 3) sind durch Punktschweißungen an den Akkus befestigt, und werden mit einer Zange vorsichtig abgezogen. Einen der kurzen Akkus kann man als Muster verwenden und in einem Elektronik-Bastlerladen (s. Gelbe Seiten, Rubrik „Elektronik“) kaufen. Bei den Akkus handelt es sich um den IEC-Typ KR 23/43, Größe SC, und **müssen** Lötfahnen haben (sonst geht es einfach nicht!), einen Durchmesser von 22,5 mm ( $\pm 0,5$ ) und eine Höhe von 42,5 mm ( $\pm 0,5$ ).

Ich habe dazu den Typ P-High Amp-1Z von Panasonic verwendet (der Hersteller nennt die Fa. Grevener Batteriehandel, Münsterstr. 103, 4402 Grevener, Tel. (0 25 71) 43 72 als Versandhändler).

Von anderen Herstellern sind ebenfalls solche Akkus erhältlich, wobei allerdings zu prüfen ist, ob diese auch mit Lötfahnen geliefert werden können:

- z.B. - General Electric, Cs
- Varta, RSX 1.2
- SAFT, 1.2 SC
- Everready, NCC 120
- Sanyo, N-1200 SC

Es handelt sich übrigens nicht um die sog. Babyzelle, diese ist für den Druckerakku zu groß.

An eine der neuen Zellen (Bild 2.1) wird nun eine der Anschlußlötfahnen (Bild 2.3) aus dem HP-Akkupaket an die Lötfahne des Minus-Pols (Bild 2.2) gelötet. Dazu benötigt man säurefreies Elektronik-Lot und einen LötKolben/-pistole mit ca. 30 W und einer brauchbaren Lötspitze - der Akku sollte nämlich nicht vorgeglüht werden.

Zumindest der Panasonic Akku hat unter dem Pluspol ein Sicherheitsventil, das durch besonders intensive Lötarbeiten zerstört werden kann. Bestreicht man alle Lötfahnen vorher noch mit etwas Löt fett, geht alles noch besser.

Die zweite Anschlußlötfahne aus dem HP-Akkupaket wird an den Plus-Pol einer anderen Zelle gelötet. Zum Schluß werden alle Zellen wie in Bild 3 dargestellt verbunden. Zu beachten ist, daß die Zellen 1 und 4 **keine** Verbindung haben - andernfalls geht alles in Rauch auf (schlecht!).

Beim Zusammenlöten wird immer der Minuspol einer Zelle mit dem Pluspol einer anderen verbunden, sonst geht - wenn überhaupt noch etwas geht - zuerst die BAT-Lampe des Druckers an, und der verkehrt eingesetzte Akku kaputt. In diesem Fall **sofort** nachsehen!

Nun werden die Akkus in das HP-Akkugehäuse eingesetzt, und dessen Hälften wiederum mit Klebeband verbunden. Jetzt sollten auch die Anschlußlötfahnen des Akkupaketes aus dem Gehäuse gucken können, dann ist alles richtig (hoffentlich!). Die alten Akkus gehören entweder in Sammelbehälter für Knopfzellen oder sollten als Sondermüll dem städt. Fuhramt übergeben werden. Um möglichst lange Freude mit dem neuen Akkupaket zu haben, sollte man folgendes Beachten:

Das Laden und Entladen der Akkus sollte am Besten bei Zimmertemperatur erfolgen (ca. 20° C), sonst lädt sich der Akku nicht vollständig auf, bzw. macht früher schlapp.

Wird der Akku längere Zeit nicht gebraucht, sollte er geladen und kühl gelagert werden (also nicht auf der Heizung). Nach Angaben

von Panasonic kann ein bei 0°C gelagerter Akku nach 6 Monaten immer noch über 70% seiner Leistung anbieten, ein bei 45°C gelagerter hat nach nur zwei Monaten keine 20% mehr und ist damit Tiefentladen (schlecht für die Lebenserwartung).

Spätestens beim Aufleuchten der BAT-Lampe sollte der Akku wieder geladen werden. Akkus die Tiefentladen oder ungünstig gelagert wurden, können sich teilweise wieder regenerieren. Ist nach dreimaligem Laden und Entladen (= drucken) keine ausreichende Leistung mehr vorhanden, sollte man wieder austauschen. Das Laden sollte nicht zu lange dauern, max. 15 Std., sonst leidet die Kapazität (= Lebenserwartung). Wann der Akku voll ist, kann man über die Temperatur feststellen: Es sollte nicht mehr geladen werden, wenn der Akku sich über 45°C erwärmt (also das Fieberthermometer platzt). Wer beim Laden hin und wieder unter den Drucker faßt, merkt es auch bald ohne Thermometer.

Ralf Pfeifer (116)  
Rubensstraße 5  
5000 Köln 50

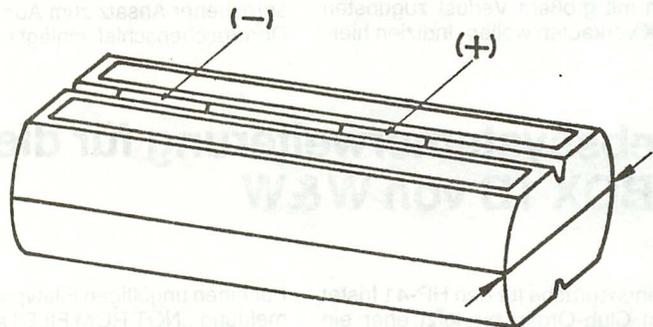


Bild 1 : HP-82033A Akkupaket (z.B. Drucker)

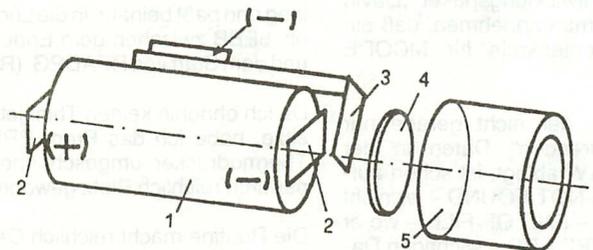


Bild 2 : Zusammenbau der Zelle Nr. 4

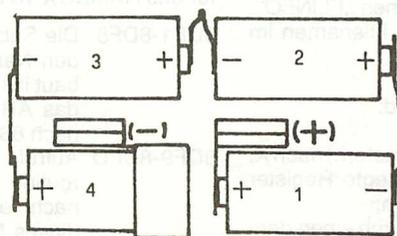


Bild 3 : Anschlußschema aller Zellen

## HP41? – HP41!

Ruhig ist's geworden um das Zugpferd unserer Clubs, den HP41. War vor zwei, drei Jahren die prisma noch voll von enthusiastischen Beiträgen zu den versteckten Fähigkeiten dieses sehr bemerkenswerten Geräts, so sieht es heute immer ruhiger aus. Der HP71 beansprucht – zu Recht – immer mehr Raum in unserer Zeitung. Schließlich gibt es hier wieder viel zu entdecken, aufzuklären, mitzuteilen. Der 71er ist ein Gerät, das man vielleicht eher als „erwachsen“ bezeichnen möchte als den 41er. Aber dennoch haben viele, vielleicht sogar die meisten, ihren HP41. Und um den, wie gesagt, ist's ruhig geworden.

Aber warum denn eigentlich? Im nachfolgenden Text möchte ich einige subjektive Überlegungen zu diesem Thema anstellen und einen Ansatz zum Aufbruch aus dem Dornröschenschlaf vorstellen.

Nun, warum sage ich denn überhaupt „ruhig geworden“? Der 41CX kam doch schließlich vor gar nicht langer Zeit auf den Markt. Mit neuem Platinenlayout und mit diversen Änderungen macht er nicht den Eindruck, die 41er-Serie sei am Ende. Das ist sicher richtig. Aber es stellt sich doch die Frage, ob der 41CX die Szene tatsächlich neu beleben wird; bisher ist jedenfalls nichts derartiges zu beobachten. Die „alten Hasen“ haben halt schon ihren 41C(V) und werden diese auch kaum mit großem Verlust zugunsten eines 41CX verkaufen wollen. Indizien hier-

für finden sich beispielsweise im Kleinanzeigenteil unserer prisma. Und dort zeigt sich vor allem die starke Fluktuation der IL-Geräte. Dies scheint mir keineswegs verwunderlich. Im Zuge des Ausbaus eines 41er-Systems gerät man bald zu der Frage, ob sich die Investition der recht großen Summen für Drucker, Laufwerk und Videointerface anderweitig nicht besser nutzen ließe. Und so stößt vielleicht manch einer seine IL-Komponenten wieder ab und beschränkt seinen 41er damit auf eine Art Nobeltaschenrechner-Dasein.

Nun, einzelne beschreiten einen anderen Weg, den ich im folgenden noch einmal ins Gedächtnis rufen möchte: die Verbindung des 41ers mit anderen Computern.

Die Vorteile liegen auf der Hand: Die Peripheriegeräte sind weitaus billiger und vor allem universeller verwendbar – oder möchte jemand unbedingt seine Korrespondenz via IL-Thermodrucker abwickeln? Leider aber sind die Probleme einer solchen Idee genauso offensichtlich: Es gibt sie nicht fertig zu kaufen, es gibt halt wenig technische Übereinkünfte.

Immer wieder berichteten in der Vergangenheit einzelne in der prisma von Verbindungen zu Sirius, Apple, CP/M, aber es blieb bei diesen Einzelmeldungen, sie wurden nicht ausführlicher diskutiert. Mein oben versprochener Ansatz zum Ausbruch aus dem Dornröschenschlaf schlägt nun ein weitest-

gehend geräteunabhängiges Programmpaket vor, das die Nutzung gewöhnlicher Mikrocomputer durch den HP41 erleichtern soll. Daß solche geräteunabhängigen Konventionen durchaus funktionieren können, dürfte vielen am Beispiel von KERMIT klar geworden sein. Ein solches Programmpaket besteht aus mindestens zwei aufeinander aufbauenden Teilen. Ein geräteabhängiger Kern, das BIOS, möge gewisse Elementarfunktionen zur Verfügung stellen wie Bildschirm- und Ausgabe, Tastaturbedienung, Disk-I/O und die Bedienung der verbindenden Schnittstelle. Darauf aufbauend können dann Programme entwickelt werden, die aufwendigere Probleme behandeln. Angefangen bei schönen DIN A4-Listings auf normalen Matrixdruckern bis hin zur Datenpflege auf den verfügbaren Massenspeichern sind die Möglichkeiten schier unbegrenzt.

Ein solcher Ansatz könnte bei vielen von uns zu neuen Aktivitäten mit dem 41er führen. Deshalb möchte ich Interessierte bitten, mit mir Kontakt aufzunehmen. Für CP/M auf dem alten Apple (Z80H oder Z80A) laufen die ersten BIOS-Komponenten und ich würde mich freuen, mit anderen gemeinsam weiterentwickeln zu können.

Udo Keller  
Kamekestraße 1  
5000 Köln 1  
CCD # 1554

## Betriebssystemerweiterung für die RAMBOX 1B von W&W

### HP41C

Die Maschinensprache für den HP-41 fristet in unserem Club-Organ bis jetzt eher ein Schattendasein.

Mit dem Aufkommen so leistungsfähiger Erweiterungen, wie z.B. die RAMBOX von W & W und das Entwicklungspaket „David Assembler“ sollte man annehmen, daß ein größerer Interessentenkreis für MCODE besteht!

Jedem Anwender, der nicht gerade nur seine „immerwährenden“ Daten in der RAMBOX von W & W ablegt, ist schon aufgefallen, daß – FL NOT FOUND – er nicht mehr genau weiß, – END OF FILE – wo er jetzt seine – FL TYPE ERR – wichtigen Daten, bzw. Tastenzuweisungen in der RAMBOX abgelegt hat, bzw. wieviel in das File noch reingeht.

Abhilfe schafft hier die Funktionen „FLINFO“, die als einzige Eingabe den Filenamen im ALPHA-Register benötigt.

Für einen gültigen Filetyp wird:

1. die Filelänge in (7 Byte Registern) nach X,
2. die Anzahl momentan belegte Register im File nach Y geschrieben;
3. der Filename in ALPHA durch einen dem Filetyp entsprechenden Buchstaben ersetzt.

Für einen ungültigen Filetyp wird die Fehlermeldung „NOT ROM FILE“ ausgegeben. Existiert kein File mit dem entsprechenden Namen, erscheint die Meldung „FILE NOT FOUND“.

Die Routine ist inklusive Namen 101 Worte lang und paßt beinahe in die Lücke von 8E5E bis 8EBB zwischen dem Ende von PRFAT und dem Start von READPG. (RAMBOX 1 B)

Da ich ohnehin keinen Thinkjet Drucker besitze, habe ich das Prgm PRFAT für den Thermodrucker umgeschrieben und habe dadurch reichlich Platz gewonnen.

Die Routine macht reichlich Gebrauch vom übrigen RAMBOX Betriebssystem und dem Betriebssystem des HP-41 selbst.

Die vorliegende Fassung gilt ausschließlich für das RAMBOX 1B Betriebssystem!

8DF1-8DF5 Die Subroutine auf 8690 liest den Namen aus ALPHA und baut in Q und (CPU) C auf. Ist das AR leer, verzweigen wir nach 8638 „NO NAME“

8DF9-8DFD Aufruf der Betriebssystemroutine SAROM (260D), die nach einem XROM-Eintrag dieses Namens sucht. Wenn nichts gefunden, rufen wir 8C95 „FL NOT FOUND“

8E00-8E09 Mit der Adresse von SAROM rufen wir nun nacheinander SAROM+1 = File Typ DTA/KEY/BUF 280/2B0/2C0 SAROM+2 = File Länge SAROM+3 = belegte Register im File ab und speichern die Werte in A, M und N

8E0A-8E16 Durch Vergleiche wird der Filetyp festgestellt. Konnte keine Übereinstimmung festgestellt werden, verzweigen wir nach 8EF8 „NOT ROM FILE“

8E19-8E27 je nach Filetyp wird D, K oder B geladen, mit CLA (10D1) das AR gelöscht und mit APNDNW (2D14) der Buchstabe ins AR gesetzt.

8E29-8E4F Nacheinander wird nun die Anzahl der belegten Register (aus N) und die Filelänge (aus M) in der Subroutine 8E30-8E4F in eine normalisierte BCD-Zahl umgerechnet und unter Berücksichtigung des Zustandes des Stacklifts auf den Stack geschoben. (CPU-Flag 11 ist gelöscht, wenn der Stacklift ausgeschaltet ist.)

Das Prgm wurde mit dem David Assembler geschrieben.

```

FNC FLINFO
8DF1 GOSUB 8690
8DF4 ?C#0 ALL
8DF5 JC +04 8DF9
8DF6 GOTO 8638
8DF9 NC X0 SAROM
8DFB ?C#0 ALL
8DFC JC +04 8E00
8DFD GOTO 8C95
8E00 RCR 11
8E01 C=C+1 M
8E02 FETCH
8E03 A=C S&X
8E04 C=C+1 M
8E05 FETCH
8E06 N=C
8E07 C=C+1 M
8E08 FETCH
8E09 M=C
8E0A LDI
8E0B HEX 200 @
8E0C ?A#C S&X
8E0D JNC +0C 8E19
8E0E LDI
8E0F HEX 200 @
8E10 ?A#C S&X
8E11 JNC +0B 8E1C
8E12 LDI
    
```

```

8E13 HEX 200 @
8E14 ?A#C S&X
8E15 JNC +0A 8E1F
8E16 GOTO 8EF8
8E19 LDI
8E1A HEX 044 D
8E1B JNC +06 8E21
8E1C LDI
8E1D HEX 04B K
8E1E JNC +03 8E21
8E1F LDI
8E20 HEX 042 B
8E21 A=C S&X
8E22 NC X0 CLA
8E24 R= 0
8E25 A<>C S&X
8E26 G=C
8E27 NC X0 APNDNW
8E29 C=M
8E2A A=C S&X
8E2B GOSUB 8E30
8E2E C=N
8E2F A=C S&X
8E30 LDI
8E31 HEX 010 P
8E32 RAM SLCT
8E33 R= 13
8E34 LDOR 4
    
```

```

8E35 A=C MS
8E36 NC X0 GENNUM
8E38 A<>C ALL
8E39 R= 8
8E3A C=0 R<-
8E3B RAM SLCT
8E3C R= 0
8E3D LDOR 3
8E3E LDOR 0
8E3F A<>C ALL
8E40 ?A#0 ER
8E41 JC +04 8E45
8E42 LSHFA M
8E43 A=A-1 S&X
8E44 JNC -04 8E40
8E45 ?A#0 M
8E46 JC +02 8E48
8E47 A=0 ALL
8E48 A<>B ALL
8E49 ?FSET 11
8E4A C X0 R↑SUB
8E4C SETF 11
8E4D C=B ALL
8E4E WRIT 3(X)
8E4F RTN
    
```

Armin Jakob  
 St. Georgen Straße 218  
 CH-9011 St. Gallen

## Mathematische Genauigkeit

HP41

### Alle Rechner – Programmbeispiele für den HP-41

Oft genug kommt es vor, daß eine Berechnung nicht das erhoffte Ergebnis liefert: Eine kleine Ungenauigkeit (Rundungsfehler) taucht auf und bei einem Test (z.B.  $X=0?$ ) oder einer mathematischen Operation (z.B. SWRT) steht eine sehr kleine Zahl wie z.B.  $-3 \text{ E}-10$  die dann zur falschen Verzweigung oder DATA ERROR führt. Der Grund liegt darin, daß der Rechner mit einer begrenzten Stellenzahl (41/15 mit 10, 28/71/75 mit 12 Stellen) arbeitet, während die reine Mathematik solche Begrenzungen nicht kennt. So sieht  $1/3$  in der Mathematik so aus  $0,3 = 0,333 \dots$ , auf dem HP-41 jedoch so:  $0,33333333330000 \dots$ , d.h. die ersten 10 Ziffern hat der 41er im Rechenregister, von den Nachfolgenden nimmt er jedoch an, daß sie alle 0 sind, weil er sie ja nicht speichern kann.

Es ist daher grundsätzlich nicht möglich, Rundungsfehler zu vermeiden, wenn während der Berechnung Dezimalzahlen entstehen, die mehr Stellen haben, als der Rechner sich merken kann. Irrationale Zahlen wie Pi oder periodische Dezimalbrüche wie  $1/22$  werden vom Rechner auf 10 Stellen genau gerundet. Es ist wie mit dem Rauschen in einer HiFi-Anlage, es läßt sich bessern aber es verschwindet nicht. Um Rundungsfehler zu vermindern möchte ich hier ein paar empirische Rezepte vorstellen.

Zunächst ist es wichtig, so lange wie möglich mit ganzen Zahlen zu rechnen (sofern es der Zusammenhang erlaubt). So ergibt 30, ENTER, 3, / genau 10, nicht aber 30, ENTER, 3 1/X, \*; Besonders tückisch sind Rundungsfehler, die nicht erkannt werden, z.B. 3, ENTER, 1/X, \* ist selbst bei FIX 9 als 1 dargestellt (Abhilfe: SCI 9 oder ENG 9).

Ein weiterer wichtiger Punkt ist, so wenig mathematische Operationen wie möglich zu verwenden, was meist das selbe Ergebnis liefert, wie bei der vorangegangenen Empfehlung. Betrachtet man die dort gezeigten Beispiele, so haben die schlechten Beispiele 2 mathematische Operationen (1/X und \*) die guten aber nur eine. Ein kleines Programm (Nr. 1) zeigt, wie man Pi durch eine Zahl dividiert. Version a) ist dabei die schlechte, was man erkennt wenn  $\text{Pi}/5$  vor dem Programmstart eingegeben wird.

Oft muß man jedoch zunächst das Problem mathematisch durchleuchten, um eine Gleichung, Formel oder Term so umstellen zu können, daß er sich gut berechnen läßt. Mathematiker und Naturwissenschaftler beweisen ihren Sinn für Ästhetik oft an solchen Formeln, wie zwei Beispiele zeigen werden. Wachstumsvorgänge werden oft mit der e-Funktion ( $e^x$ , oft auch  $\exp(x)$ ) beschrieben z.B. die Vermehrung von Bakterien oder der radioaktive Zerfall. Die entsprechende Gleichung (Nr. 2) wird durch Äquivalenzumformungen (von links nach rechts) in eine verbesserte Form für eine vereinfachte Berechnung gebracht. Dabei ist  $x$  (wird errechnet)

die verbliebene Menge einer radioaktiven Substanz mit der Halbwertszeit H nach der Zeit t, wenn am Anfang die Menge a vorhanden war. Für einige Härtefälle (z.B.  $a = 32$ ,  $H = 3$ ,  $t = 12$  ergibt  $x = 2$ ) zeigt das Programm b, daß es nicht nur schneller und kürzer, sondern auch genauer sein kann. Ein weiteres Beispiel ist der Areatangens (artanh, Umkehrfunktion zu Tangens hyperbolicus). Die Version b) in Nr. 3 liefert auch zwischen  $-1 \text{ E}-10$  und  $+1 \text{ E}-10$  noch recht genaue Ergebnisse, während Version a) hier nur 0 bieten kann.

Hier zeigt sich auch, das man die vom Rechner gebotenen Funktionen so weit wie möglich ausnutzen muß um dem Gebot, wenige mathematische Funktionen zu benutzen, nachzukommen.

Um zu möglichst genauen Ergebnissen zu kommen, muß man sich überlegen, wie aufwendig die verwendeten math. Op. sind: Harmlos sind ABS, CHS oder X( )Y (Austausch X mit Y), weil sie nichts an den Dezimalstellen einer Zahl ändern. Auch die Division/Multiplikation mit 1 Exy verändert die Mantisse nicht, wie die verschiedenen Beispiele (Nr. 4) zeigen – die jeweils bessere Variante ist unterstrichen. Operationen der „2. Art“ sind z.B. 1/X, SQRT, + oder / und die der unheimlichen „3. Art“ sind ln, log,  $10^x$ ,  $y^x$ , LN1+X, SIN u.s.w. Diese Hierarchie ist wichtig, wenn man verschiedene Darstellungen für eine Formel gefunden hat und nun bestimmen will, welche die günstigste ist, weil in ihr die Operationen der unteren

Ordnungen überwiegen – auch hier gibt Nr. 4 entsprechende Beispiele, da sich in bestimmten Fällen (im Zusammenhang mit Logarithmus- und Exponentialfunktion) Operationen der 2. Art durch solche der 1. ersetzen lassen.

Ein anderes Beispiel bietet Nr. 5; Auf ein Polynom wird das Horner Schema angewendet, weshalb das Programmbeispiel b) schneller und genauer ist, denn hier wurden Operationen der 3. Art durch solche der 2. Art ersetzt. Hier spielt außerdem die UPN mit ihren 4 Stackregistern (nicht HP-28C) ihre Vorteile gegenüber einer algebraischen Notation aus, weil die UPN ja die Klammern aus der mathematischen Darstellung nicht braucht.

Das man es mit den Äquivalenzumformungen nicht wild treiben sollte, zeigt das folgende Beispiel:  $\ln(a^b) = b \cdot \ln a$ , so steht es in jeder Formelsammlung, und für viele Fälle stimmt das so, z.B.  $\ln(x^3) = 3 \ln x$ ; Ist b jedoch eine gerade Zahl, z.B.  $b = 2$  so gilt:  $\ln(x^2) = 2 \cdot \ln|x|$ , weil das Quadrat die negativen Vorzeichen beseitigt. Im HP-28C ist z.B.  $\ln(-2)^2 = 1,386 \dots$  jedoch  $2 \cdot \ln(-2) = 1,386 \dots + i6,283$ , also nicht reell. Ebenfalls bekannt ist die Regel  $\log a - \log b = \log(a/b)$ . Wenn sowohl a als auch b negativ sind, z.B.  $\log(-60) - \log(-6)$ , dann können nur Rechner mit komplexer Arithmetik die Lösung 1 finden. Weder der 15C noch der 28C kommen da auf **genau** 1. Hingegen wird  $\log((-60)/(-6))$  von allen HP-Modellen einwandfrei bewältigt. Bei Umformungen muß man also den Definitionsbereich genau vergleichen, sonst bearbeitet der Rechner hinterher Funktionen an Stellen, die eigentlich gar nicht definiert waren, oder erweigert sich, ursprünglich definierte Stellen auszurechnen; Bei Nr. 3 hat es keine Veränderung des Definitionsbereiches gegeben.

Nr. 6 zeigt, wie abenteuerlich sich Funktionen verhalten können, und daß eine genaue Untersuchung im Einzelfall unerlässlich ist. Wird  $x = 5$  gewählt, so erweisen sich die beiden ersten Terme als ungenau, weil sie das Ergebnis  $1/30$  knapp verfehlen. Probiert man es jedoch mit  $x = 1/8192$ , so lautet das richtige Ergebnis 8191,000122 (Rechner mit 12-stelliger Genauigkeit verwenden  $x = 1/65536$ , Ergebnis 65535,0000153), so findet nur der erste Term die richtige Lösung. Hier muß man also beachten, mit welchen x-werten häufiger gerechnet wird, um sich dann für den ersten oder letzten Term zu entscheiden.

Das Problem das hier die Subtraktion bringt zeigt dieses Beispiel:  $1/0,03 - 30$  zeigt, daß die erste Stelle verschwindet und hinten eine 0, statt der richtigen 3, nachrückt, und der Rechner nur noch mit einer auf 9 Stellen genauen Zahl arbeitet – es läßt sich leicht zeigen, daß das Beispiel  $10/3$  als Ergebnis hat. Diese Probleme sind immer dann zu erwarten, wenn der Inhalt des X und Y-Registers die gleiche Größenordnung (= Zehnerexponent, EEX) haben, und treten sinngemäß auch bei Additionen mit unterschiedlichen Vorzeichen auf.

Als letztes Beispiel zeigt Nr. 7, daß Version b) die Beste ist, sofern der Rechner über eine LN1 + X Funktion verfügt. Andernfalls hat man mit allen 4 Darstellungen Probleme, wenn x sehr groß wird, und falls x größer

1 E 10 wird, gibt jede Darstellung 0 heraus. Für  $x = 1000$  zeigt übrigens die letzte Darstellung durch die Subtraktion einen Verlust von 3 signifikanten (= für die Berechnung wichtigen) Dezimalstellen.

Unter Nr. 8 werden noch einmal alle Anwendungsfälle für die Funktion LN1+X gezeigt, und im übrigen wünsche ich viel Spaß und higher Precision.

- 1) LBL A, 1/X, PI, \* (schlecht)  
 LBL B, PI, X()Y, / (gut)

$$2) x = a \cdot e^{-\left(\frac{\ln 2}{H}\right) \cdot t} = a \cdot \underbrace{(e^{\ln 2})}_{= 2}^{-t/H} = a \cdot 2^{-t/H} = \frac{a}{2^{t/H}}$$

Stackinhalte: X = H; Y = t; Z = a.

- LBL A, /, 2, LN, \*, CHS, E↑X, \* (schlecht)  
 LBL B, /, 2, X()Y, y<sup>X</sup>, / (gut)

$$3) \ln \sqrt{\frac{1+x}{1-x}} = \operatorname{artanh} x = \frac{\ln(1+x) - \ln(1-x)}{2}$$

- LBL A, 1, X()Y, +, 1, LASTX, -, /, SQRT, LN  
 LBL B, LN1+X, LASTX, CHS, LN1+X, -, 2, / (gut)

$$4) \ln\left(\frac{1}{x}\right) = -\ln x; \log\left(\frac{1}{x}\right) = -\log x; \frac{1}{x^3} = x^{-3};$$

$$\frac{1}{e^{2x}} = e^{-2x}; \log\left(\frac{x}{100}\right) = \log x - 2; \frac{e^{a \cdot x}}{\dots} = (e^a)^x;$$

$$5) 3x^3 - 5x^2 + 2x - 6 = x(x(3x - 5) + 2) - 6$$

- LBL A, ENTER, ENTER, ENTER, 3, y<sup>X</sup>, LASTX, \*, X()Y, X↑2, 5, \*, -, +, +, 6, - (schlecht)  
 LBL B, ENTER, ENTER, ENTER, 3, \*, 5, -, \*, 2, +, \*, 6, -

$$6) \frac{1}{x} - \frac{1}{x+1} = \frac{1}{x} \cdot \frac{1}{x+1} = \frac{1}{x(x+1)} = \frac{1}{x^2 + x}$$

- LBL A, 1/X, LASTX, 1, +, 1/X, -  
 LBL B, 1/X, LASTX, 1, +, 1/X, \*  
 LBL C, ENTER, X↑2, +, 1/X

$$7) \ln\left(\frac{1}{1 + \frac{1}{x}}\right) = -\ln\left(1 + \frac{1}{x}\right) = \ln\left(\frac{x}{x+1}\right) = \ln x - \ln(x+1)$$

- LBL A, 1/X, 1, +, 1/X, LN (schlecht)  
 LBL B, 1/X, LN1+X, CHS (gut)  
 (LBL B, 1/X, 1, +, LN, CHS — ohne LN1+X)

$$8) \ln(1+x) = \ln(x+1) : \text{LN1+X};$$

$$\ln(1-x) : \text{CHS, LN1+X}; \ln(x-1) : 1, -, \text{LN}$$

Ralf Pfeifer  
 Rubensstraße 5  
 5000 Köln 50

# Die Barcodes des HP41



## Barcodes, was ist das

– 1. Teil –

Um was es sich bei Barcodes im Prinzip handelt, das wissen wohl inzwischen die meisten von uns. Spätestens die komischen Strichfolgen auf den meisten Lebensmittelpackungen haben dieses Speichermedium unter Volk gebracht. Das darin Information gespeichert ist, das wissen nur noch einige wenige, für die meisten bleibt es nach wie vor eine rätselhafte Hieroglyphe mit Verwandtschaft zur ägyptischen Bilderschrift, zumindestens ebenso geheimnisvoll.

„Bar“-codes sind zu Deutsch ganz einfach Strich- oder „Balken“-codes, für kleine Datenmengen sind sie das billigste und sicherste Speichermedium, da sich fast jeder Untergrund wie z.B. Papier als Träger eignet. Für die sogenannten „Handhelds“, d.h. tragbaren Taschenrechner wie den HP 41 oder HP71 und HP75 wurden deshalb Barcodeleser entwickelt, da die Programme oder Befehlsfolgen dieser Rechner aus relativ wenig Daten bestehen. Dies trifft allerdings beim HP71 und HP75 nur noch bedingt zu.

Beim HP 41 lohnt sich die Anwendung von Barcodes wesentlich eher, die Programme sind nicht so sehr lang, wie wir am Ende jedes PRISMA sehen können. Ich möchte mit einer Artikelserie die verschiedenen Barcodearten des PH41 erläutern, sodaß nachher jeder in der Lage sein sollte selbst mit Hilfe eines egal wie gearteten Rechners mit Ausdruckgerät Barcodes herzustellen.

Genug der langen Vorrede, kommen wir zu den Tatsachen:

Der „-WAND 1F-“ oder die etwas ältere Version „-WAND 1E-“, wie sie im CATALOG 2 im Display des HP41 zu erkennen sind, bestehen aus der eigentlichen Leseinheit, dem Griffel mit dem Drucktaster zum Einschalten, und dem Stecker, in dem eine intelligente Elektronik zur Dekodierung und Aufbereitung der verschiedenen Barcodearten enthalten ist.

Der HP41-Barcodeleser kennt folgende Barcodearten:

- 1) Programm-Barcode
- 2) Direktor Ausführungs-Barcode (komplette Tastenfolgen!)
- 3) Daten-Barcode
- 4) „Tasten“-Barcode (die Tastenfunktionen des HP41)

Der 4. und letzte Barcodeart ist zugleich auch der einfachste, ich möchte deshalb auch mit diesem anfangen, zuvor müssen aber noch einige Grundlagen geklärt werden:

### 1. Barcode-Spezifikation

Der HP41 Barcode ist relativ einfach aufgebaut, er besteht lediglich aus dicken und dünnen Strichen, die in konstantem Abstand von etwa der Dicke eines dünnen Striches aufeinander folgen; ein dicker Strich ist doppelt so dick wie ein dünner.

Jeder HP41-Barcode beginnt mit zwei dünnen Strichen und endet mit einem dicken und dünnen Strich (dies dient der Erkennung der Leserichtung beim Abtasten des Barcodes, der Leser erkennt damit automatisch die Leserichtung vorwärts oder rückwärts). Dazwischen befinden sich 1-16 Byte Barcodearten, wobei immer ein dünner Strich eine „0“ und ein dicker Strich eine „1“ darstellt, man kann den Barcode also auch normal mit bloßem Auge lesen.

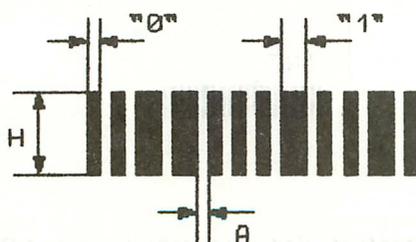


Bild 1: Physikalische Grenzen des HP41 Barcodes

### Empfohlene Werte:

|                 |     |                     |
|-----------------|-----|---------------------|
| Barcodehöhe     | H   | > 8.9 mm            |
| Barcodeabstände | A   | = 0.56 mm ± 0.08 mm |
| Barcode         | „0“ | = 0.46 mm ± 0.08 mm |
| Barcode         | „1“ | = 0.96 mm ± 0.08 mm |

Diese Werte können in einem relativ weiten Bereich nach oben und unten hin variiert werden, solange sie relativ zueinander dieselben Dimensionen behalten, d.h. man kann A, „0“, „1“ doppelt so groß machen aber immer alle Größen gemeinsam!

### 2. Barcodeleser Spezifikation

Auch für den Barcodeleser gibt es Grenzen, sie seien am folgenden Bild kurz erläutert:

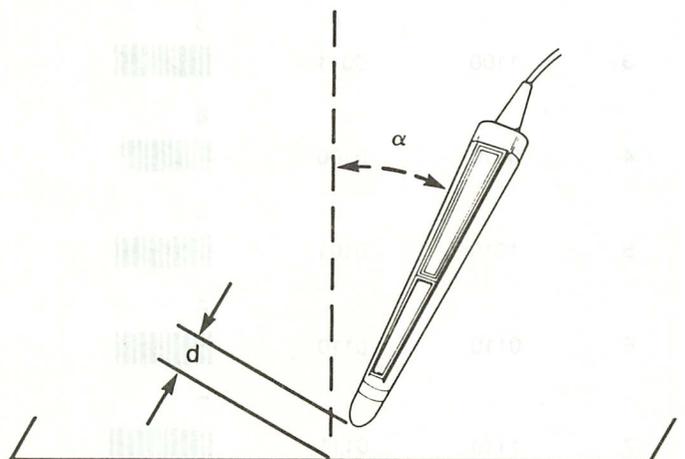


Bild 2: Barcodelesegeometriegrenzen

Der Abstand zwischen Papier (mit dem Barcode) und dem Leserkopf „d“ sollte kleiner oder gleich 0.5 mm sein, wobei der Leserkopf Berührung mit dem Papier haben sollte.

Der Neigungswinkel des Lesers „α“ sollte nicht größer als 30° sein, empfehlenswert ist ein Winkel zwischen 10° und 20°.

### 3. Barcodequalität

Der Kontrast des Barcodes sollte so groß wie irgend möglich sein, d.h. der Unterschied zwischen der Reflektion der Balken und der Zwischenräume. Der Barcodeleser arbeitet auf einer Wellenlänge von 700 nm, man kann also getrost schwarze Balken auf rotem Papier verwenden; dies ist übrigens der beste Kopierschutz, da die meisten Kopierer rot als schwarz kopieren, der Barcode also dabei unleserlich wird.

Vermeiden sollte man ebenso Dreck (kleine Punkte) zwischen den einzelnen Balken. Der Leser toleriert eine ganze Menge, aber irgendwann ist aber auch ein HP-Barcodeleser am Ende!

Die Fehlertoleranz des Barcodelesers ist übrigens wegen der fast direkten Speisung der Leseoptik mit der Batteriespannung von dieser abhängig, aber anders als man vermuten würde:

Je schwächer die Batterien oder Akkus, desto besser toleriert der Barcodeleser sogenannte „Spots“, also kleine Dreckpunkte im Barcode.

So, genug der trockenen Theorie, kommen wir zu den versprochenen „Tasten-Barcodes“. Dies sind Barcodes, die genau eine Taste des HP41 darstellen, liest man also einen solchen Barcode, so geschieht genau dasselbe, als hätte man eine Taste gedrückt, Ihr werdet ja sehen.

Dieser Barcodetyp besteht aus 1, maximal aus 2 Byte, wir beginnen natürlich der Einfachheit halber mit den

**1-Byte-Tastenbarcodes**

Aufbau ---->>>> aaaa bbbb

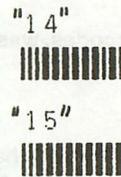
aaaa = oberes Nibble (Halbbyte), Spiegelung von bbbb  
 bbbb = unteres Nibble

| Funktion | aaaa | bbbb |     |
|----------|------|------|-----|
|          |      |      | 0   |
| 0        | 0000 | 0000 |     |
|          |      |      | 1   |
| 1        | 1000 | 0001 |     |
|          |      |      | 2   |
| 2        | 0100 | 0010 |     |
|          |      |      | 3   |
| 3        | 1100 | 0011 |     |
|          |      |      | 4   |
| 4        | 0010 | 0100 |     |
|          |      |      | 5   |
| 5        | 1010 | 0101 |     |
|          |      |      | 6   |
| 6        | 0110 | 0110 |     |
|          |      |      | 7   |
| 7        | 1110 | 0111 |     |
|          |      |      | 8   |
| 8        | 0001 | 1000 |     |
|          |      |      | 9   |
| 9        | 1001 | 1001 |     |
|          |      |      | .   |
|          | 0101 | 1010 |     |
| EEX      | 1101 | 1011 |     |
|          |      |      | CHS |
| CHS      | 0011 | 1100 |     |
|          |      |      | < - |
| <-       | 1011 | 1101 |     |

(Ob die Barcodes nach dem Satz noch lesbar sein werden kann ich nicht versprechen, sie wurden auf einem QUIETJET in Double Density Graphik = 192\*96 dpi [dots per inch] ausgedruckt, mit den heute erhältlichen Nadeldruckern dürfte die Qualität noch ein wenig besser sein, Gehörschutz nicht vergessen!)

Wie man leicht erkennen kann, ist das obere Nibble, „aaaa“ nur die Spiegelung des unteren Nibbles „bbbb“. Ich konnte bislang keine mathematische Herleitung zur Berechnung dieses Barcodes ergründen, HP gibt auch keine an, vielleicht findet sich ja jemand mit einer Formel, da Bit-Spiegelungen wohl in den wenigsten Programmiersprachen zu finden sein werden. Ansonsten hilft eben nur die gute alte Wertetabelle.

Die verbleibenden 2 Möglichkeiten dieses einfachsten Barcodetyps sind offiziell nicht definiert. Probiert man sie trotzdem aus, so führt der Rechner lediglich die Taste R/S aus:



So, nun kommen wir zu einem etwas schwierigeren Kapitel, den

**2-Bytes-Tastenbarcodes**

Dieser Typ besteht, wie schon sein Name sagt, aus genau 2 Byte, die je zwei Start und Stopbits nicht zu vergessen. Dieser Barcodetyp gliedert sich in folgende 5 Klassen:

1. Programmierbare Grundfunktionen des HP41
2. Buchstabentasten
3. Die Taste INDirekt, d.h. wenn man STO drückt und dann die SHIFT-Taste
4. Nichtprogrammierbare Funktionen des HP41 wie z.B. CAT oder PACK
5. XROM, d.h. alle Funktionen, die nicht zum Standardbefehlssatz des HP41 gehören, also auch die zusätzlichen X-Funktionen und TIME Funktionen!

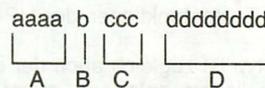
Wir fangen natürlich gleich mit dem schlimmsten an:

**1. Programmierbare Grundfunktionen des HP41**

Unter diesen Oberbegriff fallen alle Funktionen des HP41, die man in seiner Funktionstabelle an folgenden Stellen findet:

- Reihe 4, Spalte 0 bis Reihe 9, Spalte 15;
- Reihe 10, Spalten 8 bis 14;
- Reihe 12, Spalten 0, 14 und 15;
- Reihe 13, Spalte 0;
- Reihe 14, Spalte 0.

Das Format dieses Barcodes ist wie folgt:



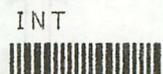
- A = 4-Bit Prüfsumme
- B = 0
- C = 000
- D = 8-Bit Funktionscode

Was für ein Funktionscode, werden die meisten jetzt wohl fragen, die ist aber ganz einfach:

Man nimmt z.B. von der Funktion INT die Adresse der Reihe, die ist 6, macht binär 0110, setzt die Adresse der Spalte, die ist 8, binär 1000, einfach dahinter, schon ist das Puzzle fertig:

Funktionscode von INT = 0110 1000<sub>2</sub> (68<sub>16</sub> oder 104)

Zur Probe wollen wir uns natürlich gleich den Barcode dazu anschauen:



Die nächste Frage folgt sofort, wie errechnet man die 4-Bit Prüfsumme, dies ist aber auch nicht schwer, hier der Algorithmus dafür:

Prinzipiell ist die 4-Bit Prüfsumme nichts anderes als die Summe der restlichen 3 4-Bit Gruppen des Barcodes. Da aber in einer 4-Bit Prüfsumme maximal die Zahl 15 unterzubringen ist, das wäre  $1111_2$ , wird beim Aufaddieren der Nybbles, so heißen diese sogenannten „Halbbytes“, ganz einfach der Übertrag (Carry) zur jeweiligen Summe dazuaddiert.

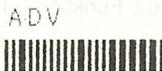
Bei „INT“ geht das dann wie folgt:

|             |   |     |                            |
|-------------|---|-----|----------------------------|
| (B und C)   | = | 0   | 0000                       |
| dddd (high) | = | + 6 | 0110                       |
| dddd (low)  | = | + 8 | 1000                       |
|             | = | 14  | 1110 (kein Carry/Überlauf) |

So, dasselbe noch einmal mit der Funktion ADV:

|                   |     |       |                  |
|-------------------|-----|-------|------------------|
| (B und C)         | =   | 0     | 0000             |
| Reihe dddd (high) | =   | + 8   | 1000             |
| Spalte dddd (low) | =   | +15   | 1111             |
|                   | =   | 7 (1) | 0111 (Carry = 1) |
|                   | + 1 | +     | 1                |
|                   | 8   |       | 1000             |

Der Barcode sieht dann folgendermaßen aus:



Bevor wir mit dem nächsten Tastenbarcodetyp weiter machen noch ein kleiner Hinweis für Selberrmacher:

- a) Bei GTO nn muß aus der Tabelle die Position 13,0 gewählt werden
- b) Hier gilt dasselbe für XEQ nn, Position 14,0 wählen
- c) Reihe 12, Spalte 0 ergab bei mir nichts brauchbares

## 2. Buchstabentasten

der prinzipielle Aufbau dieses Barcodetyps ist derselbe wie bei den Tastenfunktionen:

- A = 4-Bit Prüfsumme
- B = 0
- C = 001 (1)
- D = modifizierter Buchstabencode, 8 Bit

Dieser eben erwähnte modifizierte Buchstabencode wird wie folgt ermittelt:

Alle Buchstaben werden in der ASCII-Tabelle (American Standard Code of Information Interchange) definiert, beim HP41 gibt es da die Ausnahmen wie zum Beispiel das Winkelzeichen oder das Zeichen für den Buchstabencode 0, die Raute. Man nimmt nun den Binärcode für das zu drückende Zeichen, verdoppelt dessen obere 4 Bit und erhält das Byte D des Barcodes. Da die Theorie immer grau ist will ich sofort ein Beispiel zeigen:

„q“ ist laut ASCII-Tabelle (findet sich in jedem Druckerhandbuch) das Zeichen Nummer 64, macht binär  $0100\ 0000_2$ .

Die oberen 4 Bit sind 4, verdoppelt man diese ergibt sich 8, macht binär  $1000_2$ .

Der modifizierte Code lautet also  $1000\ 0000_2$ .

Für alle, die die erwähnte Tabelle noch nicht kennen oder ihr Eigen nennen, sei sie noch einmal abgedruckt:

|    | 0                                            | 1              | 2       | 3              | 4      | 5              | 6        | 7               | 8                  | 9       | 10      | 11       | 12                | 13      | 14          | 15      |
|----|----------------------------------------------|----------------|---------|----------------|--------|----------------|----------|-----------------|--------------------|---------|---------|----------|-------------------|---------|-------------|---------|
| 0  | NULL                                         | LBL 00         | LBL 01  | LBL 02         | LBL 03 | LBL 04         | LBL 05   | LBL 06          | LBL 07             | LBL 08  | LBL 09  | LBL 10   | LBL 11            | LBL 12  | LBL 13      | LBL 14  |
| 1  | digit 0                                      | 1              | 2       | 3              | 4      | 5              | 6        | 7               | 8                  | 9       | .       | EEX      | (digit entry) CHS | GTO α   | XEQ α       |         |
| 2  | RCL 00                                       | RCL 01         | RCL 02  | RCL 03         | RCL 04 | RCL 05         | RCL 06   | RCL 07          | RCL 08             | RCL 09  | RCL 10  | RCL 11   | RCL 12            | RCL 13  | RCL 14      | RCL 15  |
| 3  | STO 00                                       | STO 01         | STO 02  | STO 03         | STO 04 | STO 05         | STO 06   | STO 07          | STO 08             | STO 09  | STO 10  | STO 11   | STO 12            | STO 13  | STO 14      | STO 15  |
| 4  | +                                            | -              | .       | /              | X < Y? | X > Y?         | X < = Y? | Σ +             | Σ -                | HMS +   | HMS -   | MOD      | %                 | %CH     | P - R       | R - P   |
| 5  | LN                                           | x <sup>2</sup> | SQRT    | y <sup>x</sup> | CHS    | e <sup>x</sup> | LOG      | 10 <sup>x</sup> | e <sup>x</sup> - 1 | SIN     | COS     | TAN      | ASIN              | ACOS    | ATAN        | DEC     |
| 6  | 1/X                                          | ABS            | FACT    | X ≠ 0?         | X > 0? | LN(1 + X)      | X < 0?   | X = 0?          | INT                | FRAC    | D - R   | R - D    | HMS               | HR      | RND         | OCT     |
| 7  | CL                                           | X < > Y        | PI      | CLST           | RI     | RDN            | LASTX    | CLX             | X = Y?             | X ≠ Y?  | SIGN    | X < = 0? | MEAN              | SDEV    | AVIEW       | CLD     |
| 8  | DEG                                          | RAD            | GRAD    | ENTERI         | STOP   | RTN            | BEEP     | CLA             | ASHF               | PSE     | CLRG    | AOFF     | AON               | OFF     | PROMPT      | ADV     |
| 9  | RCL nn                                       | STO nn         | ST + nn | ST - nn        | ST' nn | ST/ nn         | ISG nn   | DSE nn          | VIEW nn            | ΣREG nn | ASTO nn | ARCL nn  | FIX n             | SCI n   | ENG n       | STONE n |
| 10 | XROM                                         | XROM           | XROM    | XROM           | XROM   | XROM           | XROM     | XROM            | SF nn              | CF nn   | F?C nn  | FC?C nn  | FS? nn            | FC? nn  | GTO/XEQ IND |         |
| 11 |                                              | GTO 00         | GTO 01  | GTO 02         | GTO 03 | GTO 04         | GTO 05   | GTO 06          | GTO 07             | GTO 08  | GTO 09  | GTO 10   | GTO 11            | GTO 12  | GTO 13      | GTO 14  |
| 12 | ----- ALPHA LABEL AND END INSTRUCTIONS ----- |                |         |                |        |                |          |                 |                    |         |         |          |                   |         |             |         |
| 13 | ----- GTO nn -----                           |                |         |                |        |                |          |                 |                    |         |         |          |                   |         |             |         |
| 14 | ----- XEQ nn -----                           |                |         |                |        |                |          |                 |                    |         |         |          |                   |         |             |         |
| 15 |                                              | TEXT 1         | TEXT 2  | TEXT 3         | TEXT 4 | TEXT 5         | TEXT 6   | TEXT 7          | TEXT 8             | TEXT 9  | TEXT 10 | TEXT 11  | TEXT 12           | TEXT 13 | TEXT 14     | TEXT 15 |

HP41 Funktion Table (Quelle: Hewlett Packard)

Berechnung der Prüfsumme:

|             |     |   |                            |
|-------------|-----|---|----------------------------|
| (B und C)   | =   | 1 | 0001                       |
| dddd (high) | = + | 8 | 1000                       |
| dddd (low)  | = + | 0 | 0000                       |
|             | = + | 9 | 1110 (kein Carry/Überlauf) |

Der Barcode sieht also so aus:



Gleich noch ein zweites Beispiel für die Wissensdurstigen, diesmal nehmen wir uns das Zeichen „N“ vor:

Der ASCII-Code ist 78, binär 0100 1110<sub>2</sub>. Durch Verdoppelung der oberen 4 Bit erhält man 1000 1110<sub>2</sub>.

Berechnung der Prüfsumme:

|             |     |       |                |
|-------------|-----|-------|----------------|
| (B und C)   | =   | 1     | 0001           |
| dddd (high) | = + | 8     | 1000           |
| dddd (low)  | = + | 14    | 1110           |
|             | = + | 7 (1) | 0111 Carry = 1 |
|             | + 1 | +     | 1              |
|             | 8   |       | 1000           |

Der Barcode sieht also so aus:



| Funktion | Dezimalcode |        |
|----------|-------------|--------|
| CAT      | 0           |        |
| GTOL     | 1           | (GTO.) |
| DEL      | 2           |        |
| COPY     | 3           |        |
| CLP      | 4           |        |
| R/S      | 5           |        |
| SIZE     | 6           |        |
| BST      | 7           |        |
| SST      | 8           |        |
| ON       | 9           |        |
| PACK     | 10          |        |
| DELETE   | 11          | (<-)   |
| ALPHA    | 12          |        |
| PRGM     | 13          |        |
| USER     | 14          |        |
| ASN      | 15          |        |

GTOL heißt **GTO** Line, das ist dasselbe, als wenn man GTO und dann die Dezimalpunktstaste drückt.

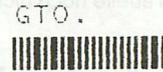
DELETE ist die Funktion der Backarrow-Taste, des Pfeils nach links zum Löschen der letzten Eingabe.

Unser Beispiel ist diesmal die Funktion GTOL oder GTO.:

Berechnung der Prüfsumme:

|             |     |   |                            |
|-------------|-----|---|----------------------------|
| (B und C)   | =   | 4 | 0100                       |
| dddd (high) | = + | 0 | 0000                       |
| dddd (low)  | = + | 1 | 0001                       |
|             | =   | 5 | 0101 (kein Carry/Überlauf) |

Der Barcode sieht also so aus:



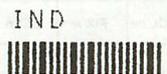
**3. Die Taste INDirekt**

Bei einigen Funktionen wie STO, RCL oder auch ISG gibt es indirekte Adressierung; was dies ist möchte ich an dieser Stelle nicht noch einmal erläutern, dazu kann jeder sein Rechnerhandbuch zu Rate ziehen. Auf der Tastatur erreicht man diese Adressierungsart durch Drücken der gewünschten Funktion (STO/RCL etc.) und Drücken der gelben SHIFT-Taste, probiert es einmal.

Die Definition sieht folgendermaßen aus:

- A = 1010 (10)
- B = 0
- C = 010 (2)
- D = 1000 0000 (128)

Die Prüfsumme brauche ich ja nicht herzuleiten, sie ist ja festgelegt, also gleich der Barcode zum Anschauen:



**4. Nicht programmierbare Funktionen des HP 41**

Zu diesen Funktionen gehören zum Beispiel CAT oder DEL, gibt man sie im PRGM-Modus ein, so werden sie ausgeführt und nicht als Programmschritt akzeptiert.

Für dieses Barcodetyp gilt folgende Definition:

- A = 4-Bit Prüfsumme
- B = 0
- C = 100 (4)
- D = Dezimalcode der folgenden Funktionstabelle

**5. XROM**

Mit dieser Funktion kann jede Modulfunktion des HP41 erreicht werden, man findet die zur Funktion gehörende XROM-Nummer meistens am Ende der Handbücher der entsprechenden Module. Der Barcodeleser ist zum Beispiel auch eines dieser Einsteckmodule, bevor wir uns aber eine seiner Funktionen als Barcode anschauen noch ein kleiner Schreck vorweg, die Barcodeerzeugung für XROM-Funktionen ist etwas kompliziert, davon sollte sich aber niemand abschrecken lassen!

- A = 4-Bit Prüfsumme
- B = 1
- C = f f f
- D = g g h h

fffgg ist die sogenannte ROM-ID, die Identifizierungsnummer eines jeden Moduls. Da diese nur 5 Bit lang ist kann es auch nur maximal 32 verschiedene geben; Einige werden dies schon gemerkt haben, man kann deshalb manche Module nicht gemeinsam im Rechner haben.

hhhhh ist die fortlaufende Funktionsnummer im Modul, der 6 Bit wegen kann es also maximal 64 Funktionen pro Modul geben.

Da wir uns schon mit Barcodes beschäftigen, warum also nicht eine Funktion des Barcodelesers, ich nehme die Funktion WNDSCN. Aus dem Handbuch entnimmt man, das diese Funktion XROM 27.05 heißt.

Die ROM-ID ist also 27, die Funktionsnummer 5:

Berechnung der Prüfsumme:

$$\begin{array}{r}
 \text{(B und C)} = 14 \quad 1110 \\
 \text{gghh (high)} = +12 \quad 1100 \\
 \hline
 = 10 \quad (1) \quad 1010 \quad (\text{Carry} = 1) \\
 \quad + 1 \quad + \quad 1 \\
 \hline
 \quad 11 \quad + \quad 1011 \\
 \text{hhhh (low)} = + 5 \quad 0101 \\
 \hline
 = 0 \quad (1) \quad 0000 \quad (\text{Carry} = 1) \\
 \quad + 1 \quad + \quad 1 \\
 \hline
 \quad 1 \quad 0001
 \end{array}$$

Barcode = pppp 1fff gghh hhhh  
 0001 1110 1100 0101

Der Barcode sieht also so aus:



## TABellen-Editor

156 Zeilen, 302 Bytes, 43 Regs., SIZE 5 + Anzahl Spalten, FIX 5, HP41CV, X-F/M, IL, (HP 2225-IL)

Das beiliegende Programm **TAB** (TABellen-Editor) war mir schon öfters eine große Hilfe. Wie sein Name besagt, erlaubt es die formatierte (!) Eingabe beliebiger alphanumerischer Tabellen mit bis zu 132 Zeichen Breite. Das Programm erkennt selbstständig, wenn die Standardbreite von 80 Zeichen überschritten wird und fragt, ob es auf den compressed mode des Druckers umschalten soll. Der Ausdruck erfolgt auf einen HP2225 ThinkJet-Printer, ist aber leicht an jeden 80 Zeichenspalten-Drucker anzupassen.

Weiteres steht im beigelegten Text.

### Programmbeschreibung, Gleichungen, Variable

Das Programm bietet eine einfach zu benutzende, und nur durch die Tastatur des HP 41 und des verwendeten Druckers begrenzte, Möglichkeiten zur Eingabe beliebiger alphanumerischer Tabellen. Die Ausgabe dieser Tabellen erfolgt Zeilenweise auf einen HP2225 ThinkJet Drucker oder einen beliebigen anderen 80 Zeichen IL-Drucker.

Es können bis zu 132 Zeichen je Zeile eingegeben werden, wobei das Programm selbständig erkennt, wenn 80 Zeichen überschritten werden. In diesem Fall wird der Benutzer gefragt, ob er die Tabelle im compressed mode drucken möchte, der bei einer positiven Antwort selbständig eingestellt wird. Bei mehr als 132 Zeichen muß das Programm mit neuer Spaltenbreite erneut gestartet oder aber die Tabelle umorganisiert werden.

Der benötigte Speicherplatz wird überprüft und nötigenfalls ange-mahnt.

Bei Anpassung an einen anderen Drucker als den ThinkJet müssen die beiden folgenden Escape-Sequenzen geändert werden:

**Zeile 61:** Unterstreichen an: ESC & d D (dec: 27,38,100, 68)  
**Zeile 151:** compressed mode:ESC & k 2 S (dec: 27,38,107,50,83)

Beide Sequenzen werden je einmal an den Drucker übergeben.

Das beiliegende Listing wurde mit dem ThinkJet im Display-mode gedruckt, so daß alle Drucker-Control-Zeichen als kleine Buchstaben mit ausgedruckt wurden. Da der ThinkJet weder „,“ noch „≠“ drucken kann, wurden einige Zeilen des Listings per Hand berich-tigt.

Bei diesem Barcodetyp wünsche ich allen Programmierern beson-ders viel Spaß, das Zerlegen und wieder Zusammensetzen der Bits ist wirklich eine wahre Freude, ich hatte ja selbst schon dieses schö-ne Vergnügen.

Sollte jemand Schwierigkeiten haben selbst lesbare Barcodes zu drucken, so darf er nicht zu schnell aufgeben, man muß das Verhält-nis der dünnen Striche und des Abstandes, er sollte etwa gleich sein, doch ein paar Mal ausprobieren, nicht die Geduld verlieren!

In der nächsten Folge kommen dann die Programmbarcodes zum Zuge, auch dies ein erquickendes Vergnügen für Programmierer, bis dahin viel Spaß beim Programmieren.

Quelle:  
 Hewlett Packard  
 „Creating Your Own HP-41 Bar Code“

Happy Programming,

Martin Meyer (1000)  
 Redaktion

Achtung: Der Drucker wird bei Programmstart nicht zurückgesetzt! Führende Leerzeichen einer Spalte müssen per Hand eingegeben werden, nachfolgende Leerzeichen werden vom Programm einge-setzt.

Bei Verwendung eines anderen Druckers als dem ThinkJet müssen die beiden oben aufgeführten Escape-Sequenzen angepaßt wer-den.

Ein eventuell verwendetes Spaltentrennzeichen muß > = 32 (dec) sein.

Literatur: HP2225B ThinkJet Druckerhandbuch  
 Die beiliegende Tabelle wurde eingegeben und gedruckt. Aufgrund des Umfangs wird hier nur das Prinzip der Dateneingabe darge-stellt.

Lösung:

| Anzeige   | Eingabe | Funktion | Kommentar                                                                                    |
|-----------|---------|----------|----------------------------------------------------------------------------------------------|
| N         | 4       | XEQ TAB  | Programmstart                                                                                |
| NZ        | 13      | R/S      | Anzahl der Spalten                                                                           |
| SG        |         | R/S      | Anzahl der Zeilen                                                                            |
|           |         | R/S      | Sind alle Spalten gleichbreit?<br>ja: beliebige Zahl = 0 ein-geben<br>nein: R/S              |
| B         | 3       | R/S      | Breite der ersten Spalte                                                                     |
| B         | 7       | R/S      | Breite der zweiten Spalte                                                                    |
| B         | 7       | R/S      | Breite der dritten Spalte                                                                    |
| B         | 7       | RS       | Breite der vierten Spalte                                                                    |
| UL        | 1       | R/S      | Unterstreichen an:<br>ja: R/S<br>nein: beliebige Zahl = 0                                    |
| STZ       | 124     | R/S      | Spaltentrennzeichen, (:) beliebiges ASCII-Zeichen ≥ 32 (dezimal)                             |
| TEXT, N=3 | (SPACE) | R/S      | max. 3 Zeichen Text                                                                          |
| TEXT, N=7 | K 63    | R/S      | max. 7 Zeichen Text                                                                          |
| TEXT, N=7 | K 64    | R/S      | max. 7 Zeichen Text                                                                          |
| TEXT, N=7 | K 65    | R/S      | max. 7 Zeichen Text                                                                          |
| TEXT, N=3 | (SPACE) | R/S      | max. 3 Zeichen Text                                                                          |
| ...       |         |          | usw.                                                                                         |
|           |         |          | Die gesamte Tabelle ist als Kopie beigelegt. Bitte mit den dortigen Werten vervollständigen. |

# SERIE 40

## Benutzeranweisungen:

| Schritt  | Anweisungen                                                                                                                                                 | Eingabe  | Funktion         | Anzeige    |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|------------------|------------|
| 1<br>1.1 | Programm starten<br>falls SIZE zu klein,<br>neuen Speicherplatz<br>definieren (5 + Anzahl<br>der Spalten)                                                   |          | XEQ TAB          | N          |
|          |                                                                                                                                                             |          | XEQ SIZE         |            |
| 2        | Anzahl Spalten eingeb.<br>Anzahl Zeilen eingeb.                                                                                                             | N<br>NZ  | R/S<br>R/S       | NZ<br>SG   |
| 3        | definieren ob alle<br>Spalten gleich breit<br>sind:<br>ja: SG<br>nein: SG                                                                                   |          | 1...9 R/S<br>R/S | B<br>B     |
| 3.1      | Spaltenbreite eingeben<br>wenn mehr als 80 Zeichen<br>definiert wurden,<br>erscheint Fehlermeldung<br>auf compressed mode<br>schalten<br>ja: SG<br>nein: SG | B        | R/S              | Z.ZU BREIT |
|          | wenn Fehlermeldung<br>erneut auftaucht, dann<br>Spaltenbreite verringern                                                                                    |          | R/S              | COMPR?     |
| 4        | Unterstreichen an? ja:<br>nein:                                                                                                                             | UL<br>UL | R/S<br>1...9 R/S | STZ<br>STZ |
| 5        | Spaltentrennzeichen<br>definieren (beliebiges<br>ASCII ≥ 32)                                                                                                | STZ      | 32->255          | TEXT, N=N  |
| 6        | Text der Maximallänge<br>N eingeben                                                                                                                         | TEXT     | R/S              | TEXT       |
| 6.2      | Text eingeben, bis<br>Tabelle beendet                                                                                                                       | usw.     |                  | usw.       |

```

↳ 18 PROMPT%
↳ 19 STO 02%
↳ 20 CLX%
↳ 21 "SG"%
↳ 22 PROMPT%
↳ 23 X=0?%
↳ 24 GTO 00%
↳ 25 SF 01%
↳ 26 LBL 00%
↳ 27 "B"%
↳ 28 PROMPT%
↳ 29 FS? 01%
↳ 30 STO 03%
↳ 31 FS? 01%
↳ 32 GTO 12%
↳ 33 STO IND 00%
↳ 34 %+% → Σ+
↳ 35 ISG 00%
↳ 36 GTO 00%
↳ 37 %REG?% → ΣREG ?
↳ 38 RCL IND X%
↳ 39 RCL 01%
↳ 40 1%
↳ 41 -%
↳ 42 +%
↳ 43 GTO 01%
↳ 44 LBL 12%
↳ 45 RCL 03%
↳ 46 RCL 01%
↳ 47 *%
↳ 48 LASTX%
↳ 49 1%
↳ 50 -%
↳ 51 +%
↳ 52 LBL 01%
↳ 53 81%
↳ 54 X<=Y?%
↳ 55 XEQ 16%
↳ 56 CLX%
↳ 57 "UL"%
↳ 58 PROMPT%
↳ 59 X%0?%
↳ 60 GTO 03%
↳ 61 "%&dD"%
↳ 62 ACA%
↳ 63 PRBUF%
↳ 64 LBL 03%
↳ 65 "STZ"%
↳ 66 PROMPT%
↳ 67 32%
↳ 68 X<>Y%
↳ 69 X<Y?%
↳ 70 GTO 03%
↳ 71 STO 04%
↳ 72 RCL 01%
↳ 73 XEQ 17%
↳ 74 LBL 04%
↳ 75 CF 23%
↳ 76 "TEXT,N=" %
↳ 77 FIX 0%
↳ 78 FS? 01%
↳ 79 ARCL 03%

```

Abfrage, ob alle Spalten gleich breit sind:  
ja -> Zahl ≠ 0  
nein -> R/S

Eingabe der Breite der i-ten Spalte

bereitet die Prüfung der Zeilenlänge im Fall ungleicher Spaltenbreiten vor

Soll jede Zeile unterstrichen werden?  
ja -> R/S  
nein-> Zahl ≠ 0

Escapesequ.f.Unterstreichung (s.Text)  
Die Sequ. wird sofort gedruckt= Lerrzeile

Eingabe eines Spaltenseparators mit ASCII-Code ≥ 32

Anzeige, welche die erlaubte Zahl von Zeichen in der nun einzugebenden Spalte zeigt

|    | K 63  | K 64  | K 65  |
|----|-------|-------|-------|
| 1  | 1.476 | 1.800 | 1.797 |
| 2  | 1.528 | 1.613 | 1.696 |
| 3  | 1.396 | 1.556 | 1.604 |
| 4  | 1.347 | 1.481 | 1.522 |
| 5  | 1.339 | 1.427 | 1.508 |
| 6  | 1.335 | 1.564 | 1.499 |
| 7  | 1.690 | 1.768 | 1.877 |
| 8  | 2623  | 3.162 | 2.738 |
| 9  | 1.474 | 1.479 | 1.604 |
| 10 | 1.354 | 1.466 | 1.517 |
| 11 | 1.339 | 1.539 | 1.508 |

```

↳ 01 LBL "TAB"%
↳ 02 CL % → CLΣ
↳ 03 SF 21%
↳ 04 SF 17%
↳ 05 CF 01%
↳ 06 "N"% Anzahl der Spalten
↳ 07 PROMPT%
↳ 08 STO Y%
↳ 09 5% Kontrollzahl zum Test
↳ 10 +% von SIZE
↳ 11 SIZE?%
↳ 12 X<Y?%
↳ 13 XEQ 15%
↳ 14 RCL Z%
↳ 15 STO 01%
↳ 16 XEQ 17% Anzahl der zu drucken-
↳ 17 "NZ"% den Zeilen

```

```

F 80 FC? 01%
F 81 ARCL IND 00%
F 82 AON%
F 83 PROMPT%
F 84 AOFF%
F 85 FS? 23%
F 86 GTO 08%
F 87 CLA%
F 88 32%
F 89 XTOA%
F 90 LBL 08%
F 91 FIX 5%
F 92 ALENG%
F 93 FS? 01%
F 94 RCL 03%
F 95 FC? 01%
F 96 RCL IND 00%
F 97 X=Y?%
F 98 GTO 05%
F 99 X>Y?%
F 100 GTO 06%
F 101 "INP. TO LONG"%
F 102 PROMPT%
F 103 GTO 04%
F 104 LBL 06%
F 105 -%
F 106 CHS%
F 107 LBL 07%
F 108 32%
F 109 XTOA%
F 110 RDN%
F 111 DSE X%
F 112 GTO 07%
F 113 LBL 05%
F 114 OUTA%
F 115 RCL 04%
F 116 ACCHR%
F 117 CLA%
F 118 ISG 00%
F 119 GTO 04%
F 120 PRBUF%
F 121 RCL 01%
F 122 XEQ 17%
F 123 DSE 02%
F 124 GTO 04%
F 125 GTO 09%
F 126 LBL 17%
F 127 4%
F 128 +%
F 129 1 E3%
F 130 /%
F 131 5%
F 132 +%
F 133 STO 00%
F 134 RTN%
F 135 LBL 15%
F 136 "RESIZE"%
F 137 PROMPT%
F 138 RTN%
F 139 LBL 16%
F 140 "Z. ZU BREIT"%
F 141 PROMPT%

```

Eingabe des Textes  
in der aktuellen  
Spalte

Prüft, ob zu viele  
Zeichen eingegeben  
wurden

neue Eingabe einer  
langen Spalte

füllt mit nachfol-  
genden Leerz. auf

Spalte wird in den  
Druckbuffer geschr.

Spaltenseparator  
wird angehängt

nächste Spalte  
Zeile drucken

Spaltenbereich  
wiederherstellen

Benötigter Speicher  
für Spaltenbreiten  
berechnen

Programmstop, wenn  
mehr Speicher nötig  
Per Hand neues SIZE

Anzeige, wenn Zeile  
> 80 Zeich. enthält

```

F 142 CLX%
F 143 "COMPR?"%
F 144 PROMPT%
F 145 X#0?%
F 146 GTO 16%
F 147 RDN%
F 148 143%
F 149 X<=Y?%
F 150 GTO 16%
F 151 "E&k2S"%
F 152 ACA%
F 153 PRBUF%
F 154 RTN%
F 155 LBL 09%
F 156 END%

```

Compressed-Mode  
einstellen ?  
→ x#0? ja -> R/S  
nein-> Zahl# 0

Prüft Zeilenlänge  
im Compressed-Mode

Im Falle einer  
neuen Fehleran-  
zeige (Z.140) bitte  
Tabelle neu organi-  
sieren !

Escape Sequ. für  
Compr.Mode (s.Text)

Gunnar Radons  
Kolbenzeil 18  
6900 Heidelberg

### Astrofotografie

## SE

### Schwarzschildexponent

39 Zeilen, 92 Bytes, 14 Regs., SIZE 006, HP41C

Das Programm berechnet anhand der Aufnahmezeiten und erreichten Größenklassen von Astrofotografien den Schwarzschildexponenten eines Films.

Je mehr Daten von Aufnahmen vorhanden sind, desto genauer wird der Wert. Es müssen mindestens die Daten von zwei Aufnahmen bekannt sein.

Der Idealwert ist 1, wird in der Praxis jedoch noch nicht erreicht, wobei selbst die Astroemulsionen nicht diesen Wert erreichen.

Beispiel:

4 Aufnahmedaten pro Aufnahmenpaar. Es sind Daten von 4 Paaren bekannt.

- A.) T1 = 1 min T2 = 2 min MAG1 = 8,95 MAG2 = 9,44
- B.) T1 = 1 min T2 = 4 min MAG1 = 8,95 MAG2 = 10,12
- C.) T1 = 8 min T2 = 1 min MAG1 = 10,75 MAG2 = 8,95
- D.) T1 = 16 min T2 = 1 min MAG1 = 11,47 MAG2 = 8,95

Man sieht hier in den Aufnahmezeiten eine Regelmäßigkeit: Die Aufnahmezeiten müssen sich immer verdoppeln: 1, 2, 4, 8, 16 usw.

### Beispiel :

```

XEQ "SE"
MAG1?      8,9500   RUN
MAG2?      9,4400   RUN
T1(MIN)?   1,0000   RUN
T2(MIN)?   2,0000   RUN
P=0,6511
PH=0,6511

```

|           |         |     |                          |     |               |             |
|-----------|---------|-----|--------------------------|-----|---------------|-------------|
| MAG1?     | 8,9500  | RUN | PM=0,7419                | RUN | 01*LBL "SE"   | 23 /        |
| MAG2?     | 10,1200 | RUN | MAG1?                    | RUN | 02 ΣREG 00    | 24 *P="     |
| T1(MIN)?  | 1,0000  | RUN | MAG2?                    | RUN | 03 CLΣ        | 25 ARCL X   |
| T2(MIN)?  | 4,0000  | RUN | T1(MIN)?                 | RUN | 04*LBL 01     | 26 PROMPT   |
| P=0,7773  |         |     | T2(MIN)?                 | RUN | 05 "MAG1?"    | 27 Σ+       |
| PM=0,7142 |         |     |                          | RUN | 06 PROMPT     | 28 MEAN     |
| MAG1?     | 10,7500 | RUN | PM=0,7657                | RUN | 07 XEQ 00     | 29 *PM="    |
| MAG2?     | 8,9500  | RUN |                          |     | 08 "MAG2?"    | 30 ARCL X   |
| T1(MIN)?  | 8,0000  | RUN |                          |     | 09 PROMPT     | 31 PROMPT   |
| T2(MIN)?  | 1,0000  | RUN | P: Schwarzschildexponent |     | 10 XEQ 00     | 32 GTO 01   |
| P=0,7973  |         |     | PM: Mittelwert           |     | 11 X>Y?       | 33*LBL 00   |
|           |         |     |                          |     | 12 X<>Y       | 34 ,4       |
|           |         |     |                          |     | 13 /          | 35 *        |
|           |         |     |                          |     | 14 LOG        | 36 10*X     |
|           |         |     |                          |     | 15 "T1(MIN)?" | 37 2,06 E-6 |
|           |         |     |                          |     | 16 PROMPT     | 38 *        |
|           |         |     |                          |     | 17 "T2(MIN)?" | 39 END      |
|           |         |     |                          |     | 18 PROMPT     |             |
|           |         |     |                          |     | 19 X>Y?       |             |
|           |         |     |                          |     | 20 X<>Y       |             |
|           |         |     |                          |     | 21 /          |             |
|           |         |     |                          |     | 22 LOG        |             |

Stephan Trzeciak + Dennis Föh  
Fritz-Reuter-Straße 24  
2190 Cuxhaven

## Erläuterungen zu XSPLINE

Mein Beitrag soll eine Ergänzung zu der ausgezeichneten Vorarbeit von Wilfried Mlaker sein. Leider könnte schon die in der Überschrift genannte Ausrüstung davon abhalten, sich mit dem Programm zu beschäftigen. In der Tat benötigt man nämlich nur

HP 41C, MM X-M und Thermodrucker, dann bringt man immerhin, wenn es sein muß, 17 Stützstellen unter. Den CCD-ROM braucht man allenfalls, um RCL 0 bzw. STO 0 ein bzw. zweimal einzutasten. Normalerweise druckt Ihr ja die Barcodes mit ab. Dann entfällt das Problem; ebenso, wenn man mir 1 Cassette oder 10 Magnetkarten schickt. Ich sende sie beschrieben zurück.

Schön ist ja auch, wenn man eine Information zur Hand hat, damit man weiß, was man mit dem Programm machen kann und wie es läuft.

Wir hatten in den letzten 12 Monaten nun 4 Programme zum Thema „Spline“, und zu keinem wurde andeutungsweise gesagt, was es damit auf sich hat. Kurz: ein Spline ist ein durch eine vorgegebene Anzahl (mindestens 3) Stützpunkte gelegter Kurvenzug mit der Bedingung der zweimaligen stetigen Differenzierbarkeit. Im Klartext: keine Knik-ke im Verlauf. Deshalb die Anwendung z.B. im Gleisanlagenbau, im Fahr- und Flugzeug- sowie Schiffsbau. Deswegen schien es mir auch wichtig, das Programm mit dem Weg zur Berechnung der Krümmungswerte auszustatten.

Wer über den Beitrag von Wilfried Mlaker nicht mehr verfügt, kommt leicht auf folgende Weise zu Rande:

Es wird die Anzahl der Stützpunkte abgefragt und dann ein Datenfile angelegt, dem

man einen Namen geben muß (FILE?). Die Koordinaten müssen in x monoton eingegeben werden; falls nicht, springt der Rechner zurück und verlangt Berichtigung. Das Programm läßt die Berücksichtigung von Randbedingungen unter dem Stichwort MASS zu. 1, 2 bzw. 3 (R/S) bedeuten, daß man die betreffende Ableitung am Rand festlegen will. Nach A bzw. B werden dann die gewünschten Werte eingegeben. Mit MASS = 4 (A und B unwichtig) erhält man einen periodischen Spline, für den die Y-Werte, aber auch die der 1. und 2. Ableitung am unteren und oberen Rand identisch sind.

(Y<sub>0</sub> und Y<sub>D</sub> muß natürlich der Anwender selbst identisch eingeben!) MASS 0 ist ein von ein Wilfried hereingekommener Sonderfall, bei dem die 3. Ableitungen in den benachbarten Spline-Paaren an den Rändern identisch werden.

Mit „a“ läßt sich die Berechnung auf Basis der vorherigen Stützpunkte, aber mit neuen Randbedingungen wiederholen. „B“ druckt die Koeffizienten aus. „C“ plottet mit der Maßgabe, daß X INC abgefragt und entweder die Schrittweite oder die Schrittzahl (CHS) erwartet wird. Die Plotbreite ist 20% größer als die maximale Ordinatendifferenz. Mit „D“ erhält man zum x-Wert y, y', y'' und 1/p (Krümmung). E bringt Nullstellen für vorgegebene Werte von y, y' oder y'', wobei man den Plot zu Rate ziehen sollte, damit die Rechnung nicht aus dem Definitionsbereich herausgeschleudert wird. „d“ erlaubt schließlich eine Korrektur der Stützpunktwerte. Abgefragt wird die Indexzahl des zu korrigierenden Stützpunktes, mit ← wird der Wert gelöscht, und mit EEX (nach einem y) gelangt man aus der Routine wieder heraus. Das wär's denn auch schon.

Auch für den, der das Programm niemals beruflich brauchen wird, eine Anregung: man erdenke sich eine Stützpunktfolge von, sagen wir 10 Punkten, lasse den dazugehörigen Spline errechnen, ermittle dann Zwischenwerte zwischen den Punkten. Danach ändere man z.B. den y-Wert des letzten Punktes geringfügig und frage danach, wie weit sich diese Änderung rückwärts auswirkt.

Und nun frohes, schnelles Rechnen!

Liste der Abbildungen:

- Abb. 1 INFO (XEQ I)
- Abb. 2 EINGABE (XEQ A)
- Abb. 3 KOEFFIZIENTEN (XEQ B)
- Abb. 4 WERTE (XEQ D)
- Abb. 5 NULLSTELLE (XEQ E)
- Abb. 6 EDITIEREN (XEQ d)

Abb. 1 INFO (XEQ I)

```

XEQ "IN"
A ST ART→SPLINE
a NE UE RANDWERTE
B KO EFFIZIENTEN
C PL OT
D WE RTE: Y Y' Y" 1/p
d XY EDIT
E NU LLSTELLEN
I IN FO
    
```

```

N=?
XEQ "ST"
5. RUN
FILE?
BS-SP RUN
X1.=
-3. RUN
Y1.=
4. RUN
X2.=
-1.5 RUN
Y2.=
-1. RUN
X3.=
.5 RUN
Y3.=
3. RUN
X4.=
3. RUN
Y4.=
0. RUN
X5.=
5.5 RUN
Y5.=
2. RUN
MASS?
2. RUN
A?
0. RUN
B?
0. RUN
FERTIG!
    
```

Abb. 3 Koeffizienten (XEQ B)

```

XEQ "KO"
Spline 1.
D1.= 0.6354
C1.= 0.0000
B1.= -4.7631
A1.= 4.0000
Spline 2.
D2.= -0.8113
C2.= 2.8595
B2.= -0.4738
A2.= -1.0000
Spline 3.
D3.= 0.4147
C3.= -2.0082
B3.= 1.2287
A3.= 3.0000
    
```

```

Spline 4.
D4.= -0.1469
C4.= 1.1021
B4.= -1.0368
A4.= 0.0000
ENDE
    
```

Abb. 4 WERTE (XEQ D)

```

XEQ "WE"
X
-1.4999 RUN
X = -1.4999
Y = -1.0000
Y' = -0.4733
Y'' = 5.7185
KR = 4.2228
XEQ "WE"
X
-1.5 RUN
X = -1.5000
Y = -1.0000
Y' = -0.4738
Y'' = 5.7190
KR = 4.2207
    
```

Abb. 5 NULLSTELLE (XEQ E)

```

XEQ "NU"
Y:0 Y':1 Y'':2
1. RUN
SOLL?
1.2 RUN
SOLL? 1.2000
    
```

### Anmerkungen zu TR3 und TR4

Liebe Clubfreunde!

Wegen meiner Programme TR3-TR4 im Prisma Nr. 5/87 wurde ich von einem Kollegen angeschrieben, daß meine Innenfeldmomente nicht richtig wären.

Diese Feldmomente sind bewußt etwas höher angesetzt. Die rechnerisch genaueren Werte ergeben sich, wenn bei TR3, Zeile 213 der Festwert 0,4 auf 0,5 geändert und bei TR4 Zeile 666 und 679 ebenfalls von 0,4 auf 0,5 abgeändert werden.

Dann stimmts auch genau nach den anderen Programmen in älteren Prisma-Heften.

Also nichts für ungut

freundliche Grüße,

Alfred Büttner  
Ernst Kuhn Straße 5  
8000 München 50

```

X
-1.1800 RUN
X = -1.1800
Y = -0.8854
Y' = 1.1070
Y'' = 4.1613
X = -1.1577
Y = -0.8596
Y' = 1.1988
Y'' = 4.0525
    
```

```

X = -1.1574
Y = -0.8593
Y' = 1.2000
Y'' = 4.0511
X = -1.1574
    
```

```

FERTIG!
XEQ "NU"
Y:0 Y':1 Y'':2
1. RUN
SOLL?
1.2 RUN
SOLL? 1.2000
X
-1.2540 RUN
X = -1.2540
Y = -0.9556
Y' = 0.7857
Y'' = 4.5215
X = -1.1624
Y = -0.8652
Y' = 1.1796
Y'' = 4.0755
    
```

```

X = -1.1574
Y = -0.8593
Y' = 1.2000
Y'' = 4.0511
X = -1.1574
    
```

FERTIG!

Abb. 6 EDITIEREN (XEQ d)

```

XEQ "XY"
INDEX?
3. RUN
X3.= 0.5000
Y3.= 3.0000
3.5 RUN
X3.= 0.5000
Y3.= 3.5000
    
```

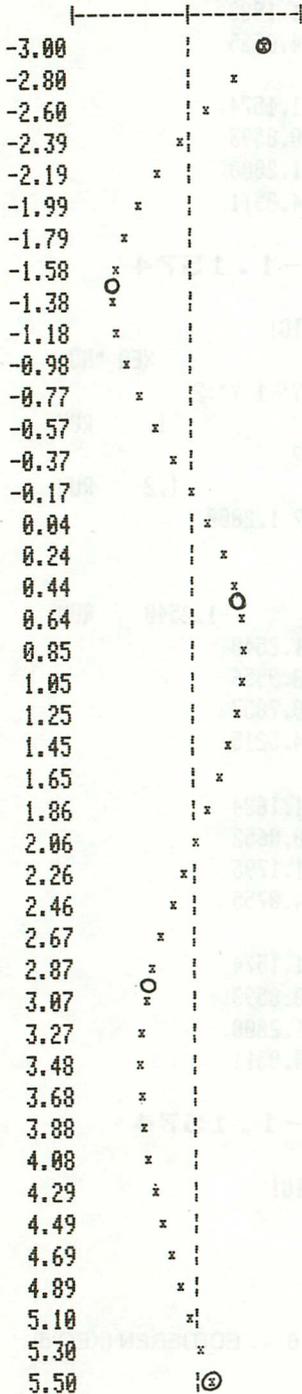
# Steiner-Satz

XEQ "PL"  
X INC?  
-42.00 RUN

208 Zeilen, 447 Bytes, 64 Regs., SIZE 019,  
HP41C, Drucker, ADVANCE

84 FC? 21  
85 GTO 02  
86 SF 12  
87 "----"  
88 ACA  
89 ACA  
90 ACA  
91 ADV  
92 ADV  
93 CF 12  
94+LBL 02

PLOT OF SPLINE  
X <UNITS= 1. > ↓  
Y <UNITS= 1. > ↑  
-2.25 1.50 5.25



Als Student des Fachbereiches Bauingenieurwesen und neues Mitglied im CCD (1. Jan. 87) möchte ich hiermit einen hoffentlich interessanten Beitrag für PRISMA leisten.

Es handelt sich um ein Programm aus der Festigkeitslehre und berechnet den Schwerpunkt, Trägheitsmomente, Zentrifugalmoment sowie die Neigung der Hauptträgheitsachsen mit den dazugehörigen Trägheitsmomenten ( $I_{min}$ ,  $I_{max}$ ) von zusammengesetzten Querschnitten (beliebig viele).

Es ist sowohl die Verwendung mit und ohne Thermodrucker möglich. Im MAN-Modus werden sowohl Ein- wie Ausgabedaten gedruckt. Sollte trotz angeschlossenem Drucker (ein- oder ausgeschaltet) kein Ausdruck erwünscht sein, so ist Flag 21 zu löschen.

Wenn vor Programmstart Flag 27 (USER) gesetzt war, so ist dies auch nach Programmende der Fall und umgekehrt.

Durch geringe Änderungen ist es möglich auf den Befehl „AIP“ aus dem Advantage-Rom zu verzichten, jedoch sind dann, sollte bei der Eingabe noch manuell gerechnet werden, bevor R/S gedrückt wird, im Display (nicht im Drucker) keine Nachkommastellen und kein Dezimalkomma zu sehen. Außerdem benötigt man dann mehr als 2 Magnetkarten.

Die Programmzeile 43 (RCL 15) kann dann gelöscht werden.

```
26+LBL 15
27 FIX 0
28 CF 29
29 ARCL 00
30 "I= ? "
31 PROMPT
32 SF 29
33 FIX 2
34 ARCL X
35 FS? 21
36 PRA
37 RTN
38+LBL 01
```

Da der Befehl „AIP“ aus dem Kapitel Matrix-Operationen das Advantage-Rom stammt, welches sich laut Bedienungshandbuch auf das CCD-ROM stützt, ist es möglich, daß dieses den Befehl ebenso enthält.

Die synthetisch erzeugte Rautentrennlinie kann durch folgende Befehl ersetzt werden:

```
80+LBL 01
81 ADV
82 ISG 00
83 GTO 16
```

- Gestartet wird das Programm mit XEQ „STZ“
- ANZAHL? Man gibt die Anzahl der Querschnitte ein
- 1A 2A Durch drücken der Tasten A oder B wählt man zwischen einachsiger oder zweiachsiger Biegung
- Abfrage und Eingabe der Daten und Drucken
- Ausgabe bzw. Ausdruck der Ergebnisse (Sollen die Neigung und  $I_{max}$  und  $I_{min}$  auch ausgedruckt werden so ist nach Erscheinen von „HAUPTTR.-A.“ R/S zu drücken)

- A = Fläche eines Einzelquerschnittes ( $cm^2$ )
- a = Abstand des Schwerpunktes eines Einzelquerschnittes vom o Punkt des a-b Koordinatensystems (cm)
- b = dto.
- la = Trägheitsmoment eines Einzelquerschnittes ( $cm^4$ )
- lb = Trägheitsmoment eines Einzelquerschnittes ( $cm^4$ )
- lab = Zentrifugalmoment eines Einzelquerschnittes ( $cm^4$ )
- aS = Lage des Gesamtschwerpunktes (o Punkt des Koordinatensystems yz)
- bS = dto.
- IY = Trägheitsmoment um y-Achse
- IZ = Trägheitsmoment um z-Achse
- IYZ = Zentrifugalmoment
- α = Neigung der Hauptträgheitsachsen (Altgrad)
- I max = Maximales Trägheitsmoment bezogen auf Hauptträgheitsachsen
- I min = Minimales Trägheitsmoment bezogen auf Hauptträgheitsachsen

Die Entscheidung, welche Achse des geneigten Koordinatensystems u-v nun  $I_{max}$  oder  $I_{min}$  ist, muß jedoch bei diesem Programm noch mit Überlegung getroffen werden.

Bei der Eingabe des Zentrifugalmomentes lab der einzelnen Querschnitte, ist auf das Vorzeichen des jeweiligen Quadranten, in dem der Einzelquerschnitt sich befindet, zu achten.

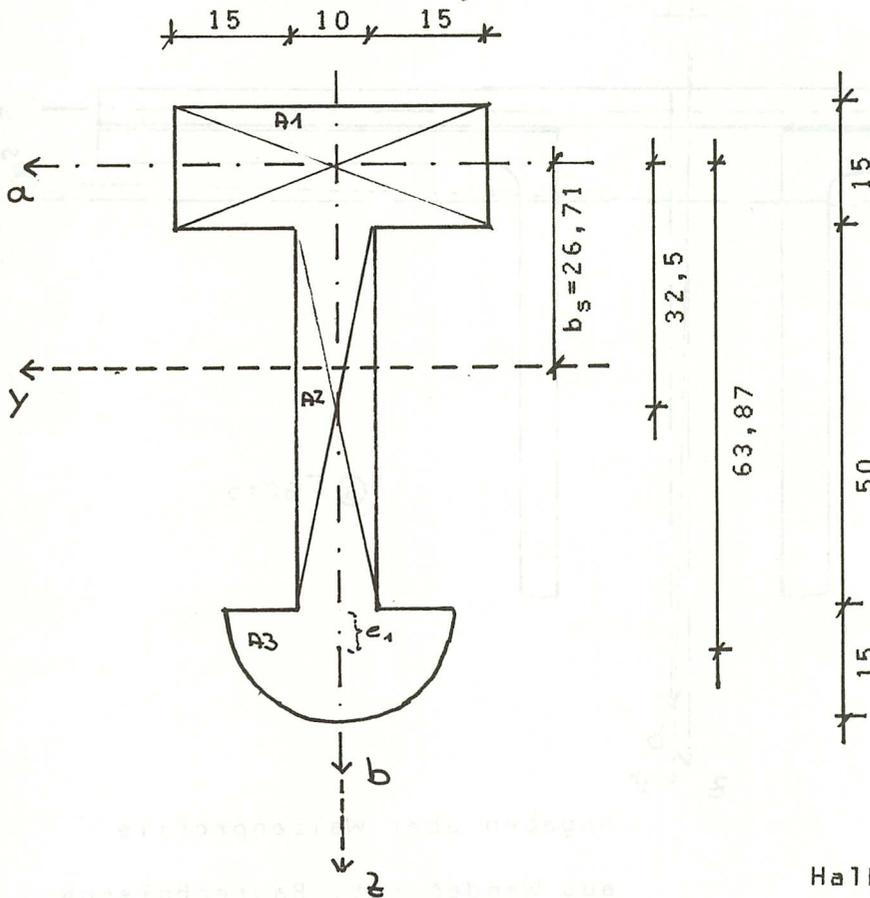
Sollte  $I_y$ ,  $I_z$  und  $I_{yz}$  bekannt sein, so startet man das Programm mit XEQ „HTA“. Es werden dann der Winkel,  $I_{max}$  und  $I_{min}$  berechnet.

Das Programm verwendet die Register 00, 06-19. Die Register 01-05 werden bewußt nicht verwendet, da dort häufig Daten von manuellem Rechnen gespeichert werden.

FERTIG!

Dr. Hans Berghaus  
An der Joch 19  
5000 Köln 80

1. Beispiel: einachsige Biegung



Formeln

$$a_s = \frac{\sum (A_i \cdot a_i)}{\sum A_i} \quad b_s = \frac{\sum (A_i \cdot b_i)}{\sum A_i}$$

$$J_y = \sum J_{ai} + \sum (A_i \cdot b_i^2) - \sum (A_i) \cdot b_s^2$$

$$J_z = \sum J_{bi} + \sum (A_i \cdot a_i^2) - \sum (A_i) \cdot a_s^2$$

$$J_{yz} = \sum J_{abi} + \sum (A_i \cdot a_i \cdot b_i) - \sum (A_i) \cdot a_s \cdot b_s$$

$$\alpha = \frac{1}{2} \arctan \left( -\frac{2 \cdot J_{yz}}{J_y - J_z} \right)$$

$$I_{\max} = \frac{1}{2} (J_y + J_z) + \frac{1}{2} \sqrt{(J_y - J_z)^2 + 4 \cdot J_{yz}^2}$$

$$I_{\min} = \frac{1}{2} (J_y + J_z) - \frac{1}{2} \sqrt{(J_y - J_z)^2 + 4 \cdot J_{yz}^2}$$

A 1 = ? 600,00  
 b 1 = ? 0,00  
 Ia 1 = ? 11.250,00

A 2 = ? 500,00  
 b 2 = ? 32,50  
 Ia 2 = ? 104.166,67

A 3 = ? 353,43  
 b 3 = ? 63,87  
 Ia 3 = ? 5.568,75

\*\*\*\*\*

bS = 26,71  
 IY = 1.053.841,26

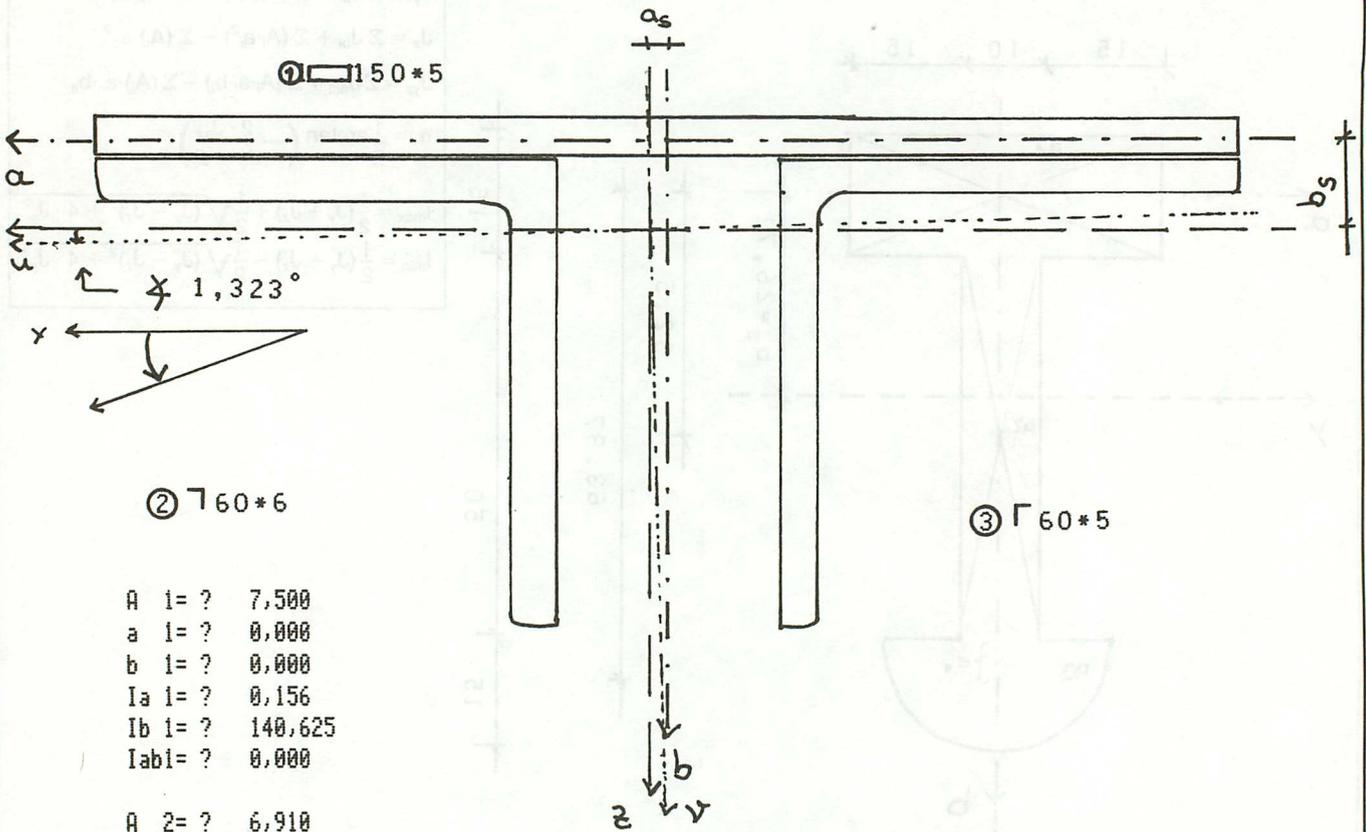
Halbkreis  $\left\{ \begin{array}{l} e_1 = \frac{4 \cdot r}{3\pi} \\ I_{a3} = 0,11 \cdot r^4 \end{array} \right.$

$$I_{a1} = \frac{40 \cdot 15^3}{12} = 11250,00$$

$$I_{a2} = \frac{10 \cdot 50^3}{12} = 104166,67$$

$$I_{a3} = 0,11 \cdot 15^4 = 5568,75$$

2. Beispiel: zweiachsige Biegung



|          |         |
|----------|---------|
| A 1 = ?  | 7,500   |
| a 1 = ?  | 0,000   |
| b 1 = ?  | 0,000   |
| Ia 1 = ? | 0,156   |
| Ib 1 = ? | 140,625 |
| Iab1 = ? | 0,000   |
|          |         |
| A 2 = ?  | 6,910   |
| a 2 = ?  | 3,190   |
| b 2 = ?  | 1,940   |
| Ia 2 = ? | 22,800  |
| Ib 2 = ? | 22,800  |
| Iab2 = ? | 13,400  |
|          |         |
| A 3 = ?  | 5,820   |
| a 3 = ?  | -3,140  |
| b 3 = ?  | 1,890   |
| Ia 3 = ? | 19,400  |
| Ib 3 = ? | 19,400  |
| Iab3 = ? | -11,300 |

\*\*\*\*\*

a<sub>S</sub> = 0,186  
 b<sub>S</sub> = 1,206  
 I<sub>Y</sub> = 59,710  
 I<sub>Z</sub> = 309,823  
 I<sub>YZ</sub> = 5,778

α = 1,323  
 I<sub>MAX</sub> = 309,956  
 I<sub>MIN</sub> = 59,577

I<sub>min</sub> = I<sub>u</sub>  
 I<sub>max</sub> = I<sub>v</sub>

Angaben über Walzenprofile  
 aus Wendehorst, Bautechnische  
 Zahlentafeln, Teubner-Verlag  
 Stuttgart 22. Auflage

A<sub>1</sub> = 15 \* 0,5 = 7,5 cm<sup>2</sup>  
 Ia<sub>1</sub> = 0,156 cm<sup>4</sup>  
 Ib<sub>1</sub> = 140,625 cm<sup>4</sup>  
 a<sub>2</sub> = 1,69 + (7,5 - 6) = 3,19 cm  
 b<sub>2</sub> = 1,69 + 0,25 = 1,94 cm  
 a<sub>3</sub> = -[1,64 + (7,5 - 6)] = -3,14 cm  
 b<sub>3</sub> = 1,64 + 0,25 = 1,89 cm

|               |             |                   |               |
|---------------|-------------|-------------------|---------------|
| 01+LBL "STZ"  | 64 XEQ 15   | 126 -             | 188 RCL 17    |
| 02 006.019    | 65 ST+ 11   | 127 STO 18        | 189 RCL 18    |
| 03 CLRGX      | 66 RCL 15   | 128 "IZ = "       | 190 +         |
| 04 "ANZAHL ?" | 67 RCL 16   | 129 XEQ 17        | 191 2         |
| 05 PROMPT     | 68 X↑2      | 130 RCL 14        | 192 /         |
| 06 1 E3       | 69 *        | 131 RCL 13        | 193 STO 16    |
| 07 /          | 70 ST+ 12   | 132 +             | 194 RCL 15    |
| 08 1          | 71 RCL 15   | 133 RCL 06        | 195 +         |
| 09 +          | 72 RCL 16   | 134 RCL 16        | 196 "I MAX= " |
| 10 STO 00     | 73 *        | 135 *             | 197 XEQ 17    |
| 11 FS? 27     | 74 RCL 17   | 136 RCL 15        | 198 RCL 16    |
| 12 SF 02      | 75 *        | 137 *             | 199 RCL 15    |
| 13 SF 27      | 76 ST+ 13   | 138 -             | 200 -         |
| 14 "1A 2A"    | 77 "Iab"    | 139 "IYZ= "       | 201 "I MIN= " |
| 15 PROMPT     | 78 XEQ 15   | 140 XEQ 17        | 202+LBL 17    |
| 16+LBL A      | 79 ST+ 14   | 141 ADV           | 203 ARCL X    |
| 17 SF 01      | 80+LBL 01   | 142 "HAUPTTR.-A." | 204 FC? 21    |
| 18+LBL B      | 81 ADV      | 143 PROMPT        | 205 PROMPT    |
| 19 FC? 02     | 82 ISG 00   | 144 SF 00         | 206 FS? 21    |
| 20 CF 27      | 83 GTO 16   | 145 XEQ "HTA"     | 207 PRA       |
| 21 CF 02      | 84 FC? 21   | 146+LBL 01        | 208 END       |
| 22+LBL 16     | 85 GTO 02   | 147 CF 01         |               |
| 23 "A "       | 86 ""       | 148 RTN           |               |
| 24 XEQ 15     | 87 ASTO ↑   | 149+LBL "HTA"     |               |
| 25 GTO 01     | 88 PRA      | 150 DEG           |               |
| 26+LBL 15     | 89 ADV      | 151 FS?C 00       |               |
| 27 RCL 00     | 90+LBL 02   | 152 GTO 01        |               |
| 28 AIP        | 91 FS? 01   | 153 "IY =?"       |               |
| 29 "t= ? "    | 92 GTO 01   | 154 PROMPT        |               |
| 30 PROMPT     | 93 RCL 07   | 155 STO 17        |               |
| 31 ARCL X     | 94 RCL 06   | 156 "IZ =?"       |               |
| 32 FS? 21     | 95 /        | 157 PROMPT        |               |
| 33 PRA        | 96 STO 15   | 158 STO 18        |               |
| 34 RTN        | 97 "aS = "  | 159 "IYZ =?"      |               |
| 35+LBL 01     | 98 XEQ 17   | 160 PROMPT        |               |
| 36 STO 15     | 99+LBL 01   | 161+LBL 01        |               |
| 37 ST+ 06     | 100 RCL 08  | 162 STO 19        |               |
| 38 FS? 01     | 101 RCL 06  | 163 2             |               |
| 39 GTO 01     | 102 /       | 164 *             |               |
| 40 "a "       | 103 STO 16  | 165 RCL 17        |               |
| 41 XEQ 15     | 104 "bS = " | 166 RCL 18        |               |
| 42 STO 16     | 105 XEQ 17  | 167 -             |               |
| 43 RCL 15     | 106 RCL 09  | 168 /             |               |
| 44 *          | 107 RCL 10  | 169 CHS           |               |
| 45 ST+ 07     | 108 +       | 170 ATAN          |               |
| 46+LBL 01     | 109 RCL 06  | 171 2             |               |
| 47 "b "       | 110 RCL 16  | 172 /             |               |
| 48 XEQ 15     | 111 X↑2     | 173 "λ = "        |               |
| 49 STO 17     | 112 *       | 174 XEQ 17        |               |
| 50 RCL 15     | 113 -       | 175 RCL 17        |               |
| 51 *          | 114 STO 17  | 176 RCL 18        |               |
| 52 ST+ 08     | 115 "IY = " | 177 -             |               |
| 53 "Ia "      | 116 XEQ 17  | 178 X↑2           |               |
| 54 XEQ 15     | 117 FS? 01  | 179 RCL 19        |               |
| 55 ST+ 09     | 118 GTO 01  | 180 X↑2           |               |
| 56 RCL 15     | 119 RCL 11  | 181 4             |               |
| 57 RCL 17     | 120 RCL 12  | 182 *             |               |
| 58 X↑2        | 121 +       | 183 +             |               |
| 59 *          | 122 RCL 06  | 184 SQRT          |               |
| 60 ST+ 10     | 123 RCL 15  | 185 2             |               |
| 61 FS? 01     | 124 X↑2     | 186 /             |               |
| 62 GTO 01     | 125 *       | 187 STO 15        |               |
| 63 "Ib "      |             |                   |               |

Stefan Schneider  
Gabelsbornstraße 30  
6200 Wiesbaden

## Der Druck- fehlerteufel

### Artikel PM (Polynommultiplikation) —>>

Seite 23, mittlere Spalte:  
statt GRAD? muß es GRAD 2? heißen, da-  
nach kommt direkt der Text, der nach der  
langen Eingabeliste steht „worauf die Daten  
des ...“

### Artikel Richtungsfelder —>>

1. Seite 31, 2. Spalte:  
die Formel  $U_{IND} = \dots$  müßte in den darüber-  
stehenden Text an die Stelle „... gegenin-  
duzierten Spannung  $U_{IND} = \dots$ “

2. Seite 31, rechts unten:  
in der Formel müßte es -R+t/L statt -R+t/dt  
heißen

3. Seite 33, Programmlisting:  
CF 21 in Zeile 02 des Programms ist über-  
flüssig, sie muß auch weg, damit auf Seite  
34 die synthetischen Textzeilen stimmen.

### Artikel EI, NST, BAIR —>>

Seite 35, das charakteristische Polynom  
hat einen Tippfehler, das Minus vor dem  
„hoch 3“ gehört weg vor das Lambda.

#### CHARAKTERISTISCHES POLYNOM

$$\Leftrightarrow \boxed{-\lambda^3 + 6\lambda^2 + 5\lambda = 0}$$

Martin Meyer (1000)  
Redaktion

# Dreiecksberechnungen

798 Zeilen, 1634 Bytes, 234 Regs. SIZE 039, HP41CV, X-F/M, X-I/O, (IL,) DRIVE

Als neues Mitglied des CCD-Clubs will ich gleich einmal meinen ersten Beitrag leisten.

Ich stelle Euch ein Programm vor, das mich viel Arbeit gekostet hat; nicht nur deshalb hoffe ich das die Anwender es nur für eigene Zwecke nutzen. Ansonsten bin ich sehr daran interessiert meine Erfahrungen und Kenntnisse an andere Club-Mitglieder weiterzugeben. Auch ich werde Probleme und Fragen an den Mann bringen.

Ich habe folgende Geräte:

- 2 HP25 – noch voll funktionsfähig –
- 1 HP41CV . . . . Time-Modul nachgerüstet
- 1 HP41CX
- 1 XF-Modul
- 1 IL-Modul
- 1 CCD-Modul
- 1 32kB RAM-Box
- 1 Magnetkartenleser
- 1 Barcodeleser
- 1 Kassettenlaufwerk
- 1 ThinkJet-Drucker

Mein erster Taschenrechner war der HP45, der allerdings Beine bekam. Das ist schade, da dieses Gerät eines der ersten in Deutschland war (wir brachten es als kleines Weltwunder aus den Vereinigten Staaten mit). Es wäre ein schönes Andenken an den Anfang der Taschenrechnerzeit gewesen. Am Schreibtisch arbeite ich nach wie vor mit Vorliebe mit dem HP25.

## Zum Programm

Bisherige, mir bekannte, Dreieck-Berechnungs-Programme haben den Nachteil, daß sie entweder nur für rechtwinklige Dreiecke ausgelegt sind, oder aber man muß die Art des Dreiecks (3 Seiten, 2 Seiten und 1 Außenwinkel usw.) bestimmen und separat eingeben.

Der Grundgedanke für das folgende Programm war; Beliebige Dreiecke in beliebiger Lage zu berechnen und es auch möglichst komfortabel und besonders für die Arbeit am Schreibtisch zweckmäßig zu gestalten.

## Programm-Vorgabe und Lösung:

1. Alle möglichen Dreiecksvarianten werden bestimmt.  
Insgesamt sind 19 Varianten zu berücksichtigen.

2. Jedes Dreieck wird in eine definierte Grundlage gebracht.  
Nachdem die Eingabe der Seiten:

- a = LBL A — STO 01
- b = LBL B — STO 02
- c = LBL C — STO 03

und Winkel:

- AL = LBL F — STO 04
- BE = LBL G — STO 05
- GA = LBL H — STO 06

fest zugeordnet sind, wird über die Abfrage der Speicherbelegung der FLAG-Status ermittelt und zur Unterprogrammverzweigung für die Steuerprogramme

genutzt. Diese Unterprogramme sind so der jeweiligen Eingabe direkt zugeordnet. Damit ist auch das jeweilige Dreieck erkannt und wird zur Berechnung in eine der Berechnungsart entsprechenden Lage gebracht.

3. Die Ergebnisse werden entsprechend der Anfangslage ausgegeben.  
Jetzt erfolgt die Verzweigung zur jeweiligen Berechnungsformel und Abspeicherung in jeweils der Ausgangslage entsprechenden Ergebnisregistern.

4. Die Ergebnisse werden in Klar-Text angezeigt,

Ergebnisanzeige:

- a = ... ; AL = ...
- b = ... ; BE = ...
- c = ... ; GA = ...

bei 2 Dreiecken zus.

- a' = ... ;AL' = ...
- b' = ... ;BE' = ...
- c' = ... ;GA' = ...

5. Die Ergebnisse sind ohne nochmalige Berechnung wiederholt anzeigbar.

Es werden alle Ergebnisse berechnet und können dann mit R/S im Rundumlauf abgefragt werden. Mit Absicht wurde darauf verzichtet die Eingabedaten mit anzuzeigen. Diese sind im Reg.-Block 13 ... 18 gespeichert.

6. Die Ergebnisse sind zum Weiterrechnen verwendbar, da die Anzeige im Alpharegister und zusätzlich im X-Register geladen wird.

7. Es wird berücksichtigt, wenn 2 Dreiecke möglich sind.

2 Dreiecke wenn:  
AL 90 und zusätzlich a und b.

8. Die Eingabe soll schnell hintereinander erfolgen können und überwacht werden. Auf die schnelle Verarbeitungszeit zwischen den Eingaben wurde besonders Wert gelegt. – Ich bin immer froh, wenn ich hintereinanderfolgende Daten schnell eintippen kann und nicht unnötige Pausen in Kauf nehmen muß –.

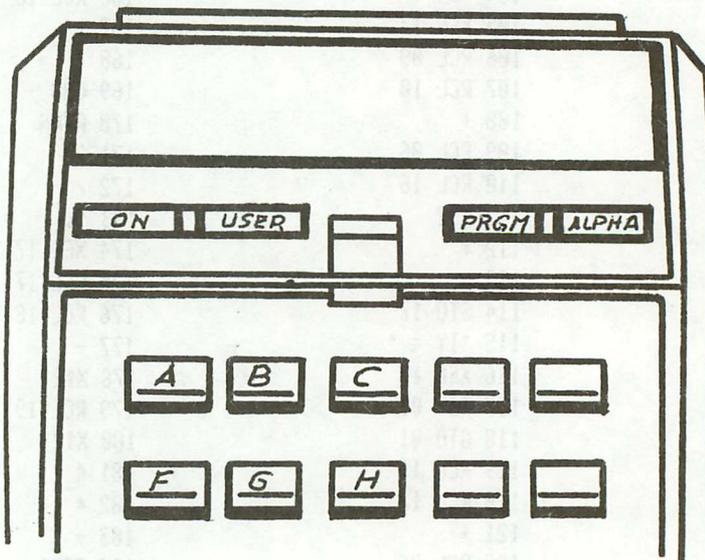
Die Eingaberoutine hätte auch so aussehen können, daß nach drei Eingaben automatisch in's Hauptprogramm gesprungen wird. Dann aber hätte ich zwischen den Eingaben auch noch die Zähl-Routine (DSE o.ä.) abwarten müssen. Eine dritte Möglichkeit: In die Register direkt einzugeben hätte die Betätigung zweier zusätzlicher Tasten erfordert.

direkt 25 STO 01  
jetzt 25 A

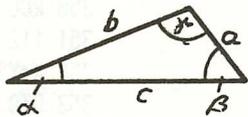
Die Eingabeüberwachung ist notwendig, um Eingabefehler von Datenfehlern zu trennen. Bei Fehleingabe (Anzahl der Eingaben) erfolgt die Anzeige der eingetippten Dreiecksbezeichnungen.

## Reg.-Belegung:

- 00 = aktiver Flag-Status
- 01 ... 06 = Dateneingabe
- 07 ... 126 = Arbeits-Reg.
- 13 ... 18 = Bereitschafts-Reg. (Eingabe-Daten)
- 19 ... 25 = Ergebnis-Reg. (1. Dreieck)
- 26 ... 32 = Ergebnis-Reg. (2. Dreieck)
- 33 ... 36 = Ergebnis-Bezeichnung
- 37, 38 = Eingabe-Bezeichnung

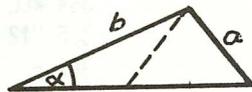


Beispiele mit dem gleichen Dreieck



Die Bezeichnung des Dreiecks erfolgt wie üblich entgegen dem Uhrzeigersinn.

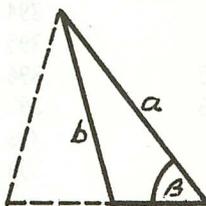
1. Eingabe: Taste:  
 a = 20 A  
 b = 100 B  
 alpha = 10 F



- a) Programm starten  
 b) Die Meldung: „DATEN ? TRI“ fordert zur Eingabe auf.  
 c) Nach der Dateneingabe mit R/S das Hauptprogramm starten.  
 d) Meldung: „ES GIBT ZWEI LÖSUNGEN C BE GA UND C' BE' GA'“  
 e) Erneut mit R/S starten.  
 f) Ergebnis:

c = 108,40  
 R/S BE = 60,25  
 R/S GA = 109,75  
 R/S FL = 941,20  
 R/S c = 108,40  
 R/S c' = 88,56  
 R/S BE' = 119,75  
 R/S GA' = 50,25  
 R/S FL' = 768,90  
 usw. rundum

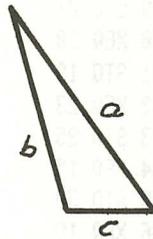
2. Eingabe: Taste:  
 a = 108,40 A  
 b = 100,00 B  
 beta = 60,25 G



Ergebnis:

c = 87,60 ( 87,57)  
 R/S AL = 70,24 ( 70,25)  
 R/S GA = 49,51 ( 49,49)  
 R/S FL = 4121,91 (4120,94)  
 c' = 19,98 ( 20,00)  
 R/S AL' = 109,76 ( 109,75)  
 R/S GA' = 9,99 ( 10,00)  
 R/S FL' = 940,41 ( 941,20)

3. Eingabe: Taste:  
 a = 108,40 A  
 b = 100,00 B  
 c = 20,00 C



Ergebnis:

AL = 109,73 (109,75)  
 BE = 60,27 ( 60,25)  
 GA = 10,00 ( 10,00)

Die Ergebnisse in Klammern sind genau und resultieren aus der Eingabe aus Beispiel 1 über Zwischenspeicherung.

Ich bin sehr gespannt wie das Programm ankommt und wäre den Anwendern für eine Beurteilung dankbar.

```

01+LBL "TRI"
02 "EINGABE:"
03 "a=A"
04 "b=B"
05 "c=C"
06 "AL=F"
07 "BE=G"
08 "GA=H"
09 39
10 PSIZE
11 "radasadolftzsr"
12 CLA
13 11
14 PASN
15 12
16 PASN
17 13
18 PASN
19 21
20 PASN
21 22
22 PASN
23 23
24 PASN
25 ZREG 01
26 CLS
27 TONE 9
28 "DATEN ? TRI"
29 PROMPT
30+LBL "Δ"
31 TONE 5
32 STOP
33 GTO 00
34+LBL A
35 STO 01
36 GTO "Δ"
37+LBL B
    
```

```

38 STO 02
39 GTO "Δ"
40+LBL C
41 STO 03
42 GTO "Δ"
43+LBL F
44 STO 04
45 GTO "Δ"
46+LBL G
47 STO 05
48 GTO "Δ"
49+LBL H
50 STO 06
51 GTO "Δ"
52+LBL 00
53 ΣREG 07
54 CLS
55 ΣREG 13
56 CLS
57 ΣREG 19
58 CLS
59 ΣREG 26
60 CLS
61 CF 08
62 0
63 X<>F
64 RCL 01
65 X=0?
66 SF 01
67 RCL 02
68 X=0?
69 SF 02
70 RCL 03
71 X=0?
72 SF 03
73 RCL 04
74 X=0?
75 SF 04
76 RCL 05
77 X=0?
78 SF 05
79 RCL 06
80 X=0?
81 SF 06
82 X<>F
83 STO 00
84 X<>F
85 CLA
86 FS? 01
87 " a"
88 FS? 02
89 "t b"
90 FS? 03
91 "t c"
92 FS? 04
93 "t AL"
94 FS? 05
95 "t BE"
96 FS? 06
97 "t GA"
98 ASTO 33
99 ASHF
    
```

# SERIE 40

|                |            |            |                    |                |
|----------------|------------|------------|--------------------|----------------|
| 100 ASTO 34    | 162 X*Y?   | 224 RCL 18 | 286 XEQ 20         | 348 RTN        |
| 101 CLA        | 163 GTO 00 | 225 STO 10 | 287 STO 20         | 349*LBL 17     |
| 102 FS? 01     | 164 RCL 14 | 226 XEQ 16 | 288 XEQ 23         | 350 RCL 00     |
| 103 "a"        | 165 STO 07 | 227 STO 23 | 289 STO 25         | 351 112        |
| 104 FS? 02     | 166 RCL 13 | 228 XEQ 18 | 290 XEQ 15         | 352 X*Y?       |
| 105 "b"        | 167 STO 08 | 229 STO 22 | 291 STO 29         | 353 GTO 00     |
| 106 FS? 03     | 168 RCL 17 | 230 XEQ 20 | 292 XEQ 18         | 354 013.007006 |
| 107 "c"        | 169 STO 10 | 231 STO 19 | 293 STO 30         | 355 REGMOVE    |
| 108 FS? 04     | 170 XEQ 16 | 232 XEQ 23 | 294 XEQ 20         | 356 XEQ 16     |
| 109 "AL"       | 171 STO 22 | 233 STO 25 | 295 STO 27         | 357 STO 22     |
| 110 FS? 05     | 172 XEQ 18 | 234 XEQ 15 | 296 XEQ 23         | 358 ENTER↑     |
| 111 "BE"       | 173 STO 24 | 235 STO 30 | 297 STO 32         | 359 RCL 07     |
| 112 FS? 06     | 174 XEQ 20 | 236 XEQ 18 | 298 GTO 22         | 360 X↑2        |
| 113 "GA"       | 175 STO 21 | 237 STO 29 | 299*LBL 15         | 361 RCL 09     |
| 114 ASTO 35    | 176 XEQ 23 | 238 XEQ 20 | 300 100            | 362 X↑2        |
| 115 ASHF       | 177 STO 25 | 239 STO 26 | 301 RCL 11         | 363 +          |
| 116 ASTO 36    | 178 XEQ 15 | 240 XEQ 23 | 302 -              | 364 RCL 08     |
| 117 CLA        | 179 STO 29 | 241 STO 32 | 303 STO 11         | 365 X↑2        |
| 118 FC? 01     | 180 XEQ 18 | 242 GTO 22 | 304 RTN            | 366 -          |
| 119 "a"        | 181 STO 31 | 243*LBL 00 | 305*LBL 16         | 367 RCL 07     |
| 120 FC? 02     | 182 XEQ 20 | 244 RCL 00 | 306 RCL 10         | 368 RCL 09     |
| 121 "b"        | 183 STO 28 | 245 100    | 307 90             | 369 *          |
| 122 FC? 03     | 184 XEQ 23 | 246 X*Y?   | 308 X<=Y?          | 370 2          |
| 123 "c"        | 185 STO 32 | 247 GTO 00 | 309 GTO 01         | 371 *          |
| 124 FC? 04     | 186 GTO 22 | 248 RCL 13 | 310 RCL 07         | 372 /          |
| 125 "AL"       | 187*LBL 00 | 249 STO 07 | 311 RCL 08         | 373 ACOS       |
| 126 FC? 05     | 188 RCL 00 | 250 RCL 15 | 312 X<=Y?          | 374 STO 11     |
| 127 "BE"       | 189 02     | 251 STO 08 | 313 GTO 01         | 375 STO 23     |
| 128 FC? 06     | 190 X*Y?   | 252 RCL 16 | 314 TONE 1         | 376 XEQ 18     |
| 129 "GA"       | 191 GTO 00 | 253 STO 10 | 315 SF 08          | 377 STO 24     |
| 130 ASTO 37    | 192 RCL 14 | 254 XEQ 16 | 316 "ES GIBT ZWEI" | 378 GTO 23     |
| 131 ASHF       | 193 STO 07 | 255 STO 24 | 317 AVIEW          | 379*LBL 00     |
| 132 ASTO 38    | 194 RCL 15 | 256 XEQ 18 | 318 PSE            | 380 RCL 00     |
| 133 001.013006 | 195 STO 08 | 257 STO 23 | 319 CLA            | 381 04         |
| 134 REGSWAP    | 196 RCL 17 | 258 XEQ 20 | 320 " LOESUNGEN"   | 382 X*Y?       |
| 135*LBL 00     | 197 STO 10 | 259 STO 20 | 321 AVIEW          | 383 GTO 00     |
| 136 RCL 00     | 198 XEQ 16 | 260 XEQ 23 | 322 PSE            | 384 013.007006 |
| 137 104        | 199 STO 24 | 261 STO 25 | 323 CLA            | 385 REGSWAP    |
| 138 X*Y?       | 200 XEQ 18 | 262 XEQ 15 | 324 ARCL 33        | 386 XEQ 03     |
| 139 GTO 00     | 201 STO 22 | 263 STO 31 | 325 ARCL 34        | 387 STO 20     |
| 140 013.007006 | 202 XEQ 20 | 264 XEQ 18 | 326 AVIEW          | 388 XEQ 16     |
| 141 REGMOVE    | 203 STO 19 | 265 STO 30 | 327 CLA            | 389 STO 22     |
| 142 XEQ 16     | 204 XEQ 23 | 266 XEQ 20 | 328 PSE            | 390 XEQ 18     |
| 143 STO 23     | 205 STO 25 | 267 STO 27 | 329 "UND"          | 391 STO 24     |
| 144 XEQ 18     | 206 XEQ 15 | 268 XEQ 23 | 330 AVIEW          | 392 GTO 23     |
| 145 STO 24     | 207 STO 31 | 269 STO 32 | 331 PSE            | 393*LBL 00     |
| 146 XEQ 20     | 208 XEQ 18 | 270 GTO 22 | 332 CLA            | 394 RCL 00     |
| 147 STO 21     | 209 STO 29 | 271*LBL 00 | 333 " "            | 395 98         |
| 148 XEQ 23     | 210 XEQ 20 | 272 RCL 00 | 334 ARCL 35        | 396 X*Y?       |
| 149 STO 25     | 211 STO 26 | 273 52     | 335 ARCL 36        | 397 GTO 00     |
| 150 XEQ 15     | 212 XEQ 23 | 274 X*Y?   | 336 TONE 0         | 398 RCL 15     |
| 151 STO 30     | 213 STO 32 | 275 GTO 17 | 337 TONE 1         | 399 STO 07     |
| 152 XEQ 18     | 214 GTO 22 | 276 RCL 15 | 338 PROMPT         | 400 RCL 14     |
| 153 STO 31     | 215*LBL 00 | 277 STO 07 | 339*LBL 01         | 401 STO 09     |
| 154 XEQ 20     | 216 RCL 00 | 278 RCL 13 | 340 RCL 10         | 402 RCL 16     |
| 155 STO 28     | 217 50     | 279 STO 08 | 341 SIN            | 403 STO 11     |
| 156 XEQ 23     | 218 X*Y?   | 280 RCL 18 | 342 RCL 08         | 404 XEQ 03     |
| 157 STO 32     | 219 GTO 00 | 281 STO 10 | 343 *              | 405 STO 19     |
| 158 GTO 22     | 220 RCL 15 | 282 XEQ 16 | 344 RCL 07         | 406 XEQ 16     |
| 159*LBL 00     | 221 STO 07 | 283 STO 22 | 345 /              | 407 STO 24     |
| 160 RCL 00     | 222 RCL 14 | 284 XEQ 18 | 346 ASIN           | 408 XEQ 18     |
| 161 08         | 223 STO 08 | 285 STO 23 | 347 STO 11         | 409 STO 23     |

|                |            |                |                   |              |
|----------------|------------|----------------|-------------------|--------------|
| 410 GTO 23     | 472 XEQ 18 | 534 RCL 00     | 596 RCL 14        | 658 *        |
| 411+LBL 00     | 473 STO 24 | 535 22         | 597 STO 07        | 659 RCL 10   |
| 412 RCL 00     | 474 XEQ 20 | 536 X*Y?       | 598 RCL 16        | 660 SIN      |
| 413 56         | 475 STO 21 | 537 GTO 00     | 599 STO 11        | 661 /        |
| 414 X*Y?       | 476 XEQ 21 | 538 RCL 15     | 600 RCL 18        | 662 STO 09   |
| 415 GTO 00     | 477 STO 20 | 539 STO 07     | 601 STO 12        | 663 RTN      |
| 416 RCL 13     | 478 GTO 23 | 540 RCL 18     | 602 XEQ 19        | 664+LBL 21   |
| 417 STO 07     | 479+LBL 00 | 541 STO 10     | 603 STO 23        | 665 RCL 11   |
| 418 RCL 14     | 480 RCL 00 | 542 RCL 17     | 604 XEQ 20        | 666 SIN      |
| 419 STO 09     | 481 44     | 543 STO 11     | 605 STO 21        | 667 RCL 07   |
| 420 RCL 18     | 482 X*Y?   | 544 XEQ 18     | 606 XEQ 21        | 668 *        |
| 421 STO 11     | 483 GTO 00 | 545 STO 22     | 607 STO 19        | 669 RCL 10   |
| 422 XEQ 03     | 484 RCL 13 | 546 XEQ 20     | 608 GTO 23        | 670 SIN      |
| 423 STO 21     | 485 STO 07 | 547 STO 19     | 609+LBL 00        | 671 /        |
| 424 XEQ 16     | 486 RCL 16 | 548 XEQ 21     | 610 RCL 00        | 672 STO 08   |
| 425 STO 22     | 487 STO 10 | 549 STO 20     | 611 70            | 673 RTN      |
| 426 XEQ 18     | 488 RCL 18 | 550 GTO 23     | 612 X*Y?          | 674+LBL 22   |
| 427 STO 23     | 489 STO 11 | 551+LBL 00     | 613 GTO 00        | 675+LBL "ZZ" |
| 428 GTO 23     | 490 XEQ 18 | 552 RCL 00     | 614 RCL 15        | 676 RCL 19   |
| 429+LBL 16     | 491 STO 23 | 553 38         | 615 STO 07        | 677 X=0?     |
| 430 RCL 08     | 492 XEQ 20 | 554 X*Y?       | 616 RCL 16        | 678 GTO 00   |
| 431 X+2        | 493 STO 20 | 555 GTO 00     | 617 STO 11        | 679 " a= "   |
| 432 RCL 09     | 494 XEQ 21 | 556 RCL 15     | 618 RCL 17        | 680 ARCL 19  |
| 433 X+2        | 495 STO 21 | 557 STO 07     | 619 STO 12        | 681 PROMPT   |
| 434 +          | 496 GTO 23 | 558 RCL 18     | 620 XEQ 19        | 682 CLA      |
| 435 RCL 07     | 497+LBL 00 | 559 STO 10     | 621 STO 24        | 683+LBL 00   |
| 436 X+2        | 498 RCL 00 | 560 RCL 16     | 622 XEQ 20        | 684 RCL 20   |
| 437 -          | 499 74     | 561 STO 11     | 623 STO 20        | 685 X=0?     |
| 438 RCL 08     | 500 X*Y?   | 562 XEQ 18     | 624 XEQ 21        | 686 GTO 00   |
| 439 RCL 09     | 501 GTO 00 | 563 STO 23     | 625 STO 19        | 687 " b= "   |
| 440 *          | 502 RCL 14 | 564 XEQ 20     | 626 GTO 23        | 688 ARCL 20  |
| 441 2          | 503 STO 07 | 565 STO 20     | 627+LBL 00        | 689 PROMPT   |
| 442 *          | 504 RCL 17 | 566 XEQ 21     | 628 BEEP          | 690 CLA      |
| 443 /          | 505 STO 10 | 567 STO 19     | 629 "PRUEFEN SIE" | 691+LBL 00   |
| 444 ACOS       | 506 RCL 16 | 568 GTO 23     | 630 AVIEW         | 692 RCL 21   |
| 445 STO 10     | 507 STO 11 | 569+LBL 18     | 631 PSE           | 693 X=0?     |
| 446 RTN        | 508 XEQ 18 | 570 180        | 632 PSE           | 694 GTO 00   |
| 447+LBL 03     | 509 STO 24 | 571 RCL 10     | 633 CLA           | 695 " c= "   |
| 448 RCL 07     | 510 XEQ 20 | 572 RCL 11     | 634 "DIE EINGABE" | 696 ARCL 21  |
| 449 X+2        | 511 STO 21 | 573 +          | 635 AVIEW         | 697 PROMPT   |
| 450 RCL 09     | 512 XEQ 21 | 574 -          | 636 PSE           | 698 CLA      |
| 451 X+2        | 513 STO 19 | 575 STO 12     | 637 PSE           | 699+LBL 00   |
| 452 +          | 514 GTO 23 | 576 RTN        | 638 CLA           | 700 RCL 22   |
| 453 RCL 07     | 515+LBL 00 | 577+LBL 00     | 639 "+"           | 701 X=0?     |
| 454 RCL 09     | 516 RCL 00 | 578 RCL 00     | 640 ARCL 37       | 702 GTO 00   |
| 455 *          | 517 26     | 579 28         | 641 ARCL 38       | 703 " AL= "  |
| 456 2          | 518 X*Y?   | 580 X*Y?       | 642 "I?"          | 704 ARCL 22  |
| 457 *          | 519 GTO 00 | 581 GTO 00     | 643 PROMPT        | 705 PROMPT   |
| 458 RCL 11     | 520 RCL 14 | 582 013.007006 | 644 CLA           | 706 CLA      |
| 459 COS        | 521 STO 07 | 583 REMOVE     | 645 GTO "TRI"     | 707+LBL 00   |
| 460 *          | 522 RCL 17 | 584 XEQ 19     | 646+LBL 19        | 708 RCL 23   |
| 461 -          | 523 STO 10 | 585 STO 22     | 647 180           | 709 X=0?     |
| 462 SQRT       | 524 RCL 18 | 586 XEQ 20     | 648 RCL 11        | 710 GTO 00   |
| 463 STO 08     | 525 STO 11 | 587 STO 21     | 649 RCL 12        | 711 " BE= "  |
| 464 RTN        | 526 XEQ 18 | 588 XEQ 21     | 650 +             | 712 ARCL 23  |
| 465+LBL 00     | 527 STO 22 | 589 STO 20     | 651 -             | 713 PROMPT   |
| 466 RCL 00     | 528 XEQ 20 | 590 GTO 23     | 652 STO 10        | 714 CLA      |
| 467 76         | 529 STO 19 | 591+LBL 00     | 653 RTN           | 715+LBL 00   |
| 468 X*Y?       | 530 XEQ 21 | 592 RCL 00     | 654+LBL 20        | 716 RCL 24   |
| 469 GTO 00     | 531 STO 21 | 593 42         | 655 RCL 12        | 717 X=0?     |
| 470 013.007006 | 532 GTO 23 | 594 X*Y?       | 656 SIN           | 718 GTO 00   |
| 471 REMOVE     | 533+LBL 00 | 595 GTO 00     | 657 RCL 07        | 719 " GA= "  |

|              |               |               |              |
|--------------|---------------|---------------|--------------|
| 720 ARCL 24  | 741 GTO 00    | 762*LBL 00    | 783 CLA      |
| 721 PROMPT   | 742 " b'=" "  | 763 RCL 30    | 784 GTO "44" |
| 722 CLA      | 743 ARCL 27   | 764 X=0?      | 785*LBL 23   |
| 723*LBL 00   | 744 PROMPT    | 765 GTO 00    | 786 RCL 11   |
| 724 RCL 25   | 745 CLA       | 766 " BE'=" " | 787 SIN      |
| 725 " FL=" " | 746*LBL 00    | 767 ARCL 30   | 788 RCL 07   |
| 726 ARCL 25  | 747 RCL 28    | 768 PROMPT    | 789 RCL 09   |
| 727 PROMPT   | 748 X=0?      | 769 CLA       | 790 *        |
| 728 CLA      | 749 GTO 00    | 770*LBL 00    | 791 *        |
| 729 FC? 08   | 750 " c'=" "  | 771 RCL 31    | 792 2        |
| 730 GTO "44" | 751 ARCL 28   | 772 X=0?      | 793 /        |
| 731 RCL 26   | 752 PROMPT    | 773 GTO 00    | 794 FS? 08   |
| 732 X=0?     | 753 CLA       | 774 " GA'=" " | 795 RTN      |
| 733 GTO 00   | 754*LBL 00    | 775 ARCL 31   | 796 STO 25   |
| 734 " a'=" " | 755 RCL 29    | 776 PROMPT    | 797 GTO "44" |
| 735 ARCL 26  | 756 X=0?      | 777 CLA       | 798 END      |
| 736 PROMPT   | 757 GTO 00    | 778*LBL 00    |              |
| 737 CLA      | 758 " AL'=" " | 779 RCL 32    |              |
| 738*LBL 00   | 759 ARCL 29   | 780 " FL'=" " |              |
| 739 RCL 27   | 760 PROMPT    | 781 ARCL 32   |              |
| 740 X=0?     | 761 CLA       | 782 PROMPT    |              |

Adolf Radas  
Prielmayerstr. 20  
8058 Erding

**Startadressen von Files**

**FLHD**

**File Header**

**55 Zeilen, 99 Bytes, 15 Regs., SIZE 000, HP41C**

„FLHD“ berechnet die Startadresse, d.h. die Adresse des 1. Headerregisters eines beliebigen Files im X-Memory. Im Gegensatz zu früheren Programmen dieser Art (z.B. den verschiedenen Versionen von „?HR“; siehe PRISMA 83.7.21, 86.2.86 und 86.5.18) funktioniert „FLHD“ ohne jede Einschränkung, also auch bei Files im Speicherbereich der X-Memory-Module.

**Anwendung:**

- 1) Eingabe des Filenamens ins Alpha-Register
- 2) XEQ „FLHD“
- 3) Ausgabe der Startadresse im X-Register

Die Inhalte des Stacks und des Last-X-Registers werden verändert; der eingegebene Filename bleibt erhalten.

Mögliche Fehlermeldung: FL NOT FOUND (Zeile 2) – der angegebene File ist nicht vorhanden.

**Funktion des Unterprogramms „EMA-“**

Das eigene globale Label „EMA-“ (extended memory adress subtract) ermöglicht es, dieses Unterprogramm auch separat aufzurufen. „EMA-“ berechnet aus der Adresse A<sub>1</sub> eines X-Memory-Registers und der relativen Registerdifferenz ΔR die Adresse A<sub>2</sub> desjenigen Registers, das im X-Memory um ΔR Register „tiefer“ liegt als A<sub>1</sub>, wobei sich der Anwender keine Gedanken darüber machen muß, ob A<sub>2</sub> im selben X-Memory-Bereich wie A<sub>1</sub> liegt oder nicht.

Neben A<sub>1</sub> und ΔR benötigt „EMA-“ noch einen weiteren Eingabeparameter, nämlich – abhängig von A<sub>1</sub> – die Adresse S<sub>1</sub> eines

der drei X-Memory-Statusregister. Welcher Wert hierfür jeweils erforderlich ist, kann man der folgenden Tabelle entnehmen:

| A          | S   |
|------------|-----|
| 191 – 65   | 64  |
| 751 – 514  | 513 |
| 1007 – 770 | 769 |

Außer A<sub>2</sub> wird auch die dazugehörige Statusregisteradresse S<sub>2</sub> berechnet, so daß man bei anschließenden Ausführungen von „EMA-“ nur noch den neuen Wert für ΔR eingeben muß.

**Anwendung von „EMA-“**

- 1) Eingaben – Z: S<sub>1</sub>  
                  Y: A<sub>1</sub>  
                  X: ΔR
- 2) XEQ „EMA-“
- 3) Ausgaben – Y: S<sub>2</sub>  
                  X: A<sub>2</sub>

ΔR darf nicht negativ sein!  
Wenn ΔR so groß ist, daß das letzte vorhandene X-Memory-Register übersprungen wird, statt A<sub>2</sub> eine Zahl ≤ 0 ausgegeben. Diese Zahl gibt an, um wieviele Register unterhalb des letzten vorhandenen X-Memory-Statusregisters man gelandet ist.

„EMA-“ verändert die Inhalte des Stacks und des Last-X-Registers.

**Anwendungsbeispiel:**

Da „FLHD“ außer der File-Startadresse auch die dazugehörige Statusregisteradresse (im Y-Register) ausgibt, kann man die Adresse des 2. Headerregisters eines Files folgendermaßen berechnen:

(Filename ins Alpha-Register schreiben)  
XEQ „FLHD“  
1  
XEQ „EMA-“

|               |           |
|---------------|-----------|
| 01*LBL "FLHD" | 29 RTN    |
| 02 FLSIZE     | 30 X<>Y   |
| 03 64         | 31 -      |
| 04 STO \      | 32 LASTX  |
| 05 5          | 33 2      |
| 06 XEQ 14     | 34 XEQ 14 |
| 07 X<> \      | 35 +      |
| 08 191        | 36 LASTX  |
| 09*LBL 01     | 37 238    |
| 10 DSE \      | 38 -      |
| 11 GTO 00     | 39 X<>Y   |
| 12 0          | 40 0      |
| 13 STO J      | 41 GTO 13 |
| 14 RDN        | 42*LBL 14 |
| 15 RTN        | 43 A+B    |
| 16*LBL 00     | 44 PEEKB  |
| 17 PEEKR      | 45 X<>Y   |
| 18 STO J      | 46 A-     |
| 19 CLX        | 47 PEEKB  |
| 20 FLSIZE     | 48 X<>Y   |
| 21 2          | 49 CLX    |
| 22 +          | 50 16     |
| 23 XEQ 13     | 51 ST* Z  |
| 24 GTO 01     | 52 /      |
| 25*LBL "EMA-" | 53 INT    |
| 26*LBL 13     | 54 +      |
| 27 -          | 55 END    |
| 28 X>Y?       |           |

Michael Kamp  
Martin-Luther-Straße 48  
5142 Hückelhoven

# COMPLEX

## Abspeichern von komplexen Zahlen

101 Zeilen, 220 Bytes, 32 Regs, SIZE 000  
HP41C, X-F/M, CCD-Modul

Uwe Mönkedieck (2797) hat mit seinem Beitrag in PRISMA 86.6.31 auf eine Idee gebracht, die ich hier vorstellen möchte.

Bei seiner Methode der Abspeicherung von komplexen Zahlen im X-Memory fällt auf, daß neben den eigentlichen 20 Datenregistern weitere 20 Headerregister benötigt werden. Mit dem CCD-Modul geht es eleganter und platzsparender.

Die komplexen Zahlen werden in einem zweidimensionalen Feld „Z“ abgelegt, welches zuvor mit 16,002 MDIM eingerichtet wird. Hierin finden 16 komplexe Zahlen Platz.

Die beiden Funktionen  $T^R(CL)$  und  $T^S(TO)$  sind den umgeschalteten Zifferntasten 7 (R) bzw. 8 (S) zugewiesen (vgl. auch CAT'6): Beim Aufruf dieser beiden Funktionen wird mittels PMTH die Registernummer abgefragt. Bei 4...WSIZE...PMTH sind die Eingabewerte 0 bis F möglich, wodurch 16 verschiedene Register angesprochen werden können. Wird WSIZE sowie die Matrix „Z“ vergrößert, so können auch wesentlich mehr Register angesprochen werden. Da die Funktion  $IJ=A$  bei den X-Werten 0 und 1 den Zeiger auf die 1. Zeile des Feldes positioniert, wird der PMTH-Eingabewert inkrementiert. Hinter ISG X befindet sich das NOP F0 (Zeile 13 und 28). Die Zeilen 22-24 und 31-33 stellen sicher, daß die Stackinhalte von Z und T erhalten bleiben. Auf diese Weise können sich zwei komplexe Zahlen im Stack befinden, ohne daß beim Abspeichern etwas verloren geht. Entsprechend wird beim Rückkauf einer 2. komplexen Zahl die 1. nach Z bzw. T verschoben.

Das Programm ermöglicht darüber hinaus die elementaren Berechnungen wie Addition, Subtraktion, Multiplikation, Division sowie Reziprokwertbildung. Außerdem können die komplexen Zahlen in die Zeigerdarstellung umgerechnet werden und umgekehrt. Diese beiden Funktionen habe ich den umgeschalteten Zifferntasten 5 (P-R) bzw. 6 (R-P) zugewiesen.

**Hinweise:** Bei allen Berechnungen wird davon ausgegangen, daß der Imaginärteil der Zahl im Stack über dem Realteil der Zahl steht. Die Berechnungen werden zwar zum Teil in der Zeigerdarstellung der komplexen Zahl durchgeführt, jedoch geht das Programm davon aus, daß die Zahlen als Realteil und Imaginärteil vorliegen.

```

01*LBL "COMPLEX"      41 "XY:"
02 FIX 3              42 ARCL X
03 4                  43 "I "
04 WSIZE              44 ARCL Y
05 16.002             45 AVIEW
06 "Z"                46 RTN
07 MDIM               47*LBL "C+"
08 STOP              48 X<>Y
09*LBL "R"            49*LBL 01
10 "RCL"              50 RDN
11 PMTH               51 +
12 ISG X              52 RDN
13 ""                 53 +
14 "Z"                54 R↑
15 IJ=A              55 GTO 00
16 RDN                56*LBL "C-"
17 R>+                57 CHS
18 R>+                58 X<>Y
19 X<>Y               59 CHS
20 RTN                60 GTO 01
21*LBL "S"            61*LBL "1/C"
22 R↑                 62 XEQ 02
23 STO L              63 GTO 00
24 RDN                64*LBL 02
25 "STO"              65 R-P
26 PMTH               66 1/X
27 ISG X              67 X<>Y
28 ""                 68 CHS
29 "Z"                69 X<>Y
30 IJ=A              70 P-R
31 CLX                71 RTN
32 LASTX              72*LBL "C*"
33 RDN                73 R-P
34 >R+                74 RDN
35 X<>Y               75 RDN
36 >R+                76 R-P
37 X<>Y               77 RDN
38 RTN                78 RDN
39*LBL 00              79 X<>Y
40 SF 21              80 RDN
    
```

```

81 *
82 RDN
83 +
84 R↑
85 P-R
86 GTO 00
87*LBL "C/"
88 XEQ 02
89 GTO "C*"
90*LBL "Z<-XY"
91 P-R
92 GTO 00
93*LBL "XY-Z<"
94 R-P
95 "Z<:"
96 ARCL X
97 "I "
98 ARCL Y
99 "I"
100 AVIEW
101 .END.
    
```

### Beispiel

```

3,000 ENTER↑
4,000
XEQ "S"
STO'0
5,000 ENTER↑
6,000
XEQ "S"
STO'1
XEQ "C+"
XY:9,000 38,000
XEQ "R"
RCL'0
XEQ "R"
RCL'1
XEQ "C/"
XY:0,639 -0,033
XEQ "R"
RCL'0
XEQ "R"
RCL'1
XEQ "C*"
XY:9,000 38,000
XEQ "XY-Z<"
Z<:39,051 76,675'
XEQ "Z<-XY"
XY:9,000 38,000
XEQ "1/C"
XY:0,006 -0,025
XEQ "XY-Z<"
Z<:0,026 -76,675'
    
```

### Tastenzuweisungen

```

↑1/C 12
↑C--51
↑R-52
↑S-53
↑C+-61
↑Z<-XY-63
↑XY-Z<-64
↑C*-71
↑C/-81
    
```

Achim Grohs  
Auf dem Berlich 26  
5000 Köln

# Korrektur zur XSPLINE

1325 Zeilen, 2096 Bytes, 300 Regs., SIZE 020, HP41CV, X-F/M, (JC), PRINTER

Ich habe in der Zwischenzeit an dem in Nr. 2 veröffentlichten Programm 3 kleine Änderungen angebracht, die im Listing enthalten sind:

- Zeile 061 erweitert den y-Maßstab im Falle ausschwingender Splines
- Zeile 105/108 Änderung am Display; beseitigt Schwierigkeiten bei Eingabe irrationaler usw. Zahlen
- Zeile 1252 ff (eingerahmter Bereich) im Falle periodischer Splines (MASS 4) können bei der Iteration auftretende Endlos-Oszillationen verhindert werden.

„Änderungen zu ‘SPL’“

```
104 PROMPT
105 FIX 4
106 ARCL X
107 PRA
108 RND
109 STO 06
```

```
60 -
61 4
62 /
```

```
1252 FC? 04
1253 GTO 17
1254 RCL 14
1255 2
1256 -
1257 E
1258 FIX 5
1259 GTO IND
12 12
1260*LBL 01
1261 ST+ 12
1262 +
1263 RCL 04
1264 SEEKPT
1265 +
1266 GETX
1267 X<>Y
1268 SEEKPT
1269 RDN
↓ 1270 ENTER↑
1271 ENTER↑
1272 GETX
1273 ST+ Z
1274 -
1275 RND
```

```
1276 X=0?
1277 GTO 17
1278 RDN
↑ 1279 GTO 00
1280*LBL 02
1281 ST- 12
1282 RDN
1283 SEEKPT
1284 GETX
1285 3
1286 *
1287 XEQ 01
1288 STO 01
1289 RDN
1290 XEQ 01
1291 2
1292 *
1293 RCL 15
1294 E
1295 -
1296 SEEKPT
1297 RDN
1298 GETX
1299 *
1300 X<>Y
1301 LASTX
1302 X↑2
1303 *
1304 +
1305 RCL 01
1306 +
1307 RCL 03
1308 SEEKPT
1309 RDN
1310 GETX
↓ 1311 +
1312 2
↑ 1313 /
1314*LBL 00
1315 STO 06
1316 STO 07
1317 RCL 14
1318 GTO 20
1319*LBL 25
1320 ST+ 03
1321 ST+ 04
1322 ST+ 05
1323 ST+ 11
1324 ST+ 13
1325 .END.
```

Im übrigen habe ich einen Spline mit 5 Stützstellen durchgerechnet. MASS=1 A = 1.5 B = 2.5 / Koeffizienten (B) / Plot (C) Werte (D) für die Enden und an einem Übergang / Nullstellen (E) von y' u. y".

Beispiel zu ‘SPL’

```
XEQ "ST"
N=?
5 RUN
FILE?
FFM RUN
X1=
-3 RUN
Y1=
6.5 RUN
X2=
-1.5 RUN
Y2=
4.25 RUN
X3=
.5 RUN
Y3=
2 RUN
X4=
3 RUN
Y4=
.25 RUN
X5=
6 RUN
Y5=
-2.5 RUN
MASS?
1 RUN
A?
1.5 RUN
B?
2.5 RUN
FERTIG!
XEQ "KO"
Spline 1
D1= 0.9529
C1=-3.4293
B1= 1.5000
A1= 6.5000
Spline 2
D2=-0.1216
C2= 0.8587
B2=-2.3560
A2= 4.2500
Spline 3
D3=-0.1028
C3= 0.1292
B3=-0.3804
A3= 2.0000
Spline 4
D4= 0.2968
C4=-0.6419
B4=-1.6622
A4= 0.2500
ENDE
```

X INC? XEQ "P!"  
 -36 UN  
 PLOT C  
 X <UNITS-  
 Y <UNITS= 1/4  
 -4.75 8.75  
 2.00  
 -----  
 -3.00 | x  
 -2.75 | x  
 -2.50 | x  
 -2.25 | x  
 -2.00 | x  
 -1.75 | x  
 -1.50 | x  
 -1.25 | x  
 -1.00 | x  
 -0.75 | x  
 -0.50 | x  
 -0.25 | x  
 0.00 | x  
 0.25 | x  
 0.50 | x  
 0.75 | x  
 1.00 | x  
 1.25 | x  
 1.50 | x  
 1.75 | x  
 2.00 | x  
 2.25 | x  
 2.50 | x  
 2.75 | x  
 3.00 | x  
 3.25 | x  
 3.50 | x  
 3.75 | x  
 4.00 | x  
 4.25 | x  
 4.50 | x  
 4.75 | x  
 5.00 | x  
 5.25 | x  
 5.50 | x  
 5.75 | x  
 6.00 | x

FERTIG!

XEQ "WE"  
 X  
 -3 RUN  
 X = -3.0000  
 Y = 6.5000  
 Y' = 1.5000  
 Y'' = -6.8587  
 KR = -1.1706

XEQ "WE"  
 X  
 6 RUN  
 X = 6.0000  
 Y = -2.5000  
 Y' = 2.5000  
 Y'' = 4.0585  
 KR = 0.2079

XEQ "WE"  
 X  
 -1.5001 RUN  
 X = -1.5001  
 Y = 4.2502  
 Y' = -2.3562  
 Y'' = 1.7168  
 KR = 0.1024

XEQ "WE"  
 X  
 -1.4999 RUN  
 X = -1.4999  
 Y = 4.2498  
 Y' = -2.3558  
 Y'' = 1.7173  
 KR = 0.1024

XEQ "NU"  
 Y:0 Y':1 Y'':2  
 1 RUN  
 SOLL?  
 0 RUN  
 SOLL? 0.0000  
 X  
 -2.7500 RUN  
 X = -2.7500  
 Y = 6.6756  
 Y' = -0.0360  
 Y'' = -5.4293

X = -2.7566  
 Y = 6.6757  
 Y' = 0.0001  
 Y'' = -5.4672

X = -2.7566  
 Y = 6.6757  
 Y' = 0.0000  
 Y'' = -5.4671  
 X = -2.7566

FERTIG!

XEQ "NU"  
 Y:0 Y':1 Y'':2  
 1 RUN  
 SOLL?  
 0 RUN  
 SOLL? 0.0000  
 X  
 5.2500 RUN

X = 5.2500  
 Y = -3.3587  
 Y' = -0.0431  
 Y'' = 2.7230

X = 5.2658  
 Y = -3.3591  
 Y' = 0.0003  
 Y'' = 2.7512

X = 5.2657  
 Y = -3.3591  
 Y' = 0.0000  
 Y'' = 2.7510

X = 5.2657

FERTIG!  
 XEQ "NU"  
 Y:0 Y':1 Y'':2  
 2 RUN  
 SOLL?  
 0 RUN  
 SOLL? 0.0000

X  
 -1.5000 RUN  
 X = -1.5000  
 Y = 4.2500  
 Y' = -2.3560  
 Y'' = 1.7173

X = -1.8004  
 Y = 5.0093  
 Y' = -2.6139  
 Y'' = 0.0000

X = -1.8004

FERTIG!  
 XEQ "NU"  
 Y:0 Y':1 Y'':2  
 2 RUN  
 SOLL?  
 0 RUN  
 SOLL? 0.0000

X  
 .5000 RUN  
 X = 0.5000  
 Y = 2.0000  
 Y' = -0.3804  
 Y'' = 0.2583

X = 0.8541  
 Y = 1.8770  
 Y' = -0.3276  
 Y'' = 0.0399

X = 0.9188  
 Y = 1.8558  
 Y' = -0.3263  
 Y'' = 0.0000

**X = 0.9188**

FERTIG!

```

XEQ "NU"
Y:0 Y':1 Y'':2
2 RUN
SOLL?
0 RUN
SOLL? 0.0000
    
```

```

X
3.0000 RUN
X = 3.0000
Y = 0.2500
Y' = -1.6622
Y'' = -1.2838
    
```

X = 0.9187  
 Y = 1.8558  
 Y' = -0.3263  
 Y'' = 0.0000

**X = 0.9187**

FERTIG!

```

XEQ "NU"
Y:0 Y':1 Y'':2
2 RUN
SOLL?
0 RUN
SOLL? 0.0000
    
```

```

X
3.1000 RUN
X = 3.1000
Y = 0.0777
Y' = -1.7816
Y'' = -1.1057
    
```

X = 3.7209  
 Y = -1.1707  
 Y' = -2.1249  
 Y'' = 0.0000

**X = 3.7209**

FERTIG!

**Listing**

```

01*LBL "SPL"
02*LBL "IN"
"A ST ART+SPLINE" PRA
"a NE UE RANDWER" "FTE"
PRA "B KO EFFIZIENTE"
"FN" PRA "C PL OT"
PRA "D WE RTE: Y Y' "
"FY" 1/p" PRA
"d XY EDIT" PRA
"E NU LLSTELLEN" PRA
"I IN FO" PRA RTN
    
```

```

23*LBL "PL"
RCL 14 ENTER↑ ENTER↑
6 * E - SEEKPT
X<>Y GETX STO 00
STO 01 DSE Y
    
```

```

37*LBL 05
RDN GETX RCL 01 X<=Y?
X<>Y STO 01 RDN
RCL 00 X>Y? X<>Y
STO 00 RDN DSE Y
GTO 05 RCL 00 RCL 01
+ 2 / STO 04 RCL 01
RCL 00 - 4 / ST+ 01
ST- 00 CLX STO 03
BEEP "X INC?" PROMPT
STO 10 "SPLINE"
ASTO 11 SF 01 SF 00
XRDM "PRPLOTP" GTO 17
    
```

```

77*LBL "SPLINE"
RCL 15 SEEKPT X<>Y
STO 12 FS?C 00 GTO 06
RCL 16 X<=Y? GTO 06
RDN RCL 13 X>Y?
GTO 06 RDN RCL 13 -
STO 12 GTO 00
    
```

```

96*LBL "NU"
"Y:0 Y':1 Y'':2" PROMPT
CHS STO 11 5 ST+ 11
"SOLL? " PROMPT FIX 4
ARCL X PRA RND STO 06
GTO 16
    
```

```

111*LBL "WE"
SF 03
    
```

```

113*LBL 16
ADV RCL 15 SEEKPT
"X " FC? 02 PROMPT
FS?C 02 RCL 07 STO 12
STO 07 RCL 08 X>Y?
GTO 02 RDN RCL 09
X<>Y GTO 02 RDN
XEQ 19
    
```

```

133*LBL 06
GETX RCL 12 X>Y?
GTO 06 RCLPT 2 -
RCL 15 X<Y? RDN
SEEKPT GETX STO 13
ST- 12 GETX STO 16
RCLPT 2 - RCL 15 -
STO 17
    
```

```

156*LBL 00
RCL 17 SEEKPT GETX
FC? 01 STO 02 RCL 12
3 Y+X * XEQ 01
FC? 01 STO 04 RCL 12
* + XEQ 01 FC? 01
STO 03 RCL 12 X+2 *
+ XEQ 01 + FS? 01
RTN "Y " XEQ 19
STO 05 RCL 12 ENTER↑
ENTER↑ ENTER↑ RCL 02
3 * * RCL 03 2 * +
* RCL 04 + "Y"
XEQ 19 STO 04 X+2 E
+ 1.5 Y+X STO 10 RDN
RCL 02 6 * STO 02 *
RCL 03 2 * + "Y"
XEQ 19 STO 03 RCL 10
/ "KR" FS?C 03 GTO 19
RCL 11 RCL IND 11
RCL 06 - ENTER↑ FIX 5
RND X=0? GTO 00 RDN
DSE Y RCL IND Y /
ST- 07 SF 02 GTO 16
    
```

```

244*LBL 00
RCL 07 "X" ADV SF 12
XEQ 19 CF 12
    
```

```

251*LBL 17
ADV "FERTIG!" AVIEW
SF 25
    
```

```

256*LBL 02
CLX FIX 0 X<>F BEEP
STO 99 RTN
    
```

```

263*LBL 01
X<>Y RCL 14 + SEEKPT
X<>Y GETX RTN
    
```

```

271*LBL "KO"
E STO 17 CLX FIX 0
    
```

```

276*LBL 07
"Spline" ARCL 17 PRA
"D" XEQ 01 ST+ X +
"C" XEQ 01 - "B"
XEQ 01 ST+ X + "A"
XEQ 01 3 * - E
ST+ 17 + RCL 14 LASTX
- X<>Y ADV X=Y?
GTO 07 "ENDE" PRA RTN
    
```

309+LBL 01  
SEEKPT ARCL 17 GETX  
XEQ 19 RDN RCL 14 RTN

317+LBL "NE"  
E GTO 00

320+LBL "ST"  
18 PSIZE CLX

324+LBL 00  
X<>F FIX 0 "N=?" 2  
PROMPT X<=Y? STO 99  
STO 14 FS? 00 GTO 00  
7 \* E - "FILE?"  
AON PROMPT AOFF SF 25  
PURFL CRFLD CF 25

347+LBL 00  
RCL 14 ENTER↑ ENTER↑  
ENTER↑ STO 03 +  
STO 04 + STO 05 +  
STO 11 + E - STO 15  
+ STO 10 CLX STO 13  
FS?C 00 GTO 01 E  
RCL 14 E3 / +  
RCL 10 RCL 15

376+LBL 00  
"X" XEQ 00 "Y" SF 00

381+LBL 00  
ARCL Z SEEKPT "I="  
PROMPT SAVEX RDN X<>Y  
FC?C 00 RTN E ST+ Y  
ST+ Z RDN ISG Z  
GTO 00

397+LBL 01  
"MASS?" PROMPT STO 12  
"A?" PROMPT STO 06  
"B?" PROMPT STO 07

407+LBL 00  
RCL 11 RCL 15 ENTER↑

411+LBL 10  
RDN SEEKPT GETX GETX  
X<>Y - X<> Z SEEKPT  
E ST+ Z + X<> Z  
SAVEX X>0? GTO 00  
TONE 0 RDN RCL 15 -  
XEQ 02 GTO 00

433+LBL "XY"  
RCL 14 5 \* E -  
STO 15 ADV "INDEX?"  
PROMPT

443+LBL 02  
STO 02 E - RCL 15 +

449+LBL 10  
FC?C 01 GTO 02 E  
ST- 02 -

455+LBL 02  
RCL 14 RCL 02 X>Y?  
RTN RDN RDN SEEKPT  
"X" XEQ 02 RCL 14 +  
SEEKPT "Y" XEQ 02  
RCL 14 - E ST+ 02 +  
GTO 10

476+LBL 02  
ARCL 02 GETX SF 05

480+LBL 19  
FIX 4 RND "I=" X#0?  
X>0? "I" ARCL X  
FIX 0 AVIEW FC?C 05  
RTN GETKEY 44 X=Y?  
GTO 02 DSE X X=Y?  
ST+ 02 R↑ RTN

501+LBL 02  
R↑ SEEKPT ATOX CLA  
XTOA RDN ARCL 02  
"I=?" PROMPT SAVEX  
RDN SF 01 RTN

515+LBL 00  
RDN RCL 10 LASTX -  
X#Y? GTO 10 CLD  
RCL 14 E - RCL 15  
SEEKPT GETX STO 00  
RDN + SEEKPT GETX  
STO 09 CLA 3 RCL 14  
X>Y? GTO 20 SF 02  
XEQ 11 2 / RCL 01  
RCL 02 + STO 00 /  
RCL 04 E + SEEKPT  
RDN XEQ IND 12 SAVEX  
GTO 22

557+LBL 00  
SF 01 RTN

560+LBL 01  
ST+ X RCL 07 RCL 06 -  
GTO 01

566+LBL 04  
4 \* CHS STO 06  
STO 07 -2 / RTN

575+LBL 02  
RCL 06 RCL 01 \*  
RCL 07 RCL 02 \* + 4  
/ GTO 01

586+LBL 03  
1.5 / RCL 02 X↑2

RCL 07 \* RCL 01 X↑2  
RCL 06 \* - 6 /

600+LBL 01  
RCL 00 / - RTN

605+LBL 20  
2 - STO 00

609+LBL 11  
RCL 10 SEEKPT GETX  
GETX - LASTX GETX -  
X<>Y "RCL 11 SEEKPT  
RDN GETX STO 01 /  
X<>Y GETX STO 02 / -  
3 \* FS?C 02 RTN  
RCL 05 SEEKPT X<>Y  
SAVEX RCL 03 SEEKPT  
RCL 01 SAVEX RCL 04  
SEEKPT X<>Y RCL 02  
SAVEX + RCL 13 SEEKPT  
X<>Y ST+ X SAVEX E  
ST+ 10 XEQ 25 DSE 00  
GTO 11 2 RCL 14 -  
ST+ 10 XEQ 25  
XEQ IND 12 SAVEX  
GTO 21

666+LBL 04  
2 ST- 12 CLX STO 06  
STO 07 SF 04

673+LBL 02  
SF 00

675+LBL 03  
SF 03 XEQ 00 FS? 00  
CHS FC? 00 X↑2 2 /  
RCL 06 \* + X<>Y  
SEEKPT RDN SAVEX  
RCL 11 3 - SEEKPT  
GETX RCL 15 E -  
SEEKPT RDN GETX STO 1  
FC? 00 X↑2 2 /  
RCL 07 \* - X<>Y  
SEEKPT X<>Y FS?C 00  
RTN SF 00 GTO 02

717+LBL 00  
RCL 05 SEEKPT GETX  
RCL 11 SEEKPT RDN  
GETX STO 01 FS?C 03  
RTN GETX ENTER↑ X<> Z  
+ / \* RCL 05 SEEKPT  
RDN SAVEX RCL 11 3 -  
STO 02 SEEKPT GETX  
RCL 15 2 - SEEKPT  
RDN GETX RCL X GETX  
STO 1 + / \* RCL 02  
SEEKPT RDN

759+LBL 02  
SAVEX 3 STO 02 CLX

FS? 00 GTO 03  
 XEQ IND 02 SAVEX E  
 ST+ 02 RCL 04

771+LBL 03  
 SEEKPT GETX RCL 01  
 FC? 00 CHS + X<>Y  
 SEEKPT X<>Y SAVEX  
 RCL IND 02 3 - SEEKPT  
 GETX RCL 1 FC?C 00  
 CHS + X<>Y SEEKPT  
 X<>Y RTN

795+LBL 01  
 SF 00 XEQ 01 RCL 05  
 SEEKPT RDN GETX X<>Y  
 - RCL 05 XEQ 23 / -  
 1.5 \* CHS RCL 11 3  
 - STO 02 SEEKPT RDN  
 GETX + RCL 02 SEEKPT  
 RDN SAVEX CLX SEEKPT  
 GETX RCL 01 2 / -  
 X<>Y SEEKPT X<>Y  
 SAVEX RCL 03 3 -  
 STO 02 SEEKPT GETX  
 RCL 15 E - SEEKPT  
 RDN GETX 2 / -  
 RCL 02 SEEKPT RDN RTN

853+LBL 21  
 2 ST- 14 SIGN ST+ 03  
 . SEEKPT GETX STO 02  
 RCL 04 SEEKPT X<>Y  
 GETX X<>Y / X<>Y  
 SEEKPT RDN SAVEX  
 RCL 05 SEEKPT GETX  
 RCL 02 / . SEEKPT  
 RDN SAVEX RCL 13 E  
 - STO 00 RCL 04 E  
 ST- 03 - STO 01

890+LBL 12  
 E XEQ 25 ST+ 00  
 ST+ 01 RCL 13 SEEKPT  
 GETX RCL 03 SEEKPT  
 RDN GETX RCL 01  
 SEEKPT RDN GETX \* -  
 STO 02 RCL 05 SEEKPT  
 GETX RCL 03 SEEKPT  
 RDN GETX RCL 00  
 SEEKPT RDN GETX \* -  
 RCL 02 / RCL 13  
 SEEKPT RDN SAVEX  
 RCL 04 SEEKPT GETX  
 X<>Y SEEKPT RDN  
 RCL 02 / SAVEX RCL 13  
 RCL 14 E - X\*Y?  
 GTO 12 CHS XEQ 25 2  
 ST+ 14 RCL 14 2 -  
 STO 02 E - SEEKPT  
 RCL 05 2 - GETX X<>Y  
 SEEKPT E - X<>Y

GTO 14  
 964+LBL 13  
 RDN X<>Y SEEKPT GETX  
 X<>Z SEEKPT X<>Y  
 X<>Z GETX ST\* L  
 X<>L -

977+LBL 14  
 SAVEX STO L RDN E  
 ST- Z ST- Y DSE 02  
 GTO 13

986+LBL 22  
 XEQ IND 12 SAVEX  
 RCL 10 RCL 05 ENTER↑  
 GTO 15

993+LBL 04  
 994+LBL 02  
 RCL 06 2 / RCL 04  
 SEEKPT RDN SAVEX  
 RCL 07 2 GTO 02

1005+LBL 01  
 RCL 10 SEEKPT GETX  
 GETX X<>Y - RCL 11  
 SEEKPT RDN GETX  
 STO 01 / RCL 06 -  
 1.5 \* FS? 00 RTN  
 RCL 04 SEEKPT RDN  
 SAVEX GETX RCL 01 \*  
 2 / - RCL 01 /  
 RCL 04 SF 05

1038+LBL 23  
 SEEKPT RDN SAVEX  
 RCL 10 RCL 14 + 2 -  
 SEEKPT FS?C 00 RCL 07  
 GETX GETX X<>Y -  
 RCL 15 E - SEEKPT  
 RDN GETX FC?C 05 RTN  
 STO 02 / RCL 07 -  
 1.5 \* CHS RCL 05 2  
 - SEEKPT RDN GETX  
 RCL 02 \* 2 / -  
 RCL 02

1081+LBL 02  
 / RCL 05 E - SEEKPT  
 RDN RTN

1089+LBL 03  
 RCL 04 E + SEEKPT  
 GETX RCL 11 SEEKPT  
 RDN GETX RCL 06 \* 2  
 / - RCL 04 SEEKPT  
 RDN SAVEX RCL 15 E  
 - SEEKPT GETX RCL 07  
 \* 2 / GTO 03

1118+LBL 00

RCL 04 E + SEEKPT  
 GETX FS?C 01 GTO 00  
 STO 02 GETX - RCL 11  
 SEEKPT RDN GETX \*  
 GETX / RCL 02 +  
 RCL 04 SEEKPT RDN  
 SAVEX RCL 05 3 -  
 SEEKPT GETX GETX X<>Y  
 - RCL 15 2 - SEEKPT  
 RDN GETX / GETX \*

1159+LBL 03  
 RCL 05 2 - SEEKPT  
 RDN GETX + RTN

1168+LBL 00  
 RCL 01 \* LASTX RCL 02  
 - / SAVEX CHS RCL 02  
 \* RCL 01 / RCL 04  
 SEEKPT RDN RTN

1185+LBL 15  
 RDN X<>Y SEEKPT X<>Y  
 GETX X<>Y SEEKPT X<>Y  
 SAVEX RDN E ST+ Z +  
 RCL 11 X\*Y? GTO 15

1202+LBL 24  
 RCL 04 SEEKPT GETX  
 STO 00 ST+ X GETX  
 ST- 00 + 3 ST/ 00 /  
 RCL 11 SEEKPT RDN  
 GETX ST/ 00 STO 02 \*  
 CHS RCL 05 SEEKPT RDN  
 GETX GETX X<>Y -  
 RCL 02 / + RCL 03  
 SEEKPT RDN SAVEX  
 RCL 00 CHS RCL 13  
 SEEKPT X<>Y SAVEX E  
 XEQ 25 RCL 13 +  
 RCL 14 X\*Y? GTO 24  
 RCL 13 CHS XEQ 25  
 FC? 04 GTO 17 RCL 14  
 2 - E FIX 5  
 GTO IND 12

1260+LBL 01  
 ST+ 12 + RCL 04  
 SEEKPT + GETX X<>Y  
 SEEKPT RDN ENTER↑  
 ENTER↑ GETX ST+ Z -  
 RND X=0? GTO 17 RDN  
 GTO 00

1280+LBL 02  
 ST- 12 RDN SEEKPT  
 GETX 3 \* XEQ 01  
 STO 01 RDN XEQ 01 2  
 \* RCL 15 E - SEEKPT  
 RDN GETX \* X<>Y  
 LASTX X↑2 \* + RCL 01  
 + RCL 03 SEEKPT RDN

GETX + 2 /

1314\*LBL 00  
STO 06 STO 07 RCL 14  
GTO 20

1319\*LBL 25  
ST+ 03 ST+ 04 ST+ 05  
ST+ 11 ST+ 13 END

Dr. Hans Berghaus  
An der Joch 19  
5000 Köln 80

# Funktionswerte von Polynomen

167 Zeilen, 305 Bytes, 44 Regs., SIZE HP41C

Anlaß für die Entstehung dieses Programmes war ein Fehler (oder eine Ungenauigkeit?) im Programm „POLY“ des Programmpaketes MATH 1C von HP.

Ich erhalte zum Beispiel immer wieder für das Polynom  $P(x)=3x^5-2x^4+x^3-x-1$  für den Wert  $x=-0,65$  den Polynomwert  $P(-0,65)=-0,4432$ . Der richtige Wert muß aber  $P(-0,65)=-1,3297$  lauten. Andere Beispiele werden korrekt bestimmt.

Aufgrund dieser Unsicherheit (ist das Ergebnis nun richtig oder nicht?) habe ich mein eigenes Programm entworfen, das die Polynomwerte (nun hoffentlich richtig) bestimmt.

Das Programm wird über XEQ „YW“ (Y-Werte) aufgerufen (Flag 27 wird gesetzt) und mit „A“ für ganzrationale, bzw. mit „B“ für gebrochenrationale Funktionen gestartet. Die weitere Bedienung ist auf Grund des hohen Bedienungskomforts sehr einfach und sofort ersichtlich.

Nach Eingabe der Koeffizienten und des x-Wertes wird der Funktionswert F(x) ausgegeben. Zur Bestimmung weiterer Funktionswerte läßt man das Programm über R/S weiterlaufen.

Es können natürlich beliebig hohe Polynomgrade eingegeben werden!

## Ankündigung

Es hatte im Laufe dieses Jahres immer wieder Fragen nach der Verfügbarkeit der Programme aus BEST OF PRISMA als Barcodes gegeben. Wir arbeiten im Augenblick an der Umsetzung der Programme, die sich auf der bislang für Kopien in der Redaktion zur Verfügung stehenden Kassette befinden. Über die Art des Bezuges können wir erst dann entscheiden, wenn wir den Umfang des Papierstoßes absehen können.

Im Heft 1/88 werdet ihr dann die genauen Bezugsmöglichkeiten erfahren, habt bitte noch ein klein wenig Geduld.

Martin Meyer (1000)  
Redaktion

Ps.: Wer besitzt einen HP41 CY (HP41 CX mit 64k) oder eine 128k RSU, zu beiden werden Testberichte gewünscht, in denen positive/negative Seiten zum Vorschein kommen, bevor man sich in Unkosten stürzt, wer kauft schon gerne eine Katze im Sack?

## Hilferuf

Unser Clubmitglied André Gerards hat folgendes Problem:

Das Programm „MK“ aus BEST OF PRISMA (synthetische Tastenzuweisungen) schlägt seinem HP41CX auf den Magen, d.h. er fängt an zu „summen“. Das Summen hört erst bei MEMORY LOST wieder auf!

Sein HP41CX hat die Seriennummer 2627S20429. Sollte jemand den Grund dieses Übels und seine eventuelle Abhilfe dafür kennen, so schreibe er dies nieder und schicke dies Dokument an die PRISMA Redaktion (z.H. Martin Meyer) und, wenn es nicht zuviel Mühe macht auch noch an André, seine Adresse lautet wie folgt:

André Gerards  
Josefstraße 68  
44054 Nettetal 1

Martin Meyer (1000)  
Redaktion

Beispiel:

|              |         |          |
|--------------|---------|----------|
|              |         | XEQ "YW" |
| Y-WERTE      |         |          |
|              |         | XEQ A    |
| GANZRATIONAL |         |          |
| GRAD?        |         |          |
|              | 5       | RUN      |
| a5=?         |         |          |
|              | 3       | RUN      |
| a4=?         |         |          |
|              | -2      | RUN      |
| a3=?         |         |          |
|              | 1       | RUN      |
| a2=?         |         |          |
|              | 0       | RUN      |
| a1=?         |         |          |
|              | -1      | RUN      |
| a0=?         |         |          |
|              | -1      | RUN      |
| X=?          |         |          |
|              | -0.6500 | RUN      |
| F(X)=-1.3297 |         |          |

|                |        |       |
|----------------|--------|-------|
|                |        | XEQ B |
| GEBR. RATIONAL |        |       |
| ZAEHLERGRAD?   |        |       |
|                | 3      | RUN   |
| NENNERGRAD?    |        |       |
|                | 2      | RUN   |
| Z3=?           |        |       |
|                | 2      | RUN   |
| Z2=?           |        |       |
|                | -1     | RUN   |
| Z1=?           |        |       |
|                | 0      | RUN   |
| Z0=?           |        |       |
|                | 5      | RUN   |
| N2=?           |        |       |
|                | 1      | RUN   |
| N1=?           |        |       |
|                | 0      | RUN   |
| N0=?           |        |       |
|                | -9     | RUN   |
| X=?            |        |       |
|                | 2.0000 | RUN   |
| F(X)=-3.4000   |        |       |
|                |        | RUN   |
| X=?            |        |       |
|                | .5000  | RUN   |
| F(X)=-0.5714   |        |       |

```

01*LBL "YW"
02 " Y-WERTE"
03 AVIEW
04 SF 27
05 STOP
06*LBL 00
07 FIX 0
08 CF 29
09 CLRG
10 "GRAD?"
11 ASTO 00
12 13
13 STO 03
14 RTN
15*LBL A
16 XEQ 00
17 CF 01
18 CF 02
19 CF 03
20 CF 05
21 "GANZRATIONAL"
22 AVIEW
23 PSE
24 CLA
25 ARCL 00
26 PROMPT
27 STO 05
28 STO 06
    
```

|                    |               |                |               |               |
|--------------------|---------------|----------------|---------------|---------------|
| 29 E               | 58 STO 12     | 87 RCL 07      | 116+LBL 04    | 145 RCL IND X |
| 30 +               | 59 *2         | 88 STO 08      | 117 *F(X)=-   | 146 RT        |
| 31 STO 07          | 60 ASTO 01    | 89 *X=?        | 118 ARCL 09   | 147 X<Y       |
| 32 STO 08          | 61 *N         | 90 PROMPT      | 119 AVIEW     | 148 /         |
| 33 "a"             | 62 ASTO 02    | 91 STO 04      | 120 STOP      | 149 RTN       |
| 34 ASTO 01         | 63+LBL 01     | 92 X=0?        | 121 GTO 02    | 150+LBL 08    |
| 35 GTO 01          | 64 CLA        | 93 GTO 05      | 122+LBL 05    | 151 CF 03     |
| 36+LBL B           | 65 ARCL 01    | 94 RCL 05      | 123 RCL 05    | 152 0         |
| 37 XEQ 00          | 66 ARCL 06    | 95 STO 06      | 124 13        | 153 STO 10    |
| 38 SF 01           | 67 *I=?       | 96 13          | 125 +         | 154 RCL 11    |
| 39 SF 05           | 68 PROMPT     | 97 STO 03      | 126 RCL IND X | 155 STO 06    |
| 40 "GEBR.RATIONAL" | 69 STO IND 03 | 98+LBL 03      | 127 FS? 02    | 156 RCL 12    |
| 41 AVIEW           | 70 E          | 99 RCL 04      | 128 XEQ 07    | 157 STO 08    |
| 42 PSE             | 71 ST+ 03     | 100 RCL 06     | 129 STO 09    | 158 GTO 03    |
| 43 "ZAEHLER"       | 72 ST- 06     | 101 Y↑X        | 130 GTO 04    | 159+LBL 09    |
| 44 ARCL 00         | 73 DSE 08     | 102 RCL IND 03 | 131+LBL 06    | 160 RCL 09    |
| 45 PROMPT          | 74 GTO 01     | 103 *          | 132 RCL 02    | 161 RCL 10    |
| 46 STO 05          | 75 FS?C 01    | 104 ST+ 10     | 133 STO 01    | 162 /         |
| 47 STO 06          | 76 GTO 06     | 105 E          | 134 RCL 11    | 163 FC? 05    |
| 48 E               | 77+LBL 02     | 106 ST+ 03     | 135 STO 06    | 164 RCL 10    |
| 49 +               | 78 FS? 05     | 107 ST- 06     | 136 RCL 12    | 165 STO 09    |
| 50 STO 07          | 79 SF 02      | 108 DSE 08     | 137 STO 08    | 166 GTO 04    |
| 51 STO 08          | 80 FS? 05     | 109 GTO 03     | 138 GTO 01    | 167 END       |
| 52 "NENNER"        | 81 SF 03      | 110 FC? 03     | 139+LBL 07    |               |
| 53 ARCL 00         | 82 SF 29      | 111 GTO 09     | 140 ENTER↑    |               |
| 54 PROMPT          | 83 FIX 4      | 112 RCL 10     | 141 RDN       |               |
| 55 STO 11          | 84 0          | 113 STO 09     | 142 RDN       |               |
| 56 E               | 85 STO 09     | 114 FS?C 02    | 143 RCL 12    |               |
| 57 +               | 86 STO 10     | 115 GTO 08     | 144 +         |               |

André Gerads  
Josefstraße 68  
4054 Nettetal 1

## Tschebyscheff Tief- und Hochpaßfilter

239 Zeile, 353 Bytes, 51 Regs., SIZE 011, HP-41C

Dieses Programm entstand durch Portierung eines BASIC-Programms aus dem Buch „Nolte-Basic: HF-Rechenprogramme“ erschienen als Band 9 in der Reihe „Franziskaner Computer-Bibliothek“. Auch die gezeigten Bilder stammen aus diesem Buch. Für das freundlicherweise bereitgestellte Material möchte ich mich an dieser Stelle bedanken.

Dieses Programm berechnet die Elemente eines passiven Tief- bzw. Hochpaßfilter mit Tschebyscheffübertragungscharakteristik.

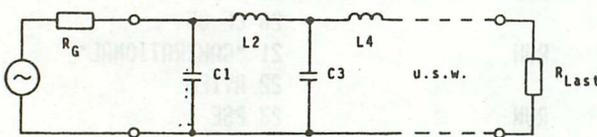


Abb. 2 Prinzipschaltbild eines passiven Tiefpaßfilters

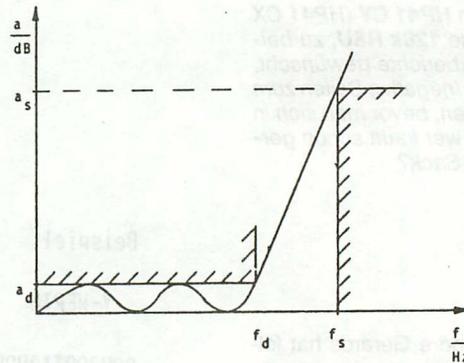


Abb. 1 Dämpfungsverlauf eines 5poligen Tschebyscheff-Tiefpaßfilters in Abhängigkeit von der Frequenz

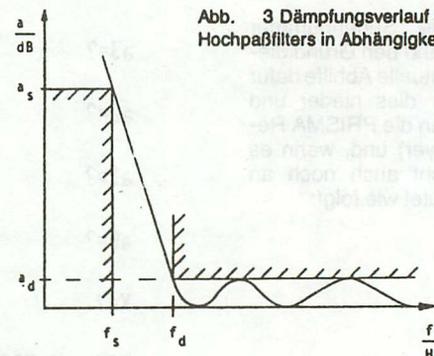


Abb. 3 Dämpfungsverlauf eines 5poligen Tschebyscheff-Hochpaßfilters in Abhängigkeit von der Frequenz

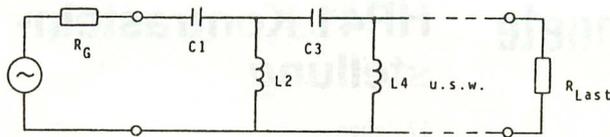


Abb. 4 Prinzipschaltbild eines passiven Hochpaßfilters

Das Programm verwendet die Flag 22 und 29 sowie die Datenregister 00 bis 10 für interne Aufgaben.

Bedienung:

<XEQ> „TTP“ für Tschebyscheff Tiefpaßfilter

bzw.  
<XEQ> „THP“ für Tschebyscheff Hochpaßfilter

Eingabe des Ein- u. Ausgangswiderstands „R0“ <R/S> [ohm]

Eingabe der Betriebsfrequenz „F0“ <R/S> [Hz]

Eingabe der Rückflußdämpfung „RD“ <R/S> [dB]

Eingabe des Typs des 1. Filterelements („L“ oder „C“)

„1. ELEMENT“ <R/S>

Eingabe der Anzahl Elemente „N“ <R/S>

Soll die Anzahl Elemente vom Programm ermittelt werden, ist ohne Eingabe <R/S> zu betätigen und anschließend sind die Sperrfrequenz „FS“ <R/S> [Hz] und die Dämpfung bei fs „A (FS)“ <R/S> [dB] einzugeben.

Anschließend werden alle Elemente ausgegeben.

Beispiel 2 :

```
XEQ "THP"
R0 =?
  75,0000 RUN
F0 =?
  170+06 RUN
RD =?
  18,0000 RUN
1. ELEMENT =?
L RUN
N =?
RUN
FS =?
  30+06 RUN
A(FS) =?
  80,0000 RUN
N=5
RUN
L(1)=65,99E-9H
RUN
C(2)=9,070E-12F
RUN
L(3)=37,08E-9H
RUN
C(4)=9,070E-12F
RUN
L(5)=65,99E-9H
RUN
```

Beispiel 1 :

```
XEQ "TTP"
R0 =?
  50,0000 RUN
F0 =?
  30+06 RUN
RD =?
  20,0000 RUN
1. ELEMENT =?
C RUN
N =?
  5,0000 RUN
C(1)=103,3E-12F
RUN
L(2)=364,0E-9H
RUN
C(3)=191,3E-12F
RUN
L(4)=364,0E-9H
RUN
C(5)=103,3E-12F
RUN

01*LBL "TTP"
02 -1
03 GTO 00
04*LBL "THP"
05 E
06*LBL 00
07 RAD
08 CF 29
09 STO 00
10 "R0"
11 XEQ 14
12 STO 01
13 "F0"
14 XEQ 14
15 ST+ X
16 PI
17 *
18 STO 02
19 "RD"
20 XEQ 14
21 -10
22 /
```

```
23 10+X
24 E
25 -
26 CHS
27 1/X
28 STO 03
29*LBL 02
30 "1. ELEMENT"
31 AON
32 XEQ 14
33 AOFF
34 ASTO X
35 "L"
36 ASTO Y
37 X*Y?
38 GTO 00
39 E
40 GTO 01
41*LBL 00
42 "C"
43 ASTO Y
44 X*Y?
45 GTO 02
46 -1
47*LBL 01
48 STO 04
49 "N"
50 CF 22
51 XEQ 14
52 FS? 22
53 GTO 00
54 "FS"
55 XEQ 14
56 ST+ X
57 PI
58 *
59 1/X
60 RCL 02
61 *
62 RCL 00
63 Y+X
64 RCL X
65 X+2
66 E
67 -
68 SQRT
69 +
70 STO 05
71 "A(FS)"
72 XEQ 14
73 10
74 /
75 10+X
76 E
77 -
78 4
79 RCL 03
80 E
81 -
82 /
83 *
84 2
85 -

86 STO 07
87 LN
88 RCL 05
89 LN
90 /
91 STO 06
92*LBL 03
93 RCL 05
94 RCL 06
95 Y+X
96 RCL X
97 RCL X
98 1/X
99 +
100 RCL 07
101 -
102 RCL Y
103 RCL X
104 1/X
105 -
106 /
107 RCL 05
108 LN
109 1/X
110 *
111 ST- 01
112 ABS
113 .01
114 X<Y?
115 GTO 03
116 RCL 06
117 2
118 /
119 INT
120 E
121 +
122 FIX 0
123 "N="
124 ARCL X
125 PROMPT
126*LBL 00
127 STO 07
128 E
129 2
130 RCL 03
131 SQRT
132 E
133 +
134 /
135 -
136 1/X
137 RCL 07
138 2
139 *
140 1/X
141 Y+X
142 RCL X
143 1/X
144 -
145 2
146 /
147 STO 06
148 PI
```

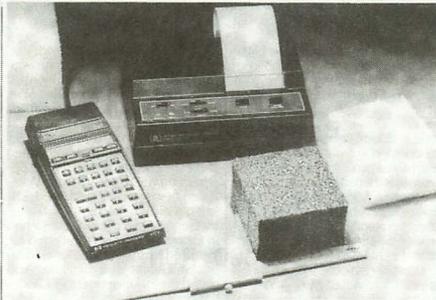
149 2  
 150 /  
 151 RCL 07  
 152 /  
 153 SIN  
 154 STO 03  
 155 ST+ X  
 156 X<>Y  
 157 /  
 158 STO 05  
 159 E  
 160 STO 08  
 161+LBL 04  
 162 RCL 01  
 163 RCL 04  
 164 Y+X  
 165 RCL 02  
 166 /  
 167 RCL 05  
 168 RCL 00  
 169 Y+X  
 170 /  
 171 RCL 04  
 172 X<0?  
 173 "C("   
 174 X>0?  
 175 "L("   
 176 FIX 0  
 177 ARCL 08  
 178 "t)="   
 179 ENG 3  
 180 ARCL Y  
 181 X<0?  
 182 "HF"  
 183 X>0?  
 184 "H"  
 185 PROMPT  
 186 -I  
 187 ST\* 04  
 188 ST- 08  
 189 RCL 07  
 190 RCL 08  
 191 X>Y?  
 192 GTO 00  
 193 RCL 03  
 194 STO 09  
 195 RCL 05  
 196 STO 10  
 197 R+  
 198 R+  
 199 2  
 200 I/X  
 201 -  
 202 X<>Y  
 203 /  
 204 PI  
 205 \*  
 206 SIN  
 207 STO 03  
 208 RCL 09

## Die elektronisch berechnete Polsterung von Dow

Das Ethacalc Programm wurde für die Verwendung mit dem normalen oder antistatischen Verpackungsmaterial Ethafoam aus Polyethylen-Schaumstoff entwickelt, das vielfach für den Transport und die Lagerung von Computern und anderen empfindlichen Geräten verwendet wird.

Das Ethacalc System benützt einen Rechner HP 41 CV von Hewlett-Packard mit Magnetkartenleser und angeschlossenen Drucker. Die Programmsoftware besteht aus Magnetkarten, die durch den Kartenleser eingelesen werden. Es bestehen besondere Datenkarten für die verschiedenen Ethafoam Typen für mehrlagigen und einmaligen Fall. Der Anwender gibt einfach die Einzelheiten des zu verpackenden Objekts wie Gewicht, Stoßempfindlichkeit und vorgeschriebene Fallhöhe ein, worauf der Drucker alle benötigten Angaben über die Gestaltung der Polsterung für die Erreichung des optimalen Schutzes bei minimalem Verpackungsvolumen ausdrückt. Mit entsprechenden Arbeitsblättern kann der Anwender seine bisherigen Kosten mit denjenigen für eine optimierte Ethafoam-Konstruktion vergleichen, wobei auch die Transportkosten in Betracht gezogen werden.

Otmar Jochum, Product Marketing Manager für Ethafoam, hat dies so formuliert: „Eine kostengünstige Verpackung bedeutet mehr als nur



Mit Ethacalc\* kann der Verpackungsdesigner die jeweils optimale stoßabsorbierende Verpackung in Sekundenschnelle berechnen.

die Verwendung des billigsten Materials, das während des Transports und bei der Lagerung einen angemessenen Schutz gewährleistet. Die Wahl des Verpackungsmaterials hat einen wesentlichen Einfluß auf das Volumen des verpackten Objekts und damit auf die Frachtkosten. Im Rahmen der möglichen Kosteneinsparungen werden alle nur denkbaren Aspekte berücksichtigt, dieser jedoch oft übersehen.

## Der HP41 langsam vorbei, wohl kaum!

Ein Beispiel für den Einsatz des HP41 in der Industrie sandte uns Werner Meschede:

In der Zeitschrift „Gut verpackt“ vom Herbst 1987 fand sich ein Werbeartikel für ein Softwarepaket für Verpackungsdesigner, um stoßabsorbierende und gleichzeitig raumsparende Verpackungen einer bestimmten Firma in Sekundenschnelle berechnen zu können.

Das Rechnersystem besteht dabei aus dem HP41 CV, dem Magnetkartenleser für das Einlesen verschiedener Verpackungsmaterialkonstanten und einem Thermodrucker für den Ausdruck der Rechnerergebnisse.

209 \*  
 210 4  
 211 \*  
 212 RCL 08  
 213 E  
 214 -  
 215 PI  
 216 \*  
 217 RCL 07  
 218 /  
 219 SIN  
 220 X+2  
 221 RCL 06  
 222 X+2  
 223 +  
 224 /  
 225 RCL 10  
 226 /  
 227 STO 05  
 228 GTO 04  
 229+LBL 14  
 230 "t =?"  
 231 PROMPT  
 232 RTN  
 233+LBL 00  
 234 CLST  
 235 CLA  
 236 SF 29  
 237 DEG  
 238 FIX 4  
 239 END

Werner Meschede  
 Sorpestraße 4  
 5788 Siedlinghausen

## HP41 Kontrasteinstellung

17 Zeilen

HP41 (neues Display), RAM-Box oder ähnliches

Dem Buch „MCOE FOR BEGINNERS“, von Ken Emery, entnahm ich die Software, die es ermöglicht, den Contrast des neuen Displays einzustellen.

Ich habe daraus eine kurze MCODE-Funktion geschrieben, die eine Contrastereinstellung von 0 – 15 erlaubt.

```
094 T
00E N
20F 0
203 C
000 NOP
130 LDI S&X
010 HEX: 010
270 RAMSLCT
3F0 PRPHSLCT
130 LDI S&X
00F HEX: 00F
0A6 A<>C S&X
1C6 A=A-C S&X
381 ?C GO
00B 02E0
168 WRIT 5(M)
3E0 RTN
```

Die Funktion erscheint im „STO.“ – Format und läßt Eingaben von 0 - 15 zu (sonst NON-EXISTENT).

Die Voreinstellung ist 5.

Marcus Otto  
 Argentinische Allee 177  
 1000 Berlin 37

## Klima-Fachingenieur/in

gesucht ab sofort oder baldmöglichst,  
 mit Berufserfahrung oder auch Jungingenieur/in.

EDV-Kenntnisse und Bereitschaft zum selbständigen  
 Arbeiten erforderlich für Planung, Beratung und  
 Überwachung von Projekten im In- und Ausland.

Bewerbung mit handschriftlichem Lebenslauf.

Ingenieurbüro R. Siegismund  
 Beratende Ingenieure VBI  
 Schlesienring 30 b, D-6368 Bad Vilbel 2

**COMPLEX 1571**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 2) CCD-Barcodes  
 Zeile 2 ( 2- 7) CCD-Barcodes  
 Zeile 3 ( 7- 12) CCD-Barcodes  
 Zeile 4 ( 12- 20) CCD-Barcodes  
 Zeile 5 ( 20- 25) CCD-Barcodes  
 Zeile 6 ( 25- 34) CCD-Barcodes  
 Zeile 7 ( 34- 41) CCD-Barcodes  
 Zeile 8 ( 41- 47) CCD-Barcodes  
 Zeile 9 ( 47- 56) CCD-Barcodes  
 Zeile 10 ( 56- 61) CCD-Barcodes  
 Zeile 11 ( 61- 68) CCD-Barcodes  
 Zeile 12 ( 68- 76) CCD-Barcodes  
 Zeile 13 ( 76- 87) CCD-Barcodes  
 Zeile 14 ( 87- 90) CCD-Barcodes  
 Zeile 15 ( 90- 93) CCD-Barcodes  
 Zeile 16 ( 93- 96) CCD-Barcodes  
 Zeile 17 ( 96- 101) CCD-Barcodes

Zeile 10 ( 34- 40) CCD-Barcodes  
 Zeile 11 ( 40- 47) CCD-Barcodes  
 Zeile 12 ( 47- 54) CCD-Barcodes  
 Zeile 13 ( 54- 63) CCD-Barcodes  
 Zeile 14 ( 63- 72) CCD-Barcodes  
 Zeile 15 ( 72- 82) CCD-Barcodes  
 Zeile 16 ( 82- 89) CCD-Barcodes  
 Zeile 17 ( 89- 93) CCD-Barcodes  
 Zeile 18 ( 93- 97) CCD-Barcodes  
 Zeile 19 ( 97- 103) CCD-Barcodes  
 Zeile 20 ( 103- 107) CCD-Barcodes  
 Zeile 21 ( 107- 111) CCD-Barcodes  
 Zeile 22 ( 111- 116) CCD-Barcodes  
 Zeile 23 ( 116- 121) CCD-Barcodes  
 Zeile 24 ( 121- 125) CCD-Barcodes  
 Zeile 25 ( 125- 129) CCD-Barcodes  
 Zeile 26 ( 129- 133) CCD-Barcodes  
 Zeile 27 ( 133- 139) CCD-Barcodes  
 Zeile 28 ( 139- 142) CCD-Barcodes  
 Zeile 29 ( 142- 147) CCD-Barcodes  
 Zeile 30 ( 147- 152) CCD-Barcodes  
 Zeile 31 ( 152- 157) CCD-Barcodes  
 Zeile 32 ( 157- 166) CCD-Barcodes  
 Zeile 33 ( 166- 173) CCD-Barcodes  
 Zeile 34 ( 173- 178) CCD-Barcodes  
 Zeile 35 ( 178- 183) CCD-Barcodes  
 Zeile 36 ( 183- 190) CCD-Barcodes  
 Zeile 37 ( 190- 199) CCD-Barcodes  
 Zeile 38 ( 199- 204) CCD-Barcodes

**TRI 1572**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 2) CCD-Barcodes  
 Zeile 2 ( 2- 5) CCD-Barcodes  
 Zeile 3 ( 5- 8) CCD-Barcodes  
 Zeile 4 ( 8- 11) CCD-Barcodes  
 Zeile 5 ( 11- 13) CCD-Barcodes  
 Zeile 6 ( 13- 20) CCD-Barcodes  
 Zeile 7 ( 20- 27) CCD-Barcodes  
 Zeile 8 ( 27- 28) CCD-Barcodes  
 Zeile 9 ( 28- 34) CCD-Barcodes

# BARCODES

Zeile 39 ( 204- 209) CCD-Barcodes



Zeile 40 ( 209- 214) CCD-Barcodes



Zeile 41 ( 214- 224) CCD-Barcodes



Zeile 42 ( 224- 230) CCD-Barcodes



Zeile 43 ( 230- 235) CCD-Barcodes



Zeile 44 ( 235- 240) CCD-Barcodes



Zeile 45 ( 240- 247) CCD-Barcodes



Zeile 46 ( 247- 256) CCD-Barcodes



Zeile 47 ( 256- 261) CCD-Barcodes



Zeile 48 ( 261- 266) CCD-Barcodes



Zeile 49 ( 266- 272) CCD-Barcodes



Zeile 50 ( 272- 281) CCD-Barcodes



Zeile 51 ( 281- 286) CCD-Barcodes



Zeile 52 ( 286- 292) CCD-Barcodes



Zeile 53 ( 292- 297) CCD-Barcodes



Zeile 54 ( 297- 304) CCD-Barcodes



Zeile 55 ( 304- 313) CCD-Barcodes



Zeile 56 ( 313- 316) CCD-Barcodes



Zeile 57 ( 316- 320) CCD-Barcodes



Zeile 58 ( 320- 326) CCD-Barcodes



Zeile 59 ( 326- 333) CCD-Barcodes



Zeile 60 ( 333- 341) CCD-Barcodes



Zeile 61 ( 341- 351) CCD-Barcodes



Zeile 62 ( 351- 354) CCD-Barcodes



Zeile 63 ( 354- 363) CCD-Barcodes



Zeile 64 ( 363- 375) CCD-Barcodes



Zeile 65 ( 375- 382) CCD-Barcodes



Zeile 66 ( 382- 385) CCD-Barcodes



Zeile 67 ( 385- 390) CCD-Barcodes



Zeile 68 ( 390- 397) CCD-Barcodes



Zeile 69 ( 397- 406) CCD-Barcodes



Zeile 70 ( 406- 411) CCD-Barcodes



Zeile 71 ( 411- 421) CCD-Barcodes



Zeile 72 ( 421- 426) CCD-Barcodes



Zeile 73 ( 426- 435) CCD-Barcodes



Zeile 74 ( 435- 448) CCD-Barcodes



Zeile 75 ( 448- 461) CCD-Barcodes



Zeile 76 ( 461- 470) CCD-Barcodes



Zeile 77 ( 470- 473) CCD-Barcodes



Zeile 78 ( 473- 478) CCD-Barcodes



Zeile 79 ( 478- 487) CCD-Barcodes



Zeile 80 ( 487- 493) CCD-Barcodes



Zeile 81 ( 493- 500) CCD-Barcodes



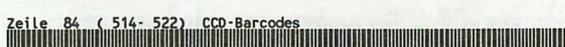
Zeile 82 ( 500- 508) CCD-Barcodes



Zeile 83 ( 508- 514) CCD-Barcodes



Zeile 84 ( 514- 522) CCD-Barcodes



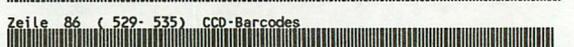
Zeile 85 ( 522- 529) CCD-Barcodes



Zeile 86 ( 529- 535) CCD-Barcodes



Zeile 87 ( 535- 544) CCD-Barcodes



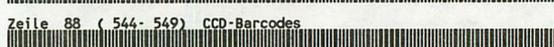
Zeile 88 ( 544- 549) CCD-Barcodes



Zeile 89 ( 549- 558) CCD-Barcodes



Zeile 90 ( 558- 564) CCD-Barcodes



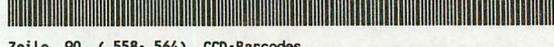
Zeile 91 ( 564- 570) CCD-Barcodes



Zeile 92 ( 570- 580) CCD-Barcodes



Zeile 93 ( 580- 583) CCD-Barcodes



Zeile 94 ( 583- 588) CCD-Barcodes



Zeile 95 ( 588- 595) CCD-Barcodes



Zeile 96 ( 595- 603) CCD-Barcodes



Zeile 97 ( 603- 608) CCD-Barcodes



Zeile 98 ( 608- 618) CCD-Barcodes



Zeile 99 ( 618- 624) CCD-Barcodes



Zeile 100 ( 624- 629) CCD-Barcodes



Zeile 101 ( 629- 634) CCD-Barcodes



Zeile 102 ( 634- 636) CCD-Barcodes



Zeile 103 ( 636- 644) CCD-Barcodes



Zeile 104 ( 644- 650) CCD-Barcodes



Zeile 105 ( 650- 662) CCD-Barcodes



Zeile 106 ( 662- 674) CCD-Barcodes



Zeile 107 ( 674- 679) CCD-Barcodes



Zeile 108 ( 679- 686) CCD-Barcodes



Zeile 109 ( 686- 692) CCD-Barcodes



Zeile 110 ( 692- 699) CCD-Barcodes



Zeile 111 ( 699- 704) CCD-Barcodes



Zeile 112 ( 704- 711) CCD-Barcodes



Zeile 113 ( 711- 719) CCD-Barcodes



Zeile 114 ( 719- 725) CCD-Barcodes



Zeile 115 ( 725- 730) CCD-Barcodes



Zeile 116 ( 730- 735) CCD-Barcodes



Zeile 117 ( 735- 742) CCD-Barcodes



Zeile 118 ( 742- 750) CCD-Barcodes



Zeile 119 ( 750- 756) CCD-Barcodes



Zeile 120 ( 756- 761) CCD-Barcodes



Zeile 121 ( 761- 766) CCD-Barcodes



Zeile 122 ( 766- 774) CCD-Barcodes



Zeile 123 ( 774- 780) CCD-Barcodes



Zeile 124 ( 780- 784) CCD-Barcodes



Zeile 125 ( 784- 795) CCD-Barcodes



Zeile 126 ( 795- 798) CCD-Barcodes



STZ 1573

Computerclub Deutschland e.V.

Zeile 1 ( 1- 2) CCD-Barcodes



Zeile 2 ( 2- 5) CCD-Barcodes



Zeile 3 ( 5- 13) CCD-Barcodes



Zeile 4 ( 13- 18) CCD-Barcodes



Zeile 5 ( 18- 24) CCD-Barcodes



Zeile 6 ( 24- 29) CCD-Barcodes



Zeile 7 ( 29- 35) CCD-Barcodes



Zeile 8 ( 35- 41) CCD-Barcodes



Zeile 9 ( 41- 48) CCD-Barcodes



Zeile 10 ( 48- 54) CCD-Barcodes



Zeile 11 ( 54- 61) CCD-Barcodes



Zeile 12 ( 61- 67) CCD-Barcodes



Zeile 13 ( 67- 76) CCD-Barcodes



Zeile 14 ( 76- 82) CCD-Barcodes



Zeile 15 ( 82- 87) CCD-Barcodes



Zeile 16 ( 87- 96) CCD-Barcodes



Zeile 17 ( 96- 102) CCD-Barcodes



Zeile 18 ( 102- 107) CCD-Barcodes



Zeile 19 ( 107- 115) CCD-Barcodes



Zeile 20 ( 115- 122) CCD-Barcodes



Zeile 21 ( 122- 129) CCD-Barcodes



Zeile 22 ( 129- 139) CCD-Barcodes



Zeile 23 ( 139- 142) CCD-Barcodes



Zeile 24 ( 142- 145) CCD-Barcodes



Zeile 25 ( 145- 149) CCD-Barcodes



Zeile 26 ( 149- 154) CCD-Barcodes



Zeile 27 ( 154- 159) CCD-Barcodes



Zeile 28 ( 159- 165) CCD-Barcodes



Zeile 29 ( 165- 173) CCD-Barcodes



Zeile 30 ( 173- 179) CCD-Barcodes



Zeile 31 ( 179- 189) CCD-Barcodes



Zeile 32 ( 189- 196) CCD-Barcodes



Zeile 33 ( 196- 201) CCD-Barcodes



Zeile 34 ( 201- 206) CCD-Barcodes



Zeile 35 ( 206- 208) CCD-Barcodes



**Hinweis:**

Wegen redaktioneller Änderungen in letzter Minute blieb für die Barcodes der Programme YW 1578, TTP 1577, FLHD 1576 und SE 1575 kein Platz. Diese werden in der nächsten PRISMA-Ausgabe abgedruckt.

## TAB 1574

Computerclub Deutschland e.V.

Zeile 1 ( 1- 5) CCD-Barcodes



Zeile 2 ( 5- 13) CCD-Barcodes



Zeile 3 ( 13- 20) CCD-Barcodes



Zeile 4 ( 20- 28) CCD-Barcodes



Zeile 5 ( 28- 36) CCD-Barcodes



Zeile 6 ( 36- 45) CCD-Barcodes



Zeile 7 ( 45- 55) CCD-Barcodes



Zeile 8 ( 55- 61) CCD-Barcodes



Zeile 9 ( 61- 68) CCD-Barcodes



Zeile 10 ( 68- 76) CCD-Barcodes



Zeile 11 ( 76- 80) CCD-Barcodes



Zeile 12 ( 80- 88) CCD-Barcodes



Zeile 13 ( 88- 96) CCD-Barcodes



Zeile 14 ( 96- 101) CCD-Barcodes



Zeile 15 ( 101- 106) CCD-Barcodes



Zeile 16 ( 106- 114) CCD-Barcodes



Zeile 17 ( 114- 122) CCD-Barcodes



Zeile 18 ( 122- 129) CCD-Barcodes



Zeile 19 ( 129- 136) CCD-Barcodes



Zeile 20 ( 136- 140) CCD-Barcodes



Zeile 21 ( 140- 143) CCD-Barcodes



Zeile 22 ( 143- 150) CCD-Barcodes



Zeile 23 ( 150- 155) CCD-Barcodes



Zeile 24 ( 155- 156) CCD-Barcodes



## SPL 1579

Computerclub Deutschland e.V.

Zeile 1 ( 1- 2) CCD-Barcodes



Zeile 2 ( 2- 3) CCD-Barcodes



Zeile 3 ( 3- 5) CCD-Barcodes



Zeile 4 ( 5- 7) CCD-Barcodes



Zeile 5 ( 7- 8) CCD-Barcodes



Zeile 6 ( 8- 11) CCD-Barcodes



Zeile 7 ( 11- 13) CCD-Barcodes



Zeile 8 ( 13- 14) CCD-Barcodes



Zeile 9 ( 14- 16) CCD-Barcodes



Zeile 10 ( 16- 18) CCD-Barcodes



Zeile 11 ( 18- 20) CCD-Barcodes



Zeile 12 ( 20- 23) CCD-Barcodes



Zeile 13 ( 23- 34) CCD-Barcodes



Zeile 14 ( 34- 45) CCD-Barcodes



Zeile 15 ( 45- 56) CCD-Barcodes



Zeile 16 ( 56- 67) CCD-Barcodes



Zeile 17 ( 67- 71) CCD-Barcodes



Zeile 18 ( 71- 76) CCD-Barcodes



Zeile 19 ( 76- 79) CCD-Barcodes



Zeile 20 ( 79- 87) CCD-Barcodes



Zeile 21 ( 87- 96) CCD-Barcodes



Zeile 22 ( 96- 97) CCD-Barcodes



Zeile 23 ( 97- 103) CCD-Barcodes



Zeile 24 ( 103- 109) CCD-Barcodes



Zeile 25 ( 109- 113) CCD-Barcodes



Zeile 26 ( 113- 121) CCD-Barcodes



Zeile 27 ( 121- 132) CCD-Barcodes



Zeile 28 ( 132- 140) CCD-Barcodes



Zeile 29 ( 140- 149) CCD-Barcodes



Zeile 30 ( 149- 158) CCD-Barcodes



Zeile 31 ( 158- 166) CCD-Barcodes



Zeile 32 ( 166- 175) CCD-Barcodes



Zeile 33 ( 175- 183) CCD-Barcodes



Zeile 34 ( 183- 194) CCD-Barcodes



Zeile 35 ( 194- 203) CCD-Barcodes



Zeile 36 ( 203- 214) CCD-Barcodes



Zeile 37 ( 214- 223) CCD-Barcodes



Zeile 38 ( 223- 230) CCD-Barcodes



Zeile 39 ( 230- 239) CCD-Barcodes



Zeile 40 ( 239- 247) CCD-Barcodes



Zeile 41 ( 247- 253) CCD-Barcodes



Zeile 42 ( 253- 259) CCD-Barcodes



Zeile 43 ( 259- 269) CCD-Barcodes



Zeile 44 ( 269- 275) CCD-Barcodes



Zeile 45 ( 275- 279) CCD-Barcodes



Zeile 46 ( 279- 285) CCD-Barcodes



Zeile 47 ( 285- 292) CCD-Barcodes



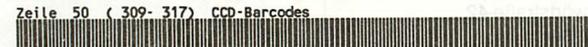
Zeile 48 ( 292- 302) CCD-Barcodes



Zeile 49 ( 302- 309) CCD-Barcodes



Zeile 50 ( 309- 317) CCD-Barcodes



Zeile 51 ( 317- 320) CCD-Barcodes



Zeile 52 ( 320- 327) CCD-Barcodes



Zeile 53 ( 327- 335) CCD-Barcodes



Zeile 54 ( 335- 343) CCD-Barcodes



Zeile 55 ( 343- 352) CCD-Barcodes



Zeile 56 ( 352- 365) CCD-Barcodes



Zeile 57 ( 365- 375) CCD-Barcodes



Zeile 58 ( 375- 382) CCD-Barcodes



Zeile 59 ( 382- 390) CCD-Barcodes



Zeile 60 ( 390- 398) CCD-Barcodes



Zeile 61 ( 398- 404) CCD-Barcodes



Zeile 62 ( 404- 414) CCD-Barcodes



Zeile 63 ( 414- 422) CCD-Barcodes



Zeile 64 ( 422- 431) CCD-Barcodes



Zeile 65 ( 431- 436) CCD-Barcodes



Zeile 66 ( 436- 443) CCD-Barcodes



Zeile 67 ( 443- 453) CCD-Barcodes



Zeile 68 ( 453- 464) CCD-Barcodes



Zeile 69 ( 464- 471) CCD-Barcodes



Zeile 70 ( 471- 479) CCD-Barcodes



Zeile 71 ( 479- 486) CCD-Barcodes



Zeile 72 ( 486- 494) CCD-Barcodes



Zeile 73 ( 494- 503) CCD-Barcodes



Zeile 74 ( 503- 510) CCD-Barcodes



Zeile 75 ( 510- 521) CCD-Barcodes



Zeile 76 ( 521- 530) CCD-Barcodes



Zeile 77 ( 530- 539) CCD-Barcodes



Zeile 78 ( 539- 549) CCD-Barcodes



Zeile 79 ( 549- 557) CCD-Barcodes



Zeile 80 ( 557- 567) CCD-Barcodes



# BARCODES

Zeile 81 ( 567- 579) CCD-Barcodes



Zeile 82 ( 579- 589) CCD-Barcodes



Zeile 83 ( 589- 602) CCD-Barcodes



Zeile 84 ( 602- 612) CCD-Barcodes



Zeile 85 ( 612- 622) CCD-Barcodes



Zeile 86 ( 622- 632) CCD-Barcodes



Zeile 87 ( 632- 641) CCD-Barcodes



Zeile 88 ( 641- 651) CCD-Barcodes



Zeile 89 ( 651- 657) CCD-Barcodes



Zeile 90 ( 657- 665) CCD-Barcodes



Zeile 91 ( 665- 674) CCD-Barcodes



Zeile 92 ( 674- 681) CCD-Barcodes



Zeile 93 ( 681- 692) CCD-Barcodes



Zeile 94 ( 692- 701) CCD-Barcodes



Zeile 95 ( 701- 711) CCD-Barcodes



Zeile 96 ( 711- 720) CCD-Barcodes



Zeile 97 ( 720- 728) CCD-Barcodes



Zeile 98 ( 728- 738) CCD-Barcodes



Zeile 99 ( 738- 748) CCD-Barcodes



Zeile 100 ( 748- 757) CCD-Barcodes



Zeile 101 ( 757- 766) CCD-Barcodes



Zeile 102 ( 766- 774) CCD-Barcodes



Zeile 103 ( 774- 783) CCD-Barcodes



Zeile 104 ( 783- 792) CCD-Barcodes



## Clubadressen:

### 1. Vorsitzender

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

### 2. Vorsitzender

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

### Schatzmeister Mitgliederverwaltung

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

### 1. Beisitzer CP/M-80 Service

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

Klaus Tannenberger (3004)  
Ober-Ramstädter-Straße 5  
6107 Reinheim 5  
☎ 06162 / 84797

### Programm-Bibliothek HP-71

Henry Schimmer (786)  
Homburger Landstraße 63  
6000 Frankfurt 50

### Beirat Programm-Bibliothek HP-75

Michael Hartmann (380)  
Wendelsgrund 13  
6690 St. Wendel

### 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  
Ursulakloster 4  
5000 Köln 1  
☎ 0221 / 131297

### Grabau GR7 Interface

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

### 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)  
Schüslerweg 18 b  
2100 Hamburg 90  
☎ 040 / 7905672

### Beirat Regionalgruppe Karlsruhe

Stefan Schwall (1695)  
Rapenwörtstraße 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

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

### Beirat

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

### Beirat

Günther Schwarz (2658)  
Bodelschwingstraße 34  
3408 Duderstadt 1

Zeile 105 ( 792- 800) CCD-Barcodes



Zeile 106 ( 800- 808) CCD-Barcodes



Zeile 107 ( 808- 819) CCD-Barcodes



Zeile 108 ( 819- 828) CCD-Barcodes



Zeile 109 ( 828- 838) CCD-Barcodes



Zeile 110 ( 838- 848) CCD-Barcodes



Zeile 111 ( 848- 857) CCD-Barcodes



Zeile 112 ( 857- 866) CCD-Barcodes



Zeile 113 ( 866- 875) CCD-Barcodes



Zeile 114 ( 875- 886) CCD-Barcodes



Zeile 115 ( 886- 894) CCD-Barcodes



Zeile 116 ( 894- 903) CCD-Barcodes



Zeile 117 ( 903- 912) CCD-Barcodes



Zeile 118 ( 912- 921) CCD-Barcodes



Zeile 119 ( 921- 930) CCD-Barcodes



Zeile 120 ( 930- 941) CCD-Barcodes



Zeile 121 ( 941- 950) CCD-Barcodes



Zeile 122 ( 950- 960) CCD-Barcodes



Zeile 123 ( 960- 969) CCD-Barcodes



Zeile 124 ( 969- 977) CCD-Barcodes



Zeile 125 ( 977- 985) CCD-Barcodes



Zeile 126 ( 985- 992) CCD-Barcodes



Zeile 127 ( 992-1003) CCD-Barcodes



Zeile 128 (1003-1012) CCD-Barcodes



Zeile 129 (1012-1021) CCD-Barcodes



Zeile 130 (1021-1030) CCD-Barcodes



Zeile 131 (1030-1040) CCD-Barcodes



Zeile 132 (1040-1050) CCD-Barcodes



Zeile 133 (1050-1059) CCD-Barcodes



Zeile 134 (1059-1069) CCD-Barcodes



Zeile 135 (1069-1080) CCD-Barcodes



Zeile 136 (1080-1092) CCD-Barcodes



Zeile 137 (1092-1101) CCD-Barcodes



Zeile 138 (1101-1111) CCD-Barcodes



Zeile 139 (1111-1122) CCD-Barcodes



Zeile 140 (1122-1130) CCD-Barcodes



Zeile 141 (1130-1139) CCD-Barcodes



Zeile 142 (1139-1148) CCD-Barcodes



Zeile 143 (1148-1158) CCD-Barcodes



Zeile 144 (1158-1169) CCD-Barcodes



Zeile 145 (1169-1181) CCD-Barcodes



Zeile 146 (1181-1190) CCD-Barcodes



Zeile 147 (1190-1200) CCD-Barcodes



Zeile 148 (1200-1207) CCD-Barcodes



Zeile 149 (1207-1216) CCD-Barcodes



Zeile 150 (1216-1225) CCD-Barcodes



Zeile 151 (1225-1235) CCD-Barcodes



Zeile 152 (1235-1244) CCD-Barcodes



Zeile 153 (1244-1252) CCD-Barcodes



Zeile 154 (1252-1261) CCD-Barcodes



Zeile 155 (1261-1270) CCD-Barcodes



Zeile 156 (1270-1279) CCD-Barcodes



Zeile 157 (1279-1287) CCD-Barcodes



Zeile 158 (1287-1296) CCD-Barcodes



Zeile 159 (1296-1308) CCD-Barcodes



Zeile 160 (1308-1318) CCD-Barcodes



Zeile 161 (1318-1324) CCD-Barcodes

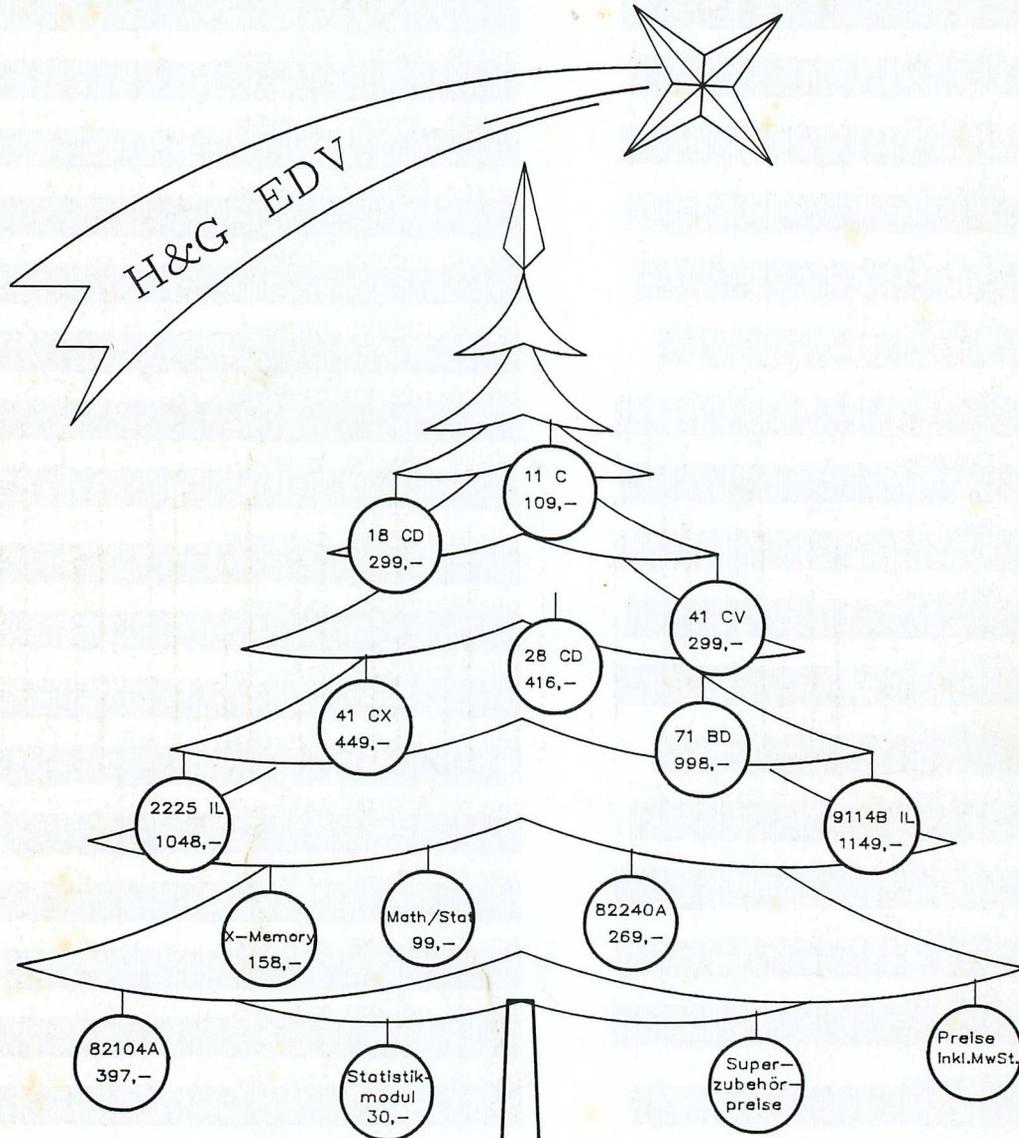


Zeile 162 (1324-1325) CCD-Barcodes



\*\*\*\*\*

# Der gute Stern am Computerhimmel



|                                                                                  |                                                                                                   |                                                                   |                                             |                                    |                               |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|---------------------------------------------|------------------------------------|-------------------------------|
| HP Laser Jet Serie II<br>aus unserer Ausstellung<br>5790,-<br>12 Monate Garantie | ATARI 1040<br>SM 124 Monitor<br>P6 Drucker Druckerkab.<br>2589,-<br>Commodore AMIGA 500<br>1048,- | <b>H&amp;G EDV</b><br>Münsterstr. 1<br>5300 Bonn 1<br>0228/729080 | Commodore AT PC 40/40<br>Komplett<br>4999,- | Epson LX 800, Druckerkab.<br>570,- | NEC P2200<br>24 Nad.<br>968,- |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|---------------------------------------------|------------------------------------|-------------------------------|

Preise für CCD Mitglieder unter Angabe der Mitgliedsnummer.....