

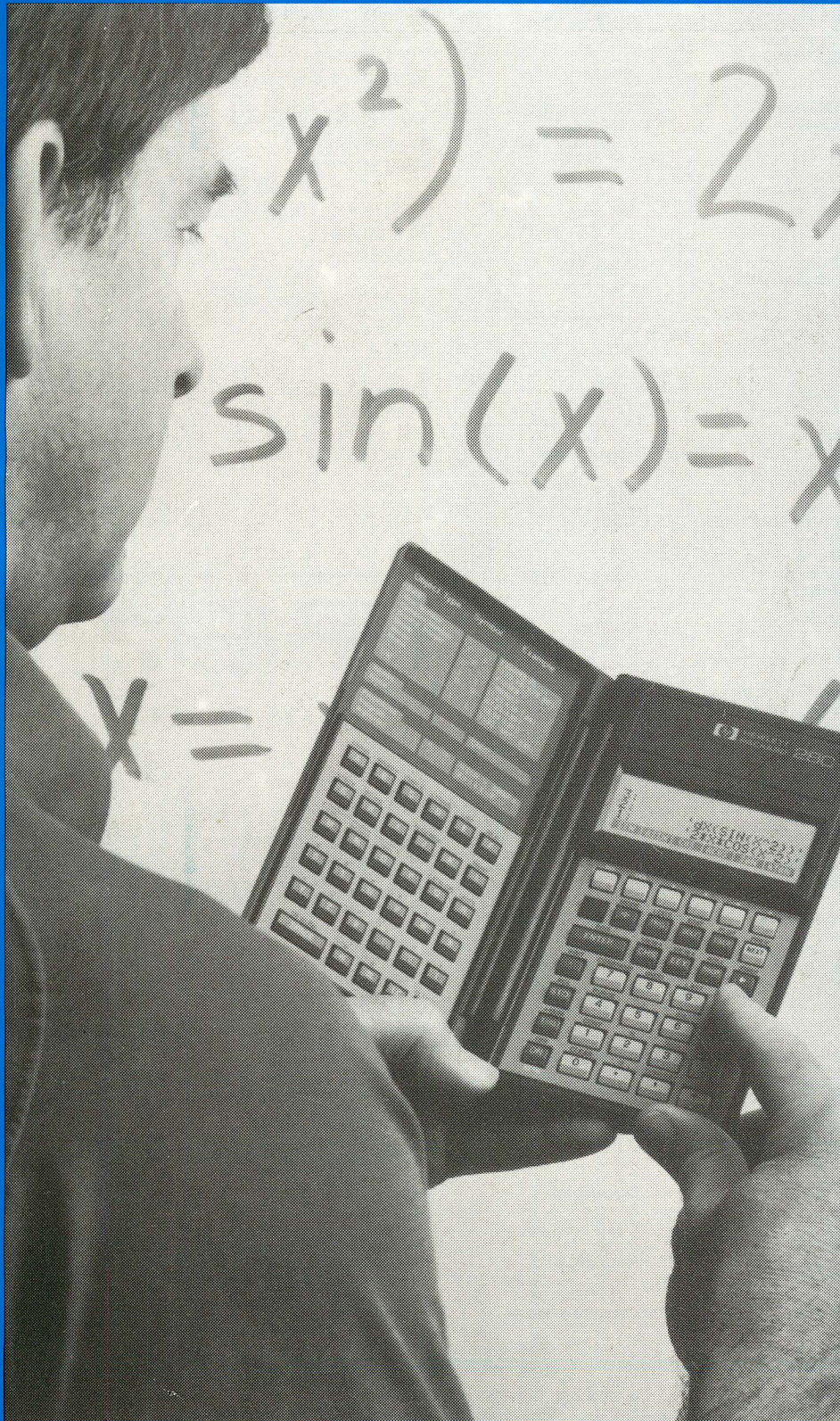
PRISMA

Computerclub Deutschland e.V.-Postfach 2129-6242 Kronberg 2

Prisma Herstellung-Limburger Str. 15-6242 Kronberg 2

Januar/Februar 1987 Nr.1

D 2856 E



Wieder ein neuer Taschenrechner! Nach dem Business HP-18C jetzt der naturwissenschaftliche Bruder. Der erste Taschenrechner, der algebraische Probleme löst indem er Symbole oder Variablen benutzt. Ein Rechner, der von der Leistung einem HP-41 mit CCD-ROM und AEC-ROM entspricht. (Ohne allerdings ein System aus vielen Komponenten zu sein). Auf dem 4-zeiligen Display kann man Funktionen plotten. Sogar eine Zoomfunktion mit einem kleinen Graphikcursor ist vorhanden. Ein Traum von einem Rechner!

Clubnachrichten

Mitgliederversammlung 1987
Welche Computer im CCD?
Programmsammlung Vermessung
Buchbesprechung

Serie 80

Assembler-Kurs

Serie 10

HP-15C Speed-Up

Serie 70

Sortierprogramme
Umlaute auf dem ThinkJet
Matrix
Lineare Gleichungssysteme
Grafik

Serie 40

64 KB RAM STORAGE UNIT
Polynomdivision
Gedächtnistest
Zeilenblockverwaltung
Elektrotechnik
AEC-ROM Testbericht
Memory String
Einkommensteuer 1986/88
Spline
Grafik auf dem ThinkJet

Clubnachrichten

Mitgliederversammlung

AM 4. April 1987
Im Volkshaus Enkheim (Frankfurt)
Borsigallee Ecke Vilbeler Landstr.
um 10.00 Uhr.

Tagesordnung

1. Begrüßung durch den Vorstand
2. Feststellung der Beschlußfähigkeit und andere Formalitäten
3. Bericht des Vorstandes
4. Bericht der Kassenprüfer
5. Entlastung des Vorstandes
6. Neuwahl des Vorstandes
7. Neuwahl der Kassenprüfer
8. Beirat
9. Haushaltsplan
10. Prisma
11. Verschiedenes

Hallo !!!!

In diesem Jahr wird der Vorstand des CCD neu gewählt. Überlegt Euch bitte, wer aus unseren Reihen Verantwortung übernehmen will. Es werden sicherlich einige Posten neu zu besetzen sein. Die Entwicklung des CCD braucht neue Leute. Nach 7 Jahren Geschichte braucht der Club frisches Blut. Neue Ideen und Power ist gefragt. Für mich kann ich sagen, das ich nur noch Half Power habe. (Übrigens einige große Firmen auch). Legt doch mal den Stapel Prisma aufeinander, vielleicht könnt Ihr Euch dann vorstellen, daß ich nicht mehr viel Kraft habe, meine Aufgaben im Club neben Familie und Beruf noch wahrzunehmen. Ich hoffe, daß Nachfolger gefunden werden, die den Club einen Schritt weiterbringen.

Rolf Hansmann

.END.

An alle Videointerfaceanwender

Ich, Vermessungsingenieur, setze im Berufsalltag mehrere Videointerfaces in Verbindung mit dem HP 71 ein, z.B. HP 82163, PacScreen V 1.1A, sowie Mountain (HP 92198) und wäre erfreut über einige Anregungen zur Gestaltung Bildschirmmasken. Sollte jemand schon Erfahrungen auf diesem Gebiet gesammelt haben, würde ich mich für Überlassung brauchbarer Anregungen gern mit einer Kassette oder Diskette bedanken.

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

.END.

Wichtige Clubadressen:

1. Vorsitzender PRISMA-Nachsendedienst

Peter Kiefer (958)
 Roßmarktgasse 11
 7798 Pfullendorf
 ☎ 07552 / 7234

2. Vorsitzender Fotosatz "PRISMA"

Rolf Hansmann (500)
 Limburger Straße 15
 6242 Kronberg II

Schatzmeister Mitgliederverwaltung

Dieter Wolf (1734)
 Pützerstraße 29
 6000 Frankfurt 90
 ☎ 069 / 765912

1. Beisitzer HP-41 „Synthetik“

Andreas Markscheffel (69)
 Nachtigallenweg 8
 6246 Glashütten

2. Beisitzer

Wilfried Kötz (1637)
 Im Aehelemaar 20
 5060 Bergisch Gladbach

Programm-Bibliothek HP-41

Klaus Tannenberger (3004)
 Ober-Ramstädter-Str. 5
 6107 Reinheim 5
 ☎ 06162 / 84797

Programm-Bibliothek HP-71

Henry Schimmer (786)
 Homburger Landstr. 63
 6000 Frankfurt 50

Serie 70 Service Programm-Bibliothek HP-75

Michael Hartmann (380)
 Wendelsgrund 13
 6690 St. Wendel

Serie 80 Service

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

CP/M-80 Service

Werner Dworak (607)
 Allewind 51
 7900 Ulm
 ☎ 07304 / 3274

MS-DOS Service

Alex Wolf (1734 jr.)
 Pützerstraße 29
 6000 Frankfurt 90
 ☎ 069 / 765912

"Hardware 41"

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

"Grabau GR7 Interface"

Holger von Stillfried
 Alsterkrugchausee 212
 2000 Hamburg 60
 ☎ 040 / 5116346

"E-Technik"

Werner Meschede (2670)
 Sorpestraße 4
 5788 Siedlingshausen

"Geowissenschaften"

Alf-Norman Tietze (1909)
 Thudichumstr. 14
 6000 Frankfurt am Main 90
 ☎ 069 / 7893995

"Mathematik"

Andreas Wolpers (349)
 Steinstraße 15
 7500 Karlsruhe

"Vermessungswesen"

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

Ortsgruppe Berlin

Jörg Warmuth (79)
 Wartburgstraße 17
 1000 Berlin 62

Ortsgruppe Cuxhaven

Dennis Föh (2374)
 Drangsweg 1
 2190 Cuxhaven
 ☎ 04721 / 24129

Ortsgruppe Hamburg

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

Ortsgruppe Hannover

Horst Ziegler (1361)
 Schüslerweg 18 b
 2100 Hannover 90
 ☎ 040 / 7905672

Ortsgruppe Karlsruhe

Rainer Maier (1708)
 Tiergartenstraße 56
 3000 Hannover 71
 ☎ 0511 / 517702

Ortsgruppe Köln

Stefan Schwall (1695)
 Rappenwörstraße 42
 7500 Karlsruhe 21
 ☎ 0721 / 576756

Ortsgruppe München

Frank Ortman (1089)
 Okerstraße 24
 5090 Leverkusen 1

Ortsgruppe Rhein-Main

Andreas Eschmann (2289)
 Lahnstraße 2
 6096 Raunheim
 ☎ 06142 / 46642

CCD-Beirat

Werner Dworak (607)
 Allewind 51
 7900 Ulm
 ☎ 07304 / 3274 priv., 0731 / 392-2689 dienstl.

Prof. Dr. Wolfgang Fritz (125)
 Kronenstraße 34
 7500 Karlsruhe

Dr. Max Hahn (983)
 Humboldtstraße 10
 4044 Kaarst 1
 ☎ 02101 / 64580 priv., 0211 / 3112533 dienstl.

Michael Hartmann (380)
 Wendelsgrund 13
 6690 St. Wendel

Erich H. Klee (1170)
 Ruhrallee 8
 4300 Essen 1

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

Eduard Wulff (556)
 Schlüterstr. 62
 2000 Hamburg 13
 ☎ 040/459765

PRISMA

Impressum

Titel:
PRISMA
Herausgeber:
CCD - Computerclub Deutschland e.V.
Sitz:
Limburger Str. 15, 6242 Kronberg II
Verantwortlicher Redakteur:
Rolf Hansmann

Redakteure:
Hans Jürgen Hübner
Martin Meyer
Henry Schimmer
Alf-Norman Tietze
Herstellung:
Rolf Hansmann
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.

Anzeigenpreise:
Es gilt unsere Anzeigenpreisliste 2 vom Juni 1982.

Erscheinungsweise:
PRISMA erscheint 9 bis 10 mal im Jahr.

Auflage:
3000

Bezug:
PRISMA wird 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.

Anschrift:
Postfach 2129, 6242 Kronberg 2.

Termine der Ortsgruppe Köln

für das Jahr 1987

An alle Clubmitglieder und Interessenten im Raum Köln

Alle Termine jeweils am zweiten Sonntag im Monat um 15.00 Uhr

- 11. Januar
- 8. Februar
- 8. März
- 12. April
- 10. Mai
- 14. Juni
- 12. Juli
- Sommerferien -
- 11. Oktober
- 8. November
- 13. Dezember

im Ev. Jugendhaus Leverkusen-Rheindorf (neben der Paul-Gerhardt-Kirche) Elbestraße 6 (links neben Eingang - Nr. 4), 5090 Leverkusen 1 (Rheindorf-Nord)

Ich würde mich sehr freuen, wieder ein paar alte Gesichter, aus der Zeit als die Ortsgruppe sich noch in Köln traf, zu sehen.

.END.

Welche Computer im CCD

Wir wissen alle, wie es angefangen hat: einst war der HP-41C der Computer des "kleinen Mannes". Man konnte sich ihn noch am ehesten leisten, da zu der Zeit ernstzunehmende Home- oder Personalcomputer noch ziemlich unerschwinglich waren. Auch war der HP-41C ein geeigneter Computer für Einsteiger, da die ersten Programmierversuche durch die direkte Tastenprogrammierung schnell zum Erfolg führten.

Heute ist nun vieles anders geworden. HC's und PC's sind inzwischen so billig und leistungsfähig geworden, daß selbst ein HP-41CX mit Speedup, oder auch ein HP-71/75 nicht mehr mitkommen. Auch die Dinge, die man an den kleinen HP-Rechnern so schätzte, Constant Memory, UPN, Calc-Modus, ein superstarkes, erweiterbares Betriebssystem (HP-71B), eingebaute Hardwareuhr, etc., sind inzwischen alle auch für die großen Computer erhältlich, wenn nicht als Standard, dann als Zubehör. Zu dem besserem Preis kommt dann bei den Desktopcomputern auch noch der Vorteil der schönen Bildschirmdarstellung, Möglichkeit der Tonausgabe, meist Zugriff auf professionelle Software, manchmal auch eine etwas leichter zu beherrschende Assembler/Maschinensprache (z.B. 6502 CPU) als bei HP, etc.

Wozu also überhaupt noch HHC's (Handheld Computers)? Wer kauft denn diese noch, und welcher Club sollte sie unterstützen? Die Antwort liegt nach meiner Meinung nur in der Portabilität. Jedem, der beruflich oder privat viel reist, der viel außer Haus zu tun hat, oder der an seinem Arbeitsplatz in der Firma keinen Platz oder keine Möglichkeit hat, einen Desktopcomputer aufzustellen, ist mit einem HHC nach wie vor bestens gedient. Ich glaube daher, PRISMA sollte eine Zeitschrift für

mobile Computerbenutzer werden und damit den Rechner HP-41/71/75 treu bleiben.

Aus den obengenannten Vorteilen der Desktopcomputer ergibt sich aber die Möglichkeit, daß viele CCD Mitglieder früher oder später sich ein zweites Computersystem für zu Hause zulegen werden. Auch ich gehöre übrigens schon seit langem zu dieser Gruppe, da ich beruflich als Pilot sehr viel und sehr lange weg bin von zu Hause, den kleinen HP-71B im Cockpit optimal nutzen kann, und ich aber zu Hause auf den Komfort einer "richtigen" Tastatur und eines großen Bildschirms nicht verzichten muß.

Für uns "Doppelbenutzer" stellt sich allerdings meist sehr bald die Frage nach einer Verbindung der beiden Computer, um bei Bedarf Daten oder auch Programme austauschen zu können. Auch kann eine solche Verbindung dazu führen, daß man sich einige Peripheriegeräte ersparen kann, die man sonst doppelt hätte (z.B. Drucker, Bildschirm, Diskdrive, Modem, etc.). Hier sehe ich ein weites Betätigungsfeld für den CCD. Von den Computerherstellern werden ja selten funktionierende Fertiglösungen für derartige Interfaces angeboten, daher müssen hier die Bastler und Tüftler ran, von denen es im CCD ja eine Menge geben soll.

Zusammenfassend möchte ich also gerne, daß der CCD im Prinzip bei "seinem Leisten" bleibt und die HHC und HP ausführlich behandelt, evtl. mit Schwerpunkt auf mobile Anwendungen. Zusätzlich dazu sollte Desktopcomputer behandelt werden, insoweit sie dazu dienen können, mit unseren HHC's zu kommunizieren. So ist bei mir z.B. die Software für ein HP-71B zu Commodore C-128 (+64) Interface erhältlich (über HP-IL und GPIO-Interface), und ich weiß von einem Kollegen, der Datenaustausch zwischen HP-71B und einem IBM-PC kompatibelem betreibt.

Alle weiteren Clubaktivitäten (z.B. Osborne-, PC-, ATARI ST-Gruppen) sollten, bei genügend starkem Interesse, ihre eigene Untergruppen im CCD aufmachen mit ihrem eigenen Journal (z.B. auf Diskette) und ihrem eigenen Unkostenbeitrag, um die PRISMA Redaktion weder finanziell noch personell zu belasten. Eventuell allgemein interessierende Artikel könnten dann vielleicht auch in PRISMA erscheinen. Die Mitgliedschaft in einer dieser Untergruppen sollte nicht unbedingt an die CCD-Mitgliedschaft gebunden sein, der Bezug von PRISMA entfällt dann.

Wie wär's mit einer Fragebogenaktion unter den Mitgliedern, in der nach den verwendeten Computern gefragt wird, den häufigsten Anwendungen, verwendeten Programmen, zu Hause, unterwegs, am Arbeitsplatz, etc.? Aus Fragen nach Interessen, Projekten, Wünschen könnte man sicher die Frage nach dem "wohin?" des CCD ganz gut ableiten.

Wohin er auch gehen mag, ich wünsche dem CCD weiterhin langjähriges Bestehen und bedanke mich ganz herzlich bei Rolf Hansmann und der PRISMA-Redaktion für die viele Mühe und die ausgezeichnete Arbeit, die sie bisher geleistet haben!

Michael Fiedler
Friedrichstr. 17
6070 Langen

.END.

Clubbörse

Suchen Kontakt zu anderen c't 68000-Usern (Hard-/Software)

Henry Schimmer (PEARL) ☎069/553520
Boris Holpisch (OSG) ☎069/871170

Verkaufe CCD-Modul für 200,-DM.
N. Thomas (2675), ☎02364/15889, abends.

Verkaufe HP-71 für 900,-DM, HP-71 Mathe-Module für 200,-DM, HP-IL/RS-232 Interface 700,-DM incl. Development Utility.
Peter Ehrenberg (1297), Bramfelder Str. 10, 2000 Hamburg

Verkaufe CP/M-80 Turbopascal zum HP 86 fpr F. 500.- und 51/2 Zoll Doppel-Floppy 82901M mit HP IB Anschluß für Fr. 1'000.-.
Philipp Baumann, Kirchmattstr. 14, 3422 Kirchberg, Schweiz

Suche folgende Prismahefte, ggf. auch brauchbare Kopien von 2,9/83; 1,3,4/85; 9/84.
Bruno Gfeller (3255), Brunnmattstr. 48, CH-3008 Bern.

HP 75 C zu verkaufen. Für HP 41: IL-Modul 82160A für 190,-DM.
B. Podbielski, ☎0571 33827.

Zu verkaufen: Detaillierte Anleitung zum Bau von Doppelmodulen (6 Seiten DIN A4) für 5,-DM. 15 Rollen schwarz- und 6 Rollen bladruckendes Thermopapier für 40,-DM plus Portokosten per Nachnahme.
Stefan Fegert (1937), Marxstr. 35, 4320 Hattingen.

HP-71: Verkaufe 32 K Speichererweiterung (VHB 450.-DM) und 64 K Eprom (Modul mit beliebiger Softwar; VHB 500.-DM). Beides wird wie ein Modul in einen der Ports eingesteckt.
Jochen Haas, Leverkusen, ☎02171 46802 oder 48926

Verkaufe: Sprachinterface f.d. HP41 71 75)PRISMA 84.9.31, fertig aufgebaut im Geh.) für 160.-DM, GAMES STANDARD Modul (1 Gehäuse) für 90.-DM, Akkuladegerät HP 82066B für 25.-DM. Bücher: The Basic HP-71 by R. Harvey für 35.-DM, IDS Band 1 + 2 für 35.-DM, Dienstspg. für den HP-41 für 10.-DM HP-41 in der Praxis für 20.-DM, Anwenderhandbuch HP-41 für 20.-DM, alle Preise VHB.
☎07131 484738

Verkaufe: HP-IL Thermodrucker 82162A. Accu, Netzteil, HP-IL Modul und Systemkoffer incl. Zubehör für 800.-DM.
Andreas Eschmann (CCD 2289), Lahnstr. 2, 6096 Raunheim. ☎06142 46642.

HP 71B, ...?-Interface, Cardreader, Translatorpac, Finanzmodul zu verkaufen, komplett für 1.900.-DM.
Juergen Gentz., Hudtwalckerstr. 12, 4578 Hamburg 60, ☎460 4578.

Verkaufe preisgünstig: HP 41C, div. Module (Quad-Ram u.a.), IL-Drucker, opt. Lesestift, HP-Drucker 82143 A BROTHER EP 44 + Zubeh., SHARP PC 1402 + CE 126 P Info gegen Freiumschlag oder ☎07464.882 (ab 18.00 Uhr)
Bernd Haug, Am Kirchberg 8, 7201 Seitingen 1

64k CP/M-Computer (Terminalaufbau, max. 640x400 Pixel, 8 Farben, ROM-Basic) + RGB-TTL Monitor + 2x320 kb Diskstation + 16 kb CMOS RAM (akkugep.) + prof. Textver. + Turbo Pascal (Originalprogramme!) + x-Software (Disketten) zu verkaufen. Hardware einwandfrei und in la hp-Qualität (ehrllich!), nur zusammen 3200.-DM VB. 48 kb Sinclair Spektrum + Kass.-Prog. 150.-DM. hp-15c 170.-DM. Beide Geräte neuwertig, da (fast) ungebraucht.
Rainer Trunz, Pfingstweidweg 10, 6455 Erlensee, ☎06183/71081.

Verkaufe: Drucker HP 82143 A, X/F-Modul, Quard-RAM. Suche: HP-IL Converter 82166A sowie "Creating Your Own HP-41 Bar Code".
Heinz Schmidt, ☎02845/6204

Verkaufe: OSBORNE 1 mit 80 Zeichen-Karte und externem Monitor + Drucker EPSON MX-80F/T. VB 2000.-DM.
Michael Roßkotten (682), Am Röttchen 9, 4060 Viersen 11, ☎02162/56583 Wo-Tag nach 18.00 Uhr, ☎02151/448606 tagsüber.

HP 110 dt. mit Lotus 1-2-3, Terminal, Memomaker, HPLink u.a. 3900,-VB; für Epson: HP IL-Interfacekarte 380,-VB, HP IL-Comverter neu 380,-VB, HP 41 Advantage - ROM neu VB.
Achim Bähr, Holzkirchener Str. 1, 8029 Arget

Verkaufe: Prisma 9/83, Key Notes 3/82, Standart-Sammlung, HP 32 E Bed.Handbuch, HP Allg. Handbuch gegen Gebot. Viedeo/F 82163 B (300,-), XF 82180 A (150,-), TIME 82182 A (150,-) Drucker 82162 A (650,-) Kass. 82161 A (700,-), Netzteil 82066 (25,-)
 Suche: Defekte Akkus und Netzteile, Key Notes 1983
Peter Winter, TEL.: 089/883221

Verkaufe: HP 82161A + 10 HP-Kassetten 900,-DM, Module für HP 75: Textformatter 190,-, Visicalc 390,-, HP 75 Forth 390,-DM, Verkauf wegen Systemwechsel.
TEL.: 06008/7235 ab 18.00 Uhr.

HP 110 (mit Memomaker, Lotus, Datacomm.) kompl. mit Handbüchern 2850,-DM incl. MwSt.
Jörg Gerhardt, TEL.: 069/666 3999

Verkaufe HP-71 Dis.-Assembler Listings (PPC), 55 Seiten, in stabilen Plastikordner, alle Seiten in Plastikfolien, für 30,- (Verrechnungsscheck)
Dennis Föh, Drangstweg 1, 2190 Cuxhaven, TEL.: 04721/24129 nach Absprache (telefonisch)
 Preis versteht sich zzgl. Porto

An Weihnachten denken, ich verkaufe: Bücher, The Basic HP-71, 35,-, Dienstprg. für den HP-41 (Vieweg), 10,-, HP-41 in der Praxis, 20,-DM, Anwenderhandbuch HP-41 (Vieweg), 20,-, Sprachinterface (fertig geb., s. PRISMA 84.9.31), 160,-, GAMES/STAND. Mod. (1 Gehäuse), VB 90,-DM, Barcode. HP 82153A, VB 220,-, Mem.Mod. HP 82420A, VB 130,-, Netzteil HP 82066B, VB 25,-.
TEL.: 07131/484738

Verkaufe für HP 71B: ein univ. eben. Stabwerkprgm. bis zu 40 Lastfälle (75 Stäbe, 40 Knoten) mit bel. Belastungen. Nach Theo 1. Ordnung werden die versch. Lastfallgruppen in ungünstiger Weise überlagert.
 Nach Theo 2. Ordnung Auswahl der min. u. max. Werte aller Lastfälle. 950,-DM.
Reiner Sievers, TEL.: 02802/1596

Verkaufe: HP 71B, ek Speicher-Modul, Mathe.Modul, IL-Modul, Disc-Drive 9114, IL-Konverter 82166-A, div. Zubehör, komplett für 2800,-DM.
 Evtl. auch einzeln.
Ralf Mulch, TEL.: 06151/312849 abends.

Advanced Pac-Screen, neu, 600,-DM/Computer-Literatur: Liste auf Anruf.
Lucke, TEL.: 040/401241

HP-Diskettenlaufwerk 9114B, Interfaces IL/RS 82164 A, IL/Video 82163 B, IL/FX80 82905, HP 75c mit Visi, IO und 8kB VHB 2000
nach 18.00 Uhr, TEL.: 04349/8118

Verkaufe aus Restbeständen günstig:
 Zubehör HP 41:
 X-Function-Modul 110,-, TIME 90,-, Advantage 120,-, Math/Stat 120,-, Games 35,-, Standart Mod. 25,-, Machine 90,- etc.
 IL-Drucker 750,-, CT-Portextender 100,-, Bar-Code Leser 230,-, IL-Drive 820,-, HP 41 CV 320,-, HP 41 CX 545,-DM etc. alles jeweils kpl. mit Garantie.

Zubehör IBM Computer:
 ADI-Monitor 14 14,-DM neu 390,-(bernstein), Microsoft Maus + PC-Paintbrush etc. 390,-, Heracles Card Impec org. 260,-. Weiteres auf Anfrage
TEL.: 07121/33320156 am Wochenende: TEL.: 0711/5300807

1 HP 71 B mit Finanz- + Mathemodul f. 1050,-DM.
Rolf Hansmann, Limburger Str. 15, 6242 Kronberg, TEL.: 06173/62405.

Verkaufe AEC ROM neu 200,-DM
Martin Meyer(1000), Robert-Stolz-Str. 5, 6232 Bad Soden 1, TEL.: 06196/23013 abends TEL.: 06196/87-2051 tags-über.

Verkaufe HP 71B 4k Speicher, Mathemodul IL Modul, Disc-Drive 9114 A, IL-Konverter 82166 A, diverses Zubehör. Insgesamt 2500,-DM.
Ralf Mulch, Kronstedter Weg 14, 6100 Darmstadt, TEL.: 06151/312849 abends

Verk. f. HP 41: Thermodrucker: HP 82143A. Opt. u. techn. einwandfrei. 320,-.
J. Goriss, TEL.: 089/1286698 (bis 16.00) oder TEL.: 82694 (ab 19.00).

.END.

ATARI ST

Unsere Wartezeit in Bezug auf Zuschriften zum Thema ATARI ST ist mittlerweile abgelaufen, es wird Zeit, Bilanz zu ziehen.

Die Redaktion erreichten - bis auf eine einzige - durchweg positive Zuschriften. Von diesen haben leider nur sehr wenige erkennen lassen, daß der Absender aktiv an der Gruppe teilnehmen will - ein potentieller Gruppenleiter war nicht dabei. - Soweit die Lage innerhalb des CCD.

Ca. 40 - 50 ATARI-ST-Besitzer wären an einer CCD-Mitgliedschaft interessiert, wenn es eine entsprechende Gruppe gäbe. Nichtmitglieder können wir bei unseren Überlegungen jedoch nicht berücksichtigen...

Kurzum: Mangels Masse erscheint es nicht gerechtfertigt, den CCD um eine ATARI-ST-Gruppe zu erweitern!

Ich persönlich besitze keinen ATARI-Rechner, erkläre mich aber bereit, Kontakte zwischen ST-Usern des CCD zu vermitteln. Mitglieder, die mitmachen wollen, senden mir bitte einen SAFU und eine Erklärung, die mir die Weitergabe der Adresse an andere CCD-Mitglieder erlaubt.

Eigentlich erübrigt sich damit die Stellungnahme zum Brief von Peter Ehrenberg (PRISMA 7/86), da es aber um Grundsätzliches geht, möchte ich es dennoch tun.

Die Frage lautet: Soll ein weiterer Rechner in unsere Palette aufgenommen werden und wenn ja, welcher?

- Der Club entstand durch den Zusammenschluß von Leuten, die den gleichen Rechner besitzen und gemeinsam damit verbundene Fragen/Probleme lösen wollten. Viele Mitglieder besitzen noch andere Rechner. Es ist daher nur konsequent, diese - bei entsprechender Nachfrage - innerhalb des nun schon mal bestehenden Interessenverbands auch zu unterstützen.

- Bei einem Blick auf unsere Clubadressen hätte Peter feststellen können, daß wir nicht nur die 40er-, 70er- und CP/M-Gruppe, sondern auch die 80er- und MS-DOS-Gruppe haben. Die CP/M- und MS-DOS-Gruppen bestehen nur zu einem kleineren Teil aus HP-Rechnern. Zur letzteren gehört so z.B. auch der IBM-PC...

3 von 5 Gruppen haben also nicht das Thema Taschenrechner, ja in 2 von 5 Gruppen gibt es sogar überwiegend Nicht-HP-Rechner...

- Das Translator Pac für den HP-71 ist nur eine bessere Krücke für diesen Rechner, interessante und wichtige HP-41-Funktionen werden nicht unterstützt, ein größerer Teil unserer 41er-Software läuft daher nicht auf dem 71er...

- Wie ich aus Erfahrung weiß, wird in mindestens 2-3 Firmen der HP-41 mit einer VAX/780 gekoppelt. Die haben ja auch keinen gemeinsamen Bus, aber dafür gibt es ja Schnittstellen-Konverter...

Welche Rechner kämen also in Frage?

- Betrachtet man die Preise für die HP-Serien 98xx, Integral, 1000, 3000, dann ist es verständlich, warum wohl die wenigsten unserer Mitglieder Beiträge für diese Rechner beisteuern könnten...

- Taschenrechner anderer Hersteller können unseren kaum das Wasser - Entschuldigung: die Bits - reichen. Die Handhelds gehören ohnehin meist zur MS-DOS-Gruppe. Bleiben

also aus Kostengründen nur noch PC's außerhalb der MS-DOS-Welt übrig.
 Was spricht also gegen ATARI-ST / AMIGA ?

Sie sind benutzerfreundlich zu bedienen und es gibt dafür Software, die preiswert und gut ist. Ein aufgemetzter Abakus wie z.B. der IBM-PC/XT ist dagegen eine Zumutung!

Als letzten Punkt möchte ich noch folgendes anführen:

CCD - Computerclub Deutschland dieser Name, den wir unserer Vereinigung gegeben haben stellt einen großen Anspruch. Mit HP-(Taschen)-Rechnern allein können wir dem kaum Rechnung tragen...

Henry (786)

.END.

Programmsammlung

Die ersten Arbeiten für die Programmsammlung Vermessungswesen HP-41 und Hp-71 sind erfolgt. Deshalb möchte ich hier die Form der Dokumentation kurz vorstellen. Ihr findet im vorliegenden Prismaheft einen Beispielausdruck. Auf Anfrage übersende ich eine Liste der vorhandenen Programmen, wenn ein Freiumschlag beiliegt. Die Programme kann ich auf eine übersendete Mini-Cassette oder eine 31/2 Zoll Diskette überspielen. Falls man kein Cassettenlaufwerk hat, würde ich auch einen Listingsausdruck erstellen. Eine Veröffentlichung im Prisma halte ich nicht für zweckmäßig, da das Eintippen der Programme doch erheblichen Zeitaufwand bedeutet. Ich bitte deshalb alle HP Benutzer im Fach Vermessung mir ihre Programme auf Cassette oder als Listing mit einer kurzen Dokumentation zu übersenden, um möglichst viele Spezialanwendungen abdecken zu können.

Damit eine möglichst optimale Betreuung im Anwendungsbereich Vermessungswesen erfolgt, werden diese Programmsammlungen getrennt von den Programmbibliotheken geführt.

Es ist auch geplant, ein Adressenverzeichnis der Anwender aus dem Bereich Vermessungswesen zu erstellen, um den Kontakt untereinander weiter zu verbessern. Darüberhinaus bin ich gerne bereit jedem HP Benutzer mit Rat und Tat zur Seite zu stehen. Damit sich jeder ein Bild machen kann, möchte ich hier kurz die bei mir vorhandene Hardware vorstellen:

HP 41 CX, IL-Modul
 HP 71 B, IL-Modul, auf 32 K ausgebaut
 Din A4 Drucker HP 82905B
 Pac-Screen-Video-Interface
 Diskettenlaufwerk HP 9114
 Cassettenlaufwerk HP 82161A
 Plotter HP 7470A

Textverarbeitungsmodul in Softwareform
 Assembler-Forthmodul HP-Modul
 HP IDS Band 1-3

Da ich mit den Programmsammlungen schon gut ausgelastet bin, bitte ich darum keine IDS-Kopien bei mir zu bestellen. Fragen beantworten ich aber auch aus diesem Bereich gerne. Ich bitte aber darum, bei Anfragen die vorhandene Hardware kurz zu erläutern, damit ich eine Vorstellung davon habe, was man als bekannt voraussetzen kann.

Berater im Supportzentrum

Hewlett-Packard ist seit Jahrzehnten auf den Gebieten der Computer- und Meßtechnik international führend. 82.000 Mitarbeiter, davon 4.800 in Deutschland, tragen diesen Erfolg. Trotz einer ungewöhnlich dynamischen Entwicklung ist es gelungen, unseren unbürokratischen, offenen und auf Eigeninitiative beruhenden Arbeitsstil in einer überschaubaren Organisation zu bewahren. Die HP-Konditionen – Gehalt, ein Paket von Sozialleistungen und Sicherheiten, Aktiensparen, Gewinnausschüttung etc. – entsprechen der Leistungskraft unseres Unternehmens. Unser Führungsstil ist betont kooperativ und läßt Ihnen viel Freiraum für die Verwirklichung neuer Ideen. Kommen Sie zu HP, wenn Sie sich mehr zutrauen. Auch Absolventen und qualifizierten Praktikern ohne Studium geben wir bei fundierten Kenntnissen gute Einstiegschancen.

Bewerbung

Ihre Bewerbung richten Sie bitte an folgende Adresse:

HEWLETT-PACKARD GMBH Vertriebszentrum Ratingen Personalabteilung
Berliner Straße 111, 4030 Ratingen
Tel. 0 21 02/494-204 (Thomas Scharrenberg)

Aufgabe

In unserem deutschen Support-Zentrum in Ratingen unterstützen wir unsere Kunden bei der Lösung ihrer Probleme in den Bereichen Software, Hardware und Applikationen mit HP-Computersystemen. Derzeit suchen wir einen Mitarbeiter für die Beratung von Produkten der Serien 10, 40 und 70 und der entsprechenden Peripherie. Die Aufgabe beinhaltet die Unterstützung unserer Fachhändler im Hinblick auf Anwendungs- und Programmierfragen sowie technische Spezifikationen. Darüber hinaus hat der Mitarbeiter die Aufgabe, in diesem Fachbereich Händlerschulungen durchzuführen.

Qualifikation

Sie sollten über eine fundierte theoretische Ausbildung (z.B. ein Hochschul- oder Fachhochschulstudium) und gute Englischkenntnisse verfügen. Unerlässlich sind einschlägige Erfahrungen in der Nutzung und Anwendung von HP-Taschenrechnern und -Taschencomputersystemen. Daneben erwarten wir die persönliche Eignung für eine beratungsintensive Tätigkeit und gute didaktische Fähigkeiten.



**HEWLETT
PACKARD**

Spitzentechnologie
entwickeln · fertigen · verkaufen

Inhaltsverzeichnis für HP 41

Prg.Nr./Programmkurzinhweis

- 001 Dreiecksberechnungen
- 002 Koordinaten der Klothoide
- 003 Festlegung der Klothoide
- 004 Parallele zur Klothoide
- 005 Eilinie
- 006 Gleisverziehung
- 007 Gradientenberechnung
- 008 2-teiliger Korbbogen
- 009 Abzweigradius Bogenweichen
- 010 Kreisbogenelemente
- 011 Klothoidenschnittpunkte
- 012 Ausgleichung von Höhennetzen
- 013 Berechnung von Klothoidenelemente
- 014 Kreisbogenabsteckelemente
- 015 Schnittpunktberechnung Gerade-Kreis
- 016 Umformung und Kleinpunktberechnung
- 017 Außendienst-Koordinatenverwaltung und Berechnungen
- 018 Außendienst-Koordinatenverwaltung und Berechnungen Vers.2
- 019 Tachymeterdatenerfassung
- 020 Tachymeterdatenausdruck

- 021 Gauß-Krüger-Koordinaten in geographische Koordinaten
- 022 Geographische Koordinaten in Gauß-Krüger-Koordinaten
- 023 Gauß-Krüger-Koordinaten in benachbartes Streifensystem
- 024 Strenge Ausgleichung des Standardpolygonzuges
- 025 Einzelpunktausgleichung
- 026 Elektromagnetische Streckenreduktion und NN-Reduktion
- 027 Elektrooptische Streckenreduktion und NN-Reduktion
- 028 Verebnung ellipsoidischer Richtungswinkel und Strecken
- 029 Geradenschnitte 3 Pkt. mit Richtungswinkel, 4 Pkt. oder 5 Pkt.
- 030 Flächenberechnung aus Koordinaten

- 031 Geradenschnitt 4 Punkte mit parallelen Abständen
- 032 Querprofilflächen und Erdmassenberechnung
- 033 Geradenschnitte mit Böschungssteigungen
- 034 Vorwärtsschnitt mit Richtungswinkeln
- 035 Rückwärtsschnitt mit Winkeln
- 036 Bogenschnitt

Inhaltsverzeichnis für HP 71

Nr./Name/Programminhalt

- 001 LADENEU2/Startprogramm zum Laden des Rechners
- 002 KOO/Lex-File zur Koordinatenverwaltung
- 003 PKTVERW2/Basic-Programm zur Punktdaten- und Programmverwaltung
- 004 GRAPHIK2/Plotten und Flächenberechnung kleiner Verm.-Aufträge
- 005 KOOTRAFO/Koordinatentransformation Meridianstreifensysteme
- 006 TRIGAUS2/Trigonometrischer Außendienst (Kleinberechnungen)
- 007 KLOTSPK2/Schnittpunkte Klothoide-Gerade, Kreis u. Klothoide
- 008 DUE41712/Umsetzungsprogramm für Koordinatendateien HP 41 <-> HP 71

Ulrich Kulle
Schnuckentrift 14
3000 Hannover 51

.END.

Clubnachrichten

CCD-Messestand auf der MICRO - COMPUTER '87 vom 19. - 23. Mai 1987

Auch dieses Jahr wird der CCD wieder auf der Frankfurter MICRO-COMPUTER Messe mit einem eigenen Stand vertreten sein. Diese Gelegenheit werden wir nutzen, um unseren Club einem breiten Publikum darzustellen. Außerdem werden auch wieder viele interessante Rechnersysteme und Programme auf unserem Messestand zum Anschauen und Ausprobieren zur Verfügung stehen.

Das reicht vom HP-18 über HP Serien 40, 70, 80 bis zu den Personal Computern mit CP/M und MS-DOS - und natürlich auch viele Peripheriegeräte.

Jetzt sind deshalb Eure Anregungen, Ideen und Vorschläge gefragt, was wir auf dem Messestand alles bieten könnten oder sollten. Außerdem suchen wir auch interessierte Clubmitglieder, die auf unserem Messestand aktiv mitarbeiten. Selbstverständlich mit einer entsprechenden Aufwandsentschädigung.

Zuschriften, Ideen oder Zusagen bitte an:
Alf-Norman Tietze
Im Sechholder 100
6000 Frankfurt/M. 80
☎069/393559
Box: GEO1:A.N.Tietze

Wie immer dient der Messestand auch als "heißer" Treffpunkt zum Erfahrungs- und Informationsaustausch, weil dort Anwender aus vielen Bereichen anzutreffen sein werden.

Als Nebenprogramm wird - wie in den letzten beiden Jahren auch - eine Vortragsveranstaltung angeboten. Dafür werden Vortragende gesucht, die ein kleines Referat (max. 1 Stunde) über eine Rechneranwendung oder über eine bestimmte Hardware halten. Der Tag dafür (ein Tag) ist noch nicht festgelegt.

Mit der Hoffnung auf eine rege Teilnahme und somit auch auf einen guten Erfolg

Alf-Norman Tietze

Stapel / Speicher und 'Vorhang' / Datenregister / Block-Operationen / Matrizen- und Datenverarbeitung / Sortieren / Zufallszahlen / gebrochene Zahlen und Rundungsrechnung / Arithmetik und Algebra / Geometrie, Trigonometrie und Analysis / Zahlensysteme / Umrechnung von Maßeinheiten und Abkürzungsvorschläge / Statistik und Kombinatorik / Uhr, Kalender und Finanzen / Kartenleser und Lesestift / der Drucker / Schrift-Fahnen / Auflösung von Gleichungen nach verschiedenen Variablen / Synthetisches Laden von Byte / Kryptographie / Anhang und dann noch ein 10-seitiges Stichwortverzeichnis mit über 1000!! Stichworten.

Das Buch ist voll mit Tips aller Art. Von so einfachen Dingen wie Generieren der nicht implementierten Funktion 'X>=Y?' oder Beschriften der Tastaturschablonen; über kleine Programme (wenige Byte) wie 'nicht-blinkende Pausen variabler Länge' oder Simulation der 'P<>S'-Funktion des HP-67/97 bis zur Datum- und Kalenderberechnung.

Die 27 Kapitel enthalten insgesamt 381 Unterprogramme und über 15 K Byte an Routinen und Programmen, wobei die Mehrzahl in der Klasse unter 50 Byte liegt.

Dieses Buch muß man einfach immer mal wieder zur Hand nehmen und darin blättern, es fällt einem jedes mal etwas Neues auf. Selbst für einen 'alten Hasen' hat das Buch immer wieder kleine Überraschungen parat, die einem das '41-Leben' leichter machen.

Werner Meschede
Sorpestr.4
5788 Siedlinghausen

.END.

Rolf verkauft 71?

Ich hatte im letzten (wie in diesem) Prisma eine Anzeige über den Verkauf eines HP-71. Wegen einiger Spekulationen zu diesem Thema möchte ich betonen, daß der HP-71 immer noch ein Rechner ist, mit dem ich gerne arbeite. Mein Arbeitgeber sah das ein und rüstete mich dienstlich mit einem HP-71 aus. Wozu jetzt 2 Rechner behalten? Das ist alles was hinter dieser Sache steckt. Ich gebe also den HP-71 nicht auf! Es ist einer meiner Lieblingsrechner

Wie immer, Euer Rolf

.END.

Buchbesprechung

Dearing J.S.
Tricks, Tips und Routinen für Taschenrechner der Serie HP-41
deutsche Ausgabe von H. Dalkowski
Heldermann Verlag, Berlin, 1984, 36,-DM.
ISBN 3-88538-801-4

Eine Besprechung zu diesem Buch zu schreiben ist kaum möglich, denn es ist kein Buch im herkömmlichen Sinne, sondern das, was der Titel bereits sagt: eine große Sammlung von Tricks, Tips und Routinen, die von den Verfassern in der ganzen Welt zusammengetragen wurden.

Die Vielfältigkeit zeigt sich bereits in den Titeln der 27 Kapitel: Elementare Funktionen und Operationen / Tips zur Programmierung / Startbedingungen und Eingabeaufforderungen / die Anzeige / Behandlung von Alpha-Ketten / Flags und Töne / Umgang mit dem

Serie 80 Service

**SERIE 80 ASSEMBLER
KURS
Folge 4**

Ich möchte nun sofort zum Kapitel 4 Controllers übergehen und muß bei dieser Gelegenheit zugeben, daß ich mich mit diesem Kapitel kaum beschäftigt habe.

Der Teil aus Kapitel 3 des Handbuches, der sich mit der Bearbeitung von Zahlenformaten und Strings beschäftigt, kann meines Erachtens von mir Übergangen werden.

Wie mittlerweile bekannt sein dürfte, handelt es sich bei der Serie 80 von HP um Multiprozessormaschinen. Der Zentralprozessor koordiniert bekanntlich über die I/O-Ports (177400 - 177777) das Zusammenspiel der Zusatzprozessoren in ihren jeweiligen Aufgabenstellungen.

CRT-Controller

Der CRT-Controller und die CPU kommunizieren über 4 RAM-Adressen nämlich CRTBAD DAD 177701, CRTSAD DAD 177700, CRTDAT DAD 177703 und CRTSTS DAD 177702. Diese Controller haben die folgenden Funktionen:

CRTBAD DAD 177701

Wird in 177701 eine 2 Byte Adresse geladen, so wird der CRT-Controller veranlaßt seinen Adressenpointer mit dieser Adresse zu laden. Man kann diesen Vorgang am ehesten mit einer Drehtür, durch die eine Adresse für den CRT-Controller durchgereicht wird, vergleichen.

CRTSAD DAD 177700

Laden einer Adresse in 177700 veranlaßt den CRT-Controller die aktuelle Bildschirmseite mit dieser Adresse zu beginnen. Man kann den Bildschirm über 177700 veranlassen zu scrollen oder auch Seitenweise zu springen. Im Graphicmode ist die Adresse CRTSAD 177700 ohne Funktion.

CRTDAT DAD 177703

Über 177703 können Datenbytes (Ziffern und Texte) byteweise aus dem Bildschirmspeicher entweder Ein- oder Ausgelesen werden. Begonnen wird mit der Adresse, auf der der CRT-Controller steht, oder auf die er mit Hilfe von CRTBAD 177701 adressiert wurde. Beim Ein- bzw. Auslesen wird der CRT-Controller automatisch inkrementiert.

Wichtig ist die Abfrage des CRT-Controllers, dessen Statusbyte über die Adresse CRTSTS DAD 177702 abgefragt werden kann. Ist das Least Significant Bit (Bit 0) gesetzt, so muß mit dem Zugriff gewartet werden bis der Controller frei ist. Bevor jedoch ausgelesen oder geschrieben werden kann, mit ein Byte mit gesetztem LSB (Bit 0) in CRTSTS geschrieben werden, um dem CRT-Controller anzuzeigen, daß man lesen bzw. schreiben möchte. Wieder muß abgewartet werden, bis der Controller frei ist. Erst jetzt können Daten Ein- bzw. Ausgelesen werden.

CRTSTS DAD 177702

Über die Adresse 177702 kann der Status des CRT-Controllers abgefragt werden, ebenso lassen sich über ein Statusbyte Grundfunktionen des CRTs wie Pagesize, Invers, Power up and down etc. einstellen. Zwei Tabellen auf den Seiten 4-2 und 4-3 zeigen, über welche Bits man die einzelnen Funktionen bzw. Statusinformationen erhalten kann. Als Beispiel möge hier die Funktion des Bit 2 dienen. Wenn es gesetzt ist, wird die Hoch-

```

1000 ! *****
1010 ! *
1020 ! *
1030 ! *           I S F O - S O F T W A R E
1040 ! *
1050 ! *           B i n a r y :   R O M R E A D
1060 ! *
1070 ! *   (c) by Ingolf Seelemann & Fokke Oosterhoff      1986
1080 ! *           Horststr. 12           Billrothstr. 36
1090 ! *           4250 Bottrop 2         4300 Essen 1
1100 ! *
1110 ! *           Tel.: 0 20 45 / 8 25 24
1120 ! *
1130 ! *
1140 ! *****
1150 !
1160 ! #####
1170 ! #
1180 ! # Das Binary dient in erster Linie als Debug-Hilfe fuer einen Disas- #
1190 ! # sembler. Der Vergleich des reassemblierten Object Codes mit dem #
1200 ! # Original ROM Code soll erleichtert werden. #
1210 ! # Ferner kann dieses Binary gut mit einem EPROM-Burner eingesetzt wer- #
1220 ! # den. #
1230 ! #
1240 ! # Folgende Befehle wurden implementiert: #
1250 ! #
1260 ! #   1. ROM LIST           gibt eine vollstaendige Liste aller ROMs aus #
1270 ! #   2. SELECT ROM # (Dec) Waehlt ein ROM zur Selektion aus #
1280 ! #   3. A$=ROM READ      Liest ROM aus und uebergibt Daten nach A$ #
1290 ! #   4. COPYR           gibt die Uebeltaeter dieser Software aus #
1300 ! #
1310 ! # Beispielprogramm: #
1320 ! #
1330 ! #   10 DIM Rom$[8192] #
1340 ! #   20 CLEAR @ ROM LIST #
1350 ! #   30 DISP @ DISP "Waehle ROM" @ INPUT R #
1360 ! #   40 SELECT ROM R #
1370 ! #   50 Rom$=ROM READ #
1380 ! #   60 DISP Rom$ #
1390 ! #   70 END #
1400 ! #
1410 ! #####
1420 !
1430 !   Date: 20. Feb. 1986
1440 !
2000 ! +-----+
2010 ! |
2020 ! | Als erstes folgt die sogenannte Binary Program Shell. Sie bildet |
2030 ! | den Kopf des Programmes, in dem nach dem Laden alle Informationen |
2040 ! | stehen, die fuer den Zugriff auf die Routinen wichtig sind. |
2050 ! | Die 5 Definitionen im Header muessen in dieser Reihenfolge im Kopf |
2060 ! | geschrieben werden !! Die Wahl der Texte ist frei, nicht jedoch die |
2070 ! | Reihenfolge Ihrer Bedeutung. |
2080 ! |
2090 ! +-----+
2100 !   GLO GLOBAL           ! Definiert Global File als Sprung-
2110 !                       ! adressen Lieferant. ( optional )
2120 !   ABS 60000 o. 115000 ! Assemblierung eines ROM bzw. absoluten
2130 !                       ! Binaries. ( optional )
2140 ! -----

```

spannung der Kathodenstrahlröhre abgeschaltet andernfalls wird die CRT eingeschaltet. Da bei abgeschalteter Röhre der Refreshzyklus der CRT wegfällt, sind Lese- und Schreiboperationen erheblich beschleunigt ! Das gilt auch für die Anwendung im BASIC mit den Keywords CRT ON und CRT OFF. Referenzen: 4-1 bis 4-3

Display Modes

Die CRT kann bekanntlich in 3 verschiedenen Modi arbeiten, nämlich NORMAL, ALPHALL und GRAPHALL.

Im NORMAL Modus wird der vorhandene Memorybereich gesplittet. Die Adressen 000000 bis 010220 sind dem Alphabereich zugeordnet. Der Bildschirmspeicher besitzt dabei eine Kapazität von 54 Zeilen. Der Graphikbereich reicht von Adresse 010340 bis 037616. Der Grafikbereich entspricht einer Bildschirmbreite von 50 alphanumerischen Zeichen.

Im ALPHALL Modus reicht der Adressenbereich von 000000 bis 037560. Im GRAPHALL Modus ist der Gesamtbereich von 544*240 Punkten auf beide Normalbereiche aufgeteilt. Referenzen: 4-4 bis 4-8

Keyboard Controller

Der Keyboard Controller beaufsichtigt folgende RAM Locations: Keyboard Scanner, 4 Timer und den Beeper.

Keyboard Scanner

Der Keyboard Scanner fordert bei Tastendruck Service von der CPU an. Bei Service wird die Routine KEYSRV aufgerufen, die wiederum einen Sprung zum KYIDLE Hook über den Befehl JSB =KYIDLE ausführt. Ist der KYIDLE Hook nicht belegt, so wird unverzüglich mit KEYSRV fortgefahren. Interrupts werden abgeschaltet, Register gesichert und der Tastendruck, der vom Keyboard Controller erzeugt wird, wird über den I/O port KEYCOD gelesen. KEYSRV überprüft ob es sich um ein Reset handelt und führt es gegebenenfalls aus; andernfalls wird nachgesehen ob weitere Tasten gedrückt wurden. Wurden weitere Tasten gedrückt, so wird der Keyboard Scanner reaktiviert und die gesicherten Register werden zurück gespeichert. Die CPU setzt ihren Ablauf dort fort, wo sie durch den Keyboard Interrupt angehalten wurde. Die Tastenanschläge werden in KEHIT gespeichert, solange, wie Bits in R17 und SVCWRD gesetzt sind.

Ist einmal eine Taste gedrückt, so werden keine weiteren Keyboard Interrupts gesehen bis eine 1 in die RAM Location KEYCOD geladen wird, welche den Keyboard Scanner reaktiviert.

Die Tabelle auf Seite 4-9 die Write und Read Funktionen für den I/O port KEYSTS.

Die I/O Location KEYCOD enthält den aktuellen Tastendruck, durch Schreiben einer 1 in KEYCOD wird der Keyboard Scanner reaktiviert.

Auf der Seite 4-10 ist die System Routine KEYSRV abgedruckt. Wer Keyboard Programmierer in Assembler schreiben möchte, sollte sich mit dieser Routine auseinandersetzen. Referenzen: 4-8 bis 4-10

Timer

Der Keyboard Controller hat ebenfalls Zugriff auf 4 Timer und auf 4 Register, jedes mit 8 BCD Ziffern geladen. Die Timer und Register werden mit einer Frequenz von 1 kHz überschrieben. Wenn Timer und Register überschrieben werden, sollte keine Lese oder Schreiboperation über CLKDAT erfolgen. Jeder Timer, der sein Zählregister erreicht, for-

2150		NAM 301,ROMR	! Binary Programm Nummer und Kurzname
2160	MYBP#	EQU 301	! Variable mit Binary # als Erleichterung
2170		DEF RUNTIME	! Pointer RUNTIME Tabelle. BYT 32,33 rel.
2180		DEF ASCII	! Pointer ASCII Tabelle. BYT 34,35 rel.
2190		DEF PARSE	! Pointer PARSE Tabelle. BYT 36,37 rel.
2200		DEF ERMSG	! Pointer ERMSG Tabelle. BYT 40,41 rel.
2210		DEF INIT	! Pointer INIT Routine. BYT 42,43 rel.
2500	!	+-----	
2510	!		
2520	!	Es muss hier die RUNTIME und die PARSE Tabelle folgen. Die Reihen-	
2530	!	folge, ob erst PARSE oder erst RUNTIME ist egal ! Die Pointer oben	
2540	!	stehen immer richtig ! Es muss im Binary der Delimiter 377,377 die	
2550	!	beiden Tabellen abschliessen ! Im ROM fehlt er ! Dieser Delimiter	
2560	!	schliesst den sogenannten relocateable Programmteil ab. Er macht es	
2570	!	erst moeglich, Binaries in variabler Reihenfolge zu laden. Deshalb	
2580	!	muss die PARSE und RUNTIME Tabelle im Programmkopf stehen.	
2590	!	Die Reihenfolge der PARSE Routinen hat der Reihenfolge der RUNTIME	
2600	!	Routinen zu entsprechen. RUNTIME Routinen, die kein PARSE erforder-	
2610	!	lich machen, erhalten in den PARSE Tabellen an entsprechender Stelle	
2620	!	den Platzhalter BSZ 2.	
2630	!		
2640	!	+-----	
2650	RUNTIME	BSZ 2	! BSZ 2 ==> Platzhalter in BPGMs. Wichtig!
2660		DEF COPYR.	! Pointer auf Runtime COPYR. Kein Parse
2670		DEF RLST.	! Pointer auf Runtime RLST. Parse
2680		DEF SELROM.	! Pointer auf Runtime SELROM. Parse
2690		DEF RMREAD.	! Pointer auf Runtime RMREAD. Kein Parse
2700	PARSE	BSZ 2	! BSZ 2 ==> Platzhalter in BPGMs. Wichtig!
2710		BSZ 2	! Platzhalter fuer COPYR.Kein Parse noetig
2720		DEF RLSTP	! Pointer auf Parse Routine fuer RLST.
2730		DEF SELROMP	! Pointer auf Parse Routine fuer SELROM.
2740		BYT 377,377	! Ende Parse und Runtime Tabelle.
2800	!	+-----	
2810	!		
2820	!	Nach dem Terminator 377,377 ist die Programmierung des Binaries	
2830	!	voellig frei ! Die im Handbuch angegebene Reihenfolge ist zwar nuetz-	
2840	!	lich, aber nicht zwingend. Im I/O-ROM als Beispiel steht die ASCII-	
2850	!	Tabelle und die Error Message Tabelle am Ende des Programmes und der	
2860	!	INIT Code mitten drin.	
2870	!	Es folgt die ASCII-Tabelle, deren Keyword-Reihenfolge natuerlich	
2880	!	der Reihenfolge der Runtime Routinen entsprechen muss ! Die Keywords	
2890	!	sind mit dem Befehl ASP und nicht mit ASC zu schreiben ! ASP bewirkt,	
2900	!	dass das letzte Zeichen ein MSB (invers) bekommt und damit fuer die	
2910	!	Maschine als Keyword erkennbar wird.	
2920	!	Die ASCII-Tabelle ist mit einem BYT 377 abzuschliessen.	
2930	!		
2940	!	+-----	
2950	ASCII	BSZ 0	! BSZ 0 ==> Kein Platzhalter noetig
2960		ASP "COPYR"	! Keyword # 1
2965		ASP "ROM LIST"	! Keyword # 2
2970		ASP "SELECT ROM"	! Keyword # 3
2980		ASP "ROM READ"	! Keyword # 4
2990		BYT 377	! Terminator Byte 377
3000	!	+-----	
3010	!		
3020	!	Hier folgt die Fehlermeldungstabelle. Fehlermeldungen in Binaries	
3030	!	besitzen hohe Error Zahlen. Die Tabelle ist so angeordnet, dass ab	
3040	!	255 Dec dekrementierend zunaechst 9 Plaetze fuer Warnings ohne Pro-	
3050	!	grammunterbrechung reserviert sind. Als Default sind hier 9 Byte	
3060	!	200 zu setzen. Es folgen ab ERRN 2460 die Error Messages.	
3070	!		
3080	!	+-----	
3090	ERMSG	BYT 200,200,200,200,200	! 9 Byte 200 als Platzhalter fuer die

dert Service von der CPU an. Er wird dann auf Null gesetzt, um eine neue Zählsequenz zu beginnen. Der Inhalt der Timer wird in 4 entsprechende Bytes abgelegt, von denen jedes 2 BCD Ziffern enthält.

Die höchste Interruptpriorität besitzt der Keyboard Scanner, es folgen Timer 0 bis 3.

Die Tabelle auf Seite 4-12 zeigt die Funktionen des Write und Read I/O ports CLKSTS. Die RAM Location CLKDAT enthält die Daten des gewählten Timers. Eine 4 Byte Multibyteoperation ist zum Auslesen bzw. Schreiben aus bzw. in CVLKDAT erforderlich. Bevor jedoch gelesen oder geschrieben wird, muß über das gesetzte Bit 7 von CLKSTS gecheckt werden, ob Timer zum Zugriff bereit sind.

Auf die Timer kann nicht, wie bei anderen Routinen üblich, über Hooks zugegriffen werden. SVCWRD muß periodisch abgefragt werden, ob irgendein Timer Service anfordert. Dies funktioniert nur dann, wenn man nie zum BASIC Interpreter zurückkehrt, weil dieser ebenfalls Timerinterrupts handelt.

Das Beispielprogramm auf Seite 4-13 zeigt wie die Systemuhr (Timer 0) ausgelesen wird. Aus der Basiszeit und dem Timerwert wird die aktuelle Zeit errechnet und wie üblich auf dem R12 Stack übergeben.

Die kleine Routine auf Seite 4-14 zeigt das Setzen des Timers 0 bei System Power up. Speaker

Der Speaker wird über die I/O Adresse KEYSTS kontrolliert. Die Bits 5 und 6 von KEYSTS erlauben den Speaker mit 1,2 kHz oder bei liebiger Frequenz zu piepen.

Referenzen: 4-11 bis 4-14

System Monitor

Meiner Meinung nach bilden ASSEMBLER ROM und SYSTEM MONITOR eine zusammengehörige Einheit! Auf die Dauer ist das Arbeiten in Assembler ohne System Monitor frustrierend. Der System Monitor bietet eine Menge features ohne die die Programmierung in Assembler ein hoffnungsloses Unterfangen werden kann. Auf die vielen Möglichkeiten die der System Monitor bietet wird im folgenden eingegangen.

Der wichtigste Befehl des System Monitor ist BKP bzw. BKP REL. Auf allen Adressen unter 200000 oct also auch im Betriebssystem und in ROMs lassen sich max. 2 Breakpoints setzen. Wenn der Program Counter auf einen Breakpoint läuft, wird die Abarbeitung des Maschinencodes unterbrochen und der Bildschirm bzw. Drucker geben einen Gesamts-tatus, wie auf Seite 5-2 beschrieben aus. Ebenso ist die Tastatur blockiert und lediglich einige Tasten dienen als Tipphilfe für Funktionen, die auf Seite 5-2 beschrieben sind.

Folgende wichtige Statusindikatoren werden an einem Breakpoint ausgegeben:

- PC: Program Counter, Inhalt von R4-R5. Kann über die Taste P verstellt werden! Man sollte aber wissen was man tut!
- DR: Aktueller Data Register Pointer
- AR: Aktueller Adress Register Pointer
- OV: Status des Overflow Flags
- CY: Status des Carry Flags
- NG: Status des Negativ Flags
- LZ: Status des Linken Nibbles (Left Digit Zero)
- RD: Status des Rechten Nibbles (Right Digit Zero)
- OD: Status des LSB (Least Significant Bit) Ungrade
- DC: Decimal Flag. CPU Status DEC oder BIN
- E: Inhalt Extended Registers

```

3100      BYT 200,200,200,200      ! Warning Meldungen
3110      ASP "ROM DOESN'T EXIST" ! Error 246D
3120      ASP "SELECT A ROM FIRST" ! Error 245D
3130      BYT 377
3200 ! +-----+
3210 ! |
3220 ! | Die INIT Routine laeuft u.a. beim Laden oder Speichern des Binaries
3230 ! | ab. Hier kann man Routinen ablaufen lassen, deren Ablauf unbedingt
3240 ! | gewaehrleistet werden soll.
3250 ! | INIT wird hier nicht benoetigt, deshalb unverzueglich RTN.
3260 ! |
3270 ! +-----+
3280 INIT      RTN
4000 ! +-----+
4010 ! |
4020 ! | PARSE Routinen zerlegen Keywords in ihre Bestandteile und sorgen
4030 ! | dafuer, dass die entsprechenden Token (Zwischencode fuer den Inter-
4040 ! | preter) in das BASIC Programm eingearbeitet werden.
4050 ! | Das Schema fuer diesen Code ist eigentlich recht simpel.
4060 ! | Fuer den hier verwendeten Befehl ROM LIST gelten folgende Token:
4070 ! |
4080 ! | Higher Adress: 371 Token fuer Binaerprogramm
4090 ! |                301 Binaerprogramm Nummer
4100 ! | Lower Adress:  002 2. Runtime Routine
4110 ! |
4120 ! | Mit einer Parse Routine fuer ein Keyword koennen durchaus mehrere
4130 ! | Runtime Routinen geparsed werden ! Siehe Beispielprogramm LINPUTS !
4140 ! |
4150 ! +-----+
4160 RLSTP    LDB R57,=371          ! Token fuer Binary nach R57
4170          LDB R56,=MYBP#        ! Binary Programmnummer nach R56
4180          LDB R55,R43           ! Aktuelles Token aus R43 nach R55
4190          STMI R55,=PTR2-      ! Token ueber Pointer 2 in BASIC einfueg.
4200          JSB =SCAN            ! Keyword "scannen"
4210          RTN                  ! Zurueck zum System
4220 ! +-----+
4230 ! |
4240 ! | Die Routine fuer das Statement ROM LIST hat die Aufgabe eine voll-
4250 ! | staendige ROM Tabelle auf dem Bildschirm zu erzeugen.
4260 ! | Unbekannte ROMs werden als Optionale ROMs erkannt und ausgegeben.
4270 ! | Die ROM Tabelle wird ab der Adresse ROMTAB (hoehere ROM Nummern)
4280 ! | zum variablen Ende der ROM Tabelle (Byte 377) abgearbeitet.
4290 ! |
4300 ! +-----+
4310          BYT 241
4320 RLST.    BIN
4330          JSB =CRTPOF          ! Bildschirm aus
4340          LDMD R14,=BINTAB      ! Startadresse (Offset) laden
4350          LDM R30,=ROMTAB       ! Anfang der ROM Tabelle laden
4360          LDM R32,=ROMEND       ! Ende der ROM Tabelle laden
4370 LOOP    LDMD R34,R30          ! ROM # nach R34
4380 SYS0    CMB R34,=000          ! SYSTEM ROM 0 ?
4390          JNZ SYS1
4400          LDM R36,=SYSR0
4410          JMP OUT*
4420 SYS1    CMB R34,=001          ! SYSTEM ROM 1 ?
4430          JNZ MKS
4440          LDM R36,=SYSR1
4450          JMP OUT*
4460 MKS     CMB R34,=016          ! MIKSAM ROM ?
4470          JNZ GRL
4480          LDM R36,=MKSR
4490          JMP OUT*
    
```

```

--- BKP1: Zeigt absolute Adresse für den 1. Breakpoint
--- BKP2: Zeigt absolute Adresse für den 2. Breakpoint
--- PTR1: Zeigt die Adresse des EMC Pointers 1
--- PTR2: Zeigt die Adresse des EMC Pointers 2
--- ROM: Gibt das aktuell selektierte ROM an
Ferner werden 100 octale Bytes angezeigt, die vorher durch MEM spezifiziert werden können. Innerhalb eines Breakpoints kann die MEM Funktion ebenfalls aufgerufen werden.
Außerdem werden die Inhalte aller CPU-Register angezeigt.
Auf den Seiten 5-4 bis 5-6 werden die weiteren Funktionen beschrieben. Beachtenswert ist die Step-Funktion, die schrittweises abarbeiten des Maschinencodes erlaubt und die Trace Funktion, die eine vorgegebene Anzahl Schritte abarbeitet.
Referenzen: 5-1 bis 5-6
*****
In Sachen Assembler Kurs
*****
Der Assembler Kurs geht seinem Ende entgegen. Leider ist nur wenig Resonanz bei mir angekommen. Ich habe kaum Rückmeldungen erhalten, ob der Assembler Kurs in dieser Form eine Hilfe dargestellt hat.
Es sollten zunächst einmal Grundlagen geschaffen werden. Beispiele habe ich deshalb bis jetzt nicht gegeben. In der nächsten Folge wird deshalb ein wirklich ausführlich kommentiertes Programmbeispiel von uns erscheinen. Es handelt sich um unser Binärprogramm ROMREAD, daß es erlaubt ein komplettes 8k ROM, im Gegensatz zu PEEK Funktionen, in nur 900 ms auszulesen. Vielleicht kann ich dann noch in einer abschließenden Folge allgemeine Probleme, die noch an uns herangetragen werden können, erörtern.

Ingolf Seelemann
-----
.END.
4500 GRL      CMB R34,=030      ! GERM LANGUAGE ROM ?
4510          JNZ ASSM
4520          LDM R36,=GRLR
4530          JMP OUT*
4540 ASSM     CMB R34,=050      ! ASSEMBLER ROM ?
4550          JNZ SYSEXT
4560          LDM R36,=ASSMR
4570          JMP OUT*
4580 SYSEXT   CMB R34,=070      ! SYSEXT EPROM (c) Andre Koppel
4590          JNZ MTX2
4600          LDM R36,=SYSEX
4610 OUT*     JMP OUTCRT
4620 MTX2     CMB R34,=260      ! MATRIX ROM 2 ?
4630          JNZ MTX1
4640          LDM R36,=MTXR2
4650          JMP OUTCRT
4660 MTX1     CMB R34,=261      ! MATRIX ROM 1 ?
4670          JNZ I/O
4680          LDM R36,=MTXR1
4690          JMP OUTCRT
4700 I/O      CMB R34,=300      ! I/O ROM ?
4710          JNZ EXT MST
4720          LDM R36,=I/OR
4730          JMP OUTCRT
4740 EXT MST  CMB R34,=317      ! EXTENDED MASS STORAGE ROM ?
4750          JNZ MSUS
4760          LDM R36,=EXTMS
4770          JMP OUTCRT
4780 MSUS     CMB R34,=320      ! MASS STORAGE ROM ?
4790          JNZ ED
4800          LDM R36,=MSUSR
4810          JMP OUTCRT
4820 ED       CMB R34,=321      ! ELECTRONIC DISC ROM ?
4830          JNZ TST
4840          LDM R36,=EDR
4850          JMP OUTCRT
4860 TST      CMB R34,=340      ! TEST ROM ?
4870          JNZ AP2
4880          LDM R36,=SYSEX
4890          JMP OUTCRT
4900 AP2      CMB R34,=347      ! ADVANCED PROGRAMMING ROM 2 ?
4910          JNZ AP1
4920          LDM R36,=APR2
4930          JMP OUTCRT
4940 AP1      CMB R34,=350      ! ADVANCED PROGRAMMING ROM 1 ?
4950          JNZ PLT
4960          LDM R36,=APR1
4970          JMP OUTCRT
4980 PLT      CMB R34,=360      ! PLOTTER ROM ?
4990          JNZ OPT
5000          LDM R36,=PLOT
5010          JMP OUTCRT
5020 OPT      CMB R34,=377      ! Ende ROM TABLE ?
5030          JZR LSTEND      ! Wenn ja, dann Ende der Tabelle
5040          LDM R36,=OPTR      ! Sonst unbekanntes ROM ausgeben
5050 !
5060 OUTCRT   STMD R36,X14,STRADR ! Textadresse in R36 speichern
5070          LDMD R26,X14,STRADR ! und nach R26 laden
5080          ADMD R26,=BINTAB ! BINTAB addieren ( absolut machen )
5090          PUMD R30,+R12      ! R30 auf R12 sichern
5100          PUMD R32,+R12      ! R32 auf R12 sichern
5110          LDM R36,=033,000 ! Stringlaenge in R36
5120          JSB =OUTSTR      ! Auf den Bildschirm

```

```

5130      POMD R32,-R12      ! R32 vom Stack abholen
5140      POMD R30,-R12      ! R30 vom Stack abholen
5150      ICM R30            ! um 2 Byte inkrementieren
5160      ICM R30
5170      CMM R30,R32        ! Vergleiche ob Ende der ROM Tabelle
5180      JZR LSTEND         ! JMP, wenn Ende der Tabelle
5190      GTO LOOP          ! Sprung nach LOOP fuer naechstes ROM
5200 LSTEND JSB =CRTPUP      ! Bildschirm wieder einschalten
5210      RTN                ! Zurueck zum System
5220 ! +-----+
5230 ! |
5240 ! |      Texttabelle fuer die Bildschirmausgabe mit der Routine OUTCRT |
5250 ! |
5260 ! +-----+
5270 SYSRO  ASP " SYSTEM ROM      # 000 "
5280 SYSR1  ASP " SYSTEM ROM      # 001 "
5290 MKSR   ASP " MIKSAM ROM      # 014 "
5300 GRLR   ASP " GERM. LANGUAGE ROM # 024 "
5310 ASSMR  ASP " ASSEMBLER ROM    # 040 "
5320 SYSEX  ASP " SYSEXT-EPROM    # 056 "
5330 MTXR1  ASP " MATRIX ROM PART I # 176 "
5340 MTXR2  ASP " MATRIX ROM PART II # 177 "
5350 I/OR   ASP " INPUT/OUTPUT ROM # 192 "
5360 EXTMS  ASP " EXTEND. MASS STORAGE # 207 "
5370 MSUSR  ASP " MASS STORAGE ROM # 208 "
5380 EDR    ASP " ELECTRONIC DISC ROM # 209 "
5390 TEST   ASP " TEST ROM        # 224 "
5400 APR2   ASP " ADV.PRGRAMMING ROM II# 231 "
5410 APR1   ASP " ADV.PRGRAMMING ROM I # 232 "
5420 PLOT   ASP " PLOTTER ROM      # 240 "
5430 OPTR   ASP " OPTIONAL ROM     # ??? "
5440 BLSPR  ASP " BLUE SPRUCE ROM  # ??? "
5450 ! -----
5460 STRADR BSZ 2            ! Platzhalter fuer Stringadresse
6000 ! +-----+
6010 ! |
6020 ! | Die Routine SELROM parsed das Statement SELECT ROM. Es wird mit der |
6030 ! | Routine NUMVA+ ein SCAN ausgefuehrt. Des weiteren wird ein numeri- |
6040 ! | scher Wert oder Variable erwartet und als Real Zahl auf dem R12 |
6050 ! | Stack an die zugehoerige Runtime Routine SELROM. uebergeben. |
6060 ! |
6070 ! +-----+
6080 SELROM PUBD R43,+R6     ! Token auf R6 Stack sichern
6090      JSB =NUMVA+        ! Keyword SCAN und Value auf R12
6100      JEZ ERR           ! JMP wenn Fehler
6110      LDB R57,=371      ! Token fuer Typ BPGM
6120      LDB R56,=MYBP#    ! Binary Program Number laden
6130      POBD R55,-R6      ! Token fuer Routine von R6 holen
6140      STMI R55,=PTR2-   ! BASIC Token in Programm einordnen
6150      RTN                ! Fertig
6160 ERR    POBD R43,-R6     ! Token zurueckholen ! Stack bereinigen !
6170 ERR+   JSB =ERROR+     ! Error Routine aufrufen
6180      BYT 88D           ! Error 'BAD STATEMENT'
6500 ! +-----+
6510 ! |
6520 ! | Die Runtime Routine fuer das Keyword SELECT ROM ist mit dem Attribute |
6530 ! | 241 als BASIC Statement gueltig auch hinter THEN definiert. |
6540 ! | Sie hat die Aufgabe, den in der Parse Routine erhaltenen Wert in den |
6550 ! | Zwischenspeicher ROM# abzulegen. |
6560 ! | Zunaechst sorgt die Routine ONEX fuer die Umwandlung der Real Zahl |
6570 ! | auf R12 in eine Integer Zahl in R46,R47. R46 enthaelt als LSB die |
6580 ! | ROM Nummer, R47 muss bei gueltiger ROM Nummer auf 0 stehen. |
6590 ! | In der Routine NXTROM wird die gewaehlte ROM Nummer auf Vorhanden- |

```

```

6600 ! | sein in der ROM Tabelle ueberprueft. Gegebenfalls erfolgt eine
6610 ! | Fehlermeldung. Bei gueltiger ROM Nummer wird diese in ROM# gespei-
6620 ! | chert. Achten Sie darauf, dass ein Zugriff auf Speicherstellen oder
6630 ! | Routinen im eigenen Binary nur indiziert ueber BINTAB (hier R14 )
6640 ! | ablaufen kann. z.B. STBD R46,X14,ROM# oder JSB X14,TEST
6650 ! |
6660 ! +-----+
6670          BYT 241          ! Attrib.: BASIC Keyword erlaubt nach THEN
6680 SELROM.  JSB =ONEX       ! Real vom R12 Stack in Integer umwandeln
6690          LDMD R14,=BINTAB ! Startadresse des Binary holen
6700          CMB R47,=0       ! Vergleiche, ob MSB=0
6710          JNZ ERR+        ! JMP, wenn ROM# > 255 Dec
6720          LDM R76,=ROMTAB  ! Beginn der ROM Tabelle laden
6730 NXTROM  LDMD R57,R76     ! ROM Nummer laden
6740          CMB R57,R46     ! Vergleiche ROM Tabelle mit SELECTED ROM
6750          JZR OKAY        ! JMP, wenn ROM vorhanden
6760          ADM R76,=2,0    ! 2 addieren fuer naechstes ROM
6770          CMB R57,=377    ! Vergleiche ob Ende der ROM Tabelle
6780          JNZ NXTROM      ! JMP, wenn Tabellenende nicht erreicht.
6790          LDB R57,=MYBP#  ! Binary Programm Nummer laden
6800          STBD R57,=ERRBP# ! In RAM Position speichern
6810          LDB R57,=377    ! Default fuer ROM# laden
6820          STBD R57,X14,ROM# ! Default = undefiniert in ROM# speichern
6830          JSB =ERROR+     ! Error Routine aufrufen
6840          BYT 246D        ! Fehler Meldung 'ROM DOESN'T EXIST'
6860 OKAY   STBD R46,X14,ROM# ! ROM Nummer in ROM# speichern
6870          RTN             ! Zurueck zum System
7000 ! +-----+
7010 ! |
7020 ! | Die Runtime Routine fuer das Keyword ROM READ ist mit dem Attribute
7030 ! | 56 als String Funktion und mit dem Secondary 0 = keine Parameter
7040 ! | definiert. Zunaechst wird auf den Default in ROM# ueberprueft. Wenn
7050 ! | Default, so wurde vorher kein ROM selektiert und es erfolgt eine
7060 ! | Fehlermeldung.
7070 ! | Bei OK beginnt die Vorbereitung des Auslesevorganges. Nach der ROM-
7080 ! | Selektion wird in R32,33 die Startadresse fuer ROMs = 60000 Oct be-
7090 ! | reitgestellt. In R34,35 steht der Byte-Zaehler auf 8192 Dec. In R14
7100 ! | wird als Summe von BINTAB und ROMDATA ein unueblicher dekrementier-
7110 ! | ender Stack erzeugt. Fuer die Uebergabe der ROM Daten auf dem R12
7120 ! | Stack muessen die Daten in ROMDATA rueckwaerts angeordnet sein !!
7130 ! | Der Kern dieses Statements ist der Teil Loop, der lediglich 5 Zeilen
7140 ! | lang ist ! In der Kuerze liegt die Wuerze! Die Routine Loop liest
7150 ! | ein ROM mit 8192 Byte in nur 900 Millisekunden aus ! Das Auslesen
7160 ! | ueber FOR NEXT Schleifen mittels PEEK oder aehnlichem duerfte eini-
7170 ! | ge Minuten dauern.
7180 ! | Nach dem Auslesen, wird in R43-47 die Uebergabe fuer den R12 Stack
7190 ! | vorbereitet. Bevor man ins Betriebssystem mit RTN zurueckgeht, muss
7200 ! | man unbedingt wieder ROM 0 selektieren ! (Absturzgefahr !) Die Rou-
7210 ! | tine JSB=ROMJSB macht das automatisch. Hier wird ueber RSELEC aber
7220 ! | Handarbeit geleistet.
7230 ! |
7240 ! +-----+
7270          BYT 0,56        ! Attributes: String Funktion, 0 Parameter
7280 RMREAD.  BIN             ! CPU in Binary Mode
7290          LDMD R14,=BINTAB ! Startadresse fuer Binary laden
7300          LDBD R20,X14,ROM# ! Selectiertes ROM aus ROM# laden
7310          CMB R20,=377    ! Vergleich ob Default
7320          JNZ OK          ! JMP, wenn kein Default vorhanden
7330          LDB R20,=MYBP#  ! Eigene Binaerprogramm Nummer laden
7340          STBD R20,=ERRBP# ! In ERRBP# speichern, damit Error Message
7350          JSB =ERROR+     ! aus Binary kommt und Error Routine rufen
7360          BYT 245D        ! Error 'SELECT ROM FIRST'
7370 OK      STBD R20,=RSELEC ! ROM select ausfuehren

```

```

7380      LDM R32,=000,140      ! ROM Start 60000 Oct nach R32 laden
7390      LDM R34,=000,040      ! Zaehler binaer auf 8192 Dec setzten
7400      ADM R14,=ROMDATA      ! Rel. ROMDATA zu BINTAB addieren => Stack
7410 LOOP2  LDBD R20,R32        ! 1 Byte aus selektiertem ROM laden
7420      PUBD R20,-R14         ! Byte ueber decreasing Stack nach ROMDATA
7430      ICM R32                ! Incrementiere ROM Adresse
7440      DCM R34                ! Decrementiere Zaehler
7450      JNZ LOOP2              ! JMP zur Schleife, wenn Zaehler noch # 0
7460      LDMD R14,=BINTAB      ! Wieder Startadresse laden
7470      LDM R43,=0,40         ! Multibyte Operation: Stringlaenge 8192
7480      DEF ROMDATA           !                               Stringadresse
7490      BYT 0                  ! Byte 0 weil 2 Byte Adresse
7500      ADMD R45,=BINTAB      ! Add BINTAB zu ROMDATA. (absolut machen!)
7510      LDB R20,=0            ! Lade 0 fuer ROM 0
7520      STBD R20,=RSELEC      ! Wieder ROM 0 selektieren. Wichtig !
7530      LDB R20,=377          ! Lade 377 als Default
7540      STBD R20,X14,ROM#     ! Default in ROM# speichern
7550      PUMD R43,+R12         ! Auf den Uebergabe-Stack
7560      RTN                    ! Fertig. Zurueck zum System
7600 ! -----
7610 ROM#   BYT 377              ! Platzhalter fuer ROM Nummer
8000 ! +-----+
8010 ! |
8020 ! |          COPYR ist eine sehr simple Stringfunktion.
8030 ! |
8040 ! +-----+
8050      BYT 0,56              ! Attributes: String Funktion, 0 Parameter
8060 COPYR.  BIN                ! CPU in BIN Modus
8070      LDM R43,=80D,0        ! Stringlaenge laden
8080      DEF CPY                ! Labeladresse ( 2 Byte )
8090      BYT 0                  ! 3. Byte auf 0
8100      ADMD R45,=BINTAB      ! Offset (Start) Addieren
8110      PUMD R43,+R12         ! Auf den Uebergabe-Stack
8120      RTN                    ! Zurueck zum System
8200 ! -----
8210      ASC "6891 beF .02 :etaD 0.1 :.veR ffohretso.F & nnameleeS.I >=== erawtfoS-OFSI )c("
8220 CPY    BSZ 0                ! Platzhalter fuer Label
8300 ! -----
8310      BSZ 8192D             ! Memory fuer komplettes ROM reservieren
8320 ROMDATA BSZ 0              ! Platzhalter fuer Label
8500 ! -----
8510 BINTAB  DAD 104070
8520 CRTPOF  DAD 12334
8530 CRTPUP  DAD 12341
8540 ERBP#   DAD 103371
8550 ERROR+  DAD 10220
8560 NUMVA+  DAD 22403
8570 ONEX    DAD 56673
8580 OUTSTR  DAD 14020
8590 PTR2+   DAD 177716
8600 PTR2-   DAD 177715
8610 ROMEND  DAD 104145
8620 ROMTAB  DAD 104105
8630 RSMEM-  DAD 31741
8640 RSELEC  DAD 177430
8650 SCAN    DAD 21110
8660      FIN

```

Serie 70 Service

SORT

HP71, FORTH/ASS-Modul

Ich bin mittlerweile eininhalb Jahre Mitglied im CCD und habe mich bisher nur recht passiv und konsumierend verhalten ohne selbst Beiträge zu leisten, was ich hiermit ändern möchte.

So möchte ich meinen Einstieg in die Forth-Umgebung des 71 dazu nutzen, meinen ersten Beitrag zum Clubgeschehen zu leisten. Hier ein Programm, das sicher viele, die in Forth einsteigen wollen, interessieren wird. Es ist ein Sortierprogramm, das auf einem früheren Beitrag von Marcel Timborn, Prisma 83.9.07 basiert.

1.1 Programmname "SORT"

1.2 Aufgabe und Zweck:

Das Programm sortiert Floatingpointdaten, die in Blockform als Buffer, Forth-Variable, Basic-Vektor oder -Matrix vorliegen.

1.3 Gerätekonstellation:

Da es in Forth geschrieben ist setzt es natürlich das Vorhandensein eines Forth-Modules voraus. Ich benutze das Forth-Assembler-Modul, es müßte aber auch mit dem HP41-Translator-Modul lauffähig sein.

1.4 Arbeitsweise:

Das Programm arbeitet nach dem Verfahren von Shell/Metzner, praktisch nach dem glei-

chen Ablaufschema wie das Programm "S-M" (1144) aus obig genanntem Beitrag. Es wurde in die für Forth nötige strukturierte Form umgeändert und arbeitet direkt mit Adressen als Zeigern.

1.5 Programmlisting:

siehe Abbildung 1

1.6 Programmausführung:

Zu starten ist das Programm, nachdem es mittels "FTHSORT" LOADF geladen wurde, durch die Angabe von Anzahl und Anfangsadresse [Stackrelation: n addr -->] und Aufruf von SORT.

1.7 Beispiel:

Im Basic-Modus liegt ein Realfeld A (100) mit der Option Base 1 vor, also 100 Realwerte. Man gibt die Anzahl auf den Stack, die Adresse des ersten Elementes ermittelt eine kurze Unteroutine ADDRA(1). Durch aufrufen von SORT startet das Programm.

100 ADDRA(1) SORT (ret)

Es erscheint die Meldung "Sorting..." im Display gefolgt von der Forth-Meldung "OK", falls diese nicht unterdrückt wurde. Für 100 rein zufällig mittels RND erzeugte Werte benötigt es ca. 12 sec., 1000 Werte ca. 215 sec.

1.8 Einschränkungen:

Die Unteroutine ADDRA(1) ist nicht sehr elegant, vielleicht könnte jemand, der sich schon mehr mit dem Assembler befaßt hat, die Maschinenroutine ADDRSS für ein Primitive nut-

zen und das Sortierprogramm universeller gestalten bzw. ergänzen. ADDRA(1) ermittelt die Adresse des ersten Elementes des Feldes A(), es dürfen keine anderen Variablen (z.B. A1, A9\$) zuvor definiert worden sein, da dann deren Adresse bzw. die Werte selber als Adresse ermittelt würden. Die Adresse kann unabhängig vom Sortiervorgang ermittelt werden. Da sich diese aber ändern kann, sollte sie direkt vor dem Aufruf von SORT erfolgen.

1.9 Bemerkungen:

Abschließend die Laufzeiten im Vergleich Basic-Unterprogramm mit Parameterübergabe (siehe Abbildung 2) ca. 39 sec. HP41-Programm laut Angaben von Marcel Timborn (1144) ca. 310 sec. CCD-Modul-Funktion SORT, bemerkenswert schnell ca. 19 sec.

Die Darstellung des Programms habe ich an den Vorschlag von Horst Ziegler in Prisma 84.8.44 angelehnt, was sicherlich zur Verständlichkeit beiträgt.

Mit diesem Beitrag möchte ich meinen Einstand in die Clubaktivitäten beenden und mich der Forderung einiger Verfasser anschließen, all diejenigen, die bisher genauso passiv waren wie ich auch, zu mehr Initiative und Aktivität auffordern.

Abbildung 1:

FTHSORT TEXT 926

```

1:( FP-Datenblock Sortierung, Stackrelation: [n addr --> ] )
2:( fuer die umgekehrte Sortierung gross -> klein sind in den )
3:( Zeilen 19 und 25 die Vergleiche X>=Y? in X<=Y? zu aendern )
4:
5:VARIABLE NN ( Elementanzahl)
6:VARIABLE ZA ( Addr 1. Element)
7:VARIABLE NP ( Zeiger-P)
8:VARIABLE ZJ ( Zeiger-J)
9:FVARIABLE FC ( F-Speicher)
10:DECIMAL
11:
12:: ADDRA(1) 193984 @ 14 + DUP @ - DUP . ;
13:
14:: SORT ." Sorting ..."
15: ZA ! DUP 16 * NN ! 2/ 16 *
16: BEGIN
17:   DUP NP ! NEGATE ZA @ NN @ + + ZA @
18:   DO
19:     I DUP NP @ + RCL RCL X>=Y?
20:     IF
21:       I NP @ + STO X<>Y FC STO I
22:       BEGIN
23:         DUP ZJ ! NP @ - ZA @ 2DUP > ROT ROT = OR DUP
24:         IF
25:           DROP FC RCL ZJ @ NP @ - RCL X>=Y?
26:           THEN
27:           WHILE
28:             ZJ @ DUP STO NP @ -

```

```

29: REPEAT
30: FC RCL ZJ @ STO
31: THEN
32: 16 +LOOP
33: NP @ 32 / 16 * DUP DUP 0< SWAP 0= OR
34: UNTIL DROP ;
    
```

Abbildung 2:

zeigt das Programm als Basic-Unterprogramm, das mittels CALL SORT (Anzahl, Variablenname) aufzurufen ist. Es setzt Option Base 1 voraus. Ist Flag(0) gelöscht wird aufwaerts sortiert, bei gesetztem Flag(0) abwaerts. Eine Aenderung des Programmes kann eine Korrektur der POKE-Befehle, die die Vergleiche steuern, erforderlich machen.

```

SORT BASIC 351

100 SUB SORT(N,R())
110 INTEGER I,J,P @ REAL C
120 DIM A#[1],B#[1] @ A#='3' @ B#='1'
130 C=HTD(ADDR$('SORT'))
140 IF FLAG(0) THEN A#='6' @ B#='4'
150 POKE DTH$(C+445),A# @ POKE DTH$(C+579),B#
160 P=IP(N/2) @ DISP 'sorting ...'
170 FOR I=1 TO N-P
180 IF R(I)>R(I+P) THEN 230
190 C=R(I+P) @ R(I+P)=R(I) @ J=I
200 IF J-P<1 THEN 220
210 IF C>R(J-P) THEN R(J)=R(J-P) @ J=J-P @ GOTO 200
220 R(J)=C
230 NEXT I @ P=IP(P/2) @ IF P>0 THEN 170
240 PUT '#43' @ END SUB
    
```

Wolfgang Dittrich
Josephinenstr. 4
4630 Bochum

2 Tips für Videointerface-besitzer

1. Wird nachfolgende kurze Routine im Star-

tupstring mit STARTUP'CALL S' aufgerufen, so erscheint auf dem Monitor die augenblickliche Filebelegung des Hauptspeichers, einschließlich des verfügbaren freien Speicherplatzes.

```

START BASIC 122 12/11/86 19:51
-----
10 SUB S @ STD @ DISP CHR$(27);'E';
20 FOR I=1 TO INF @ A#=CAT$(I)[1,22]
30 IF A#='' THEN 50
40 DISP A# @ NEXT I
50 DISP 'Speicher:':MEM;'Bytes frei'
60 END SUB
    
```

2. Durch Belegung der OFF-Taste mit DEF KEY'#99','CHR\$(27);'E' @ OFF":

wird der Bildschirm beim Ausschalten automatisch gelöscht. Sollte dieses ausnahmsweise nicht erwünscht sein, kann man es durch Eintasten von OFF und betätigen von ENDLIN umgehen.

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

.END.

SORTXT

257 Bytes, 6 Zeilen
FORTH-Unterprogramm
FOSORT nötig

Die Möglichkeit, Daten, und hierunter besonders Textdateien, zu sortieren, ist wohl in vielen Anwendungsfällen ein Muß.

Besitzer des FORTH/Assembler- oder des EDITOR-ROMs wissen die Befehle INSERT#, DELETE# und REPLACE# zu schätzen, um mit Textfiles zu arbeiten. Da sie in Maschinensprache implementiert sind, bietet sich der Sortieralgorithmus des "Binären Einfügens" an, d.h. man fängt mit dem 2 Datensatz an und sortiert ihn in den schon sortierten Dateiteil ein, der am Anfang nur aus dem 1. Element besteht. Dann macht man dasselbe mit dem 3. Element usw. bis zum Dateieinde. Die Stelle, an der das einzusortierende Element einzufügen ist, wird nach der Methode des binären Suchens ermittelt. Dies geht am schnellsten in FORTH, obwohl die Datei (RAM!) immer wieder neu eröffnet werden muß, da ein und dieselbe Datei nicht zweifach geöffnet sein darf!

Für jedes einsortierte Element wird übrigens ein Punkt auf der Anzeige ausgegeben und zwar in Gruppen zu je zehn Punkten.

Viel Erfolg mit dem Programm!

SORTXT BASIC 257

```

10 INPUT "Text-File to sort: ";F$ @ INPUT "max. Rec.Length? ", "96";M @ DIM L$(M)
15 S=FILESZR(F$) @ IF S<0 THEN LIST F$
20 ASSIGN #1 TO F$ @ DELAY 0 @ DISP "SORTING" @ WINDOW 9 @ WIDTH 10
30 FOR R=1 TO S-1 @ READ #1,R;L$ @ ASSIGN #1 TO * @ FORTHX "INSPOS"
50 I=FORTHX @ ASSIGN #1 TO F$ @ IF I#R THEN DELETE #1,R @ INSERT #1,I;L$
60 NEXT R @ DISP @ DISP "finished" @ WINDOW 1 @ WIDTH 96 @ BEEP
    
```

FOSORT TEXT 336

```

( FORTH Module for SORTXT )
DECIMAL 96 STRING L$
: OPEN " F$" BASIC$ OPENF DROP ;
: GET-L$ " L$[1,96]" BASIC$ L$ S! ;
: MIDDLE 2DUP + 2/ DUP BLOCK DUP 2- 2- 4N@ ;
: INSPOS OPEN GET-L$ 1 " R+1" BASICI BEGIN
  2DUP <> WHILE MIDDLE L$ S<
    IF 1+ ROT DROP SWAP ELSE SWAP DROP THEN
  REPEAT DROP CLOSEALL FLUSH 46 EMIT 1- ;
    
```

Oliver Lohkamp (2622)
 Alte Bahnhofstraße 180a
 Postfach 700142

Change

HP71, ThinkJet, Edtext-File

Liebe Clubmitglieder,
 es ärgerte mich immer wieder, daß die deutschen Umlaute zwar über die Tastatur eingegeben werden konnten, der ThinkJet diese aber nicht ausdrückte.

Das folgende kleine Programm schafft nun Abhilfe. Bei der Arbeit mit diesem kleinen Pro-

gramm versuchte ich auch noch ein paar Druckersteuercode mit in den Textfile aufzunehmen. Das Escapezeichen kann jedoch nicht über die Tastatur in einen Textfile eingefügt werden.

Damit aber beim Brief sofort die SteuerCodes mit eingegeben werden können, gebe ich einen Platzhalter (z.B. ASCII 3 -g CTL C-) für ein Escape-Zeichen ein. Das Programm ändert diesen Platzhalter in das Escape-Zeichen. Beim Druck des Textfiles, ob durch

'READ#1; A\$@PRINTAS' oder durch PLIST..., werden diese SteuerCodes berücksichtigt.

Ich hoffe, daß einige von Euch diese Anregung gebrauchen können und freue mich auf Eure Reaktionen.

CHANGE BASIC 274 12/17/86 08:35 0.
 01

```

10 SUB CHANGE(#1)
30 DIM A$(120),S$(9)[1] @ INTEGER C(9),P1
40 RESTORE @ READ S$( ),C( )
50 FOR I=0 TO 9
250 P=-1
270 P=SEARCH(S$(I),0,P+1,9999,1)
290 IF NOT P THEN 360
310 READ #1,P;A$
330 P1=POS(A$,S$(I)) @ IF P1 THEN A$(P1,P1)=CHR$(C(I)) @ GOTO 330
350 REPLACE #1,P;A$ @ GOTO 270
360 NEXT I
370 DATA , , , , , , , , ,
390 DATA 10,13,27,204,206,207,222,216,218,219
410 END
    
```

Zeile 370

```

DATA chr$(1),chr$(2),chr$(3),chr$(22),chr$(24),
chr$(26),chr$(5),chr$(21),chr$(23),chr$(25)
    
```

Klaus Lensing
 Heideweg 3
 4558 Bersenbrück

MATRIX

FORTH

Manchmal ist es zweckmäßig, Matrizen in BASIC und FORTH zu bearbeiten. Hier meine Lösung des Problems (stets mit OPTION BASE 1 und REAL).

Es ist sinnvoll, Matrizen in FORTH im selben Format zu definieren wie dies in BASIC automatisch geschieht. Das FORTH-Wort MATRIX tut dies. Man schreibt z.B. 10 8 MATRIX A. Dies definiert eine 10x8 Matrix A im Wörterbuch. Wenn A eingegeben wird, kommen die Parameterfeldadresse von A, die Adresse des ersten Elements von A, die Anzahl der Zeilen und die Anzahl der Spalten auf den Stack. Tippt man A 5 6 MENTRY, so erhält man die Parameterfeldadresse von A und die Adresse des Elements A(5,6), das mit RCL abgerufen oder mit STO neu belegt werden kann. Mittels der Parameterfeldadresse und NAME kann zudem der Name der Matrix wiedergegeben werden: Man tippt etwa A DROP

DROP DROP NAME TYPE und erhält A in der Anzeige.

Hat man schon eine Matrix in BASIC vorliegen, die beispielsweise M heißt und in FORTH unter dem Namen N bearbeitet werden soll, so definiert man N zunächst in FORTH (oder mittels FORTH) als leere Matrix indem man schreibt: 0 0 MATRIX N. Von BASIC-Matrix M von FORTH aus unter dem Namen N genauso zugänglich wie wenn sie in FORTH definiert wäre. Sie ist aber außerhalb des FORTH-Wörterbuchs abgespeichert. Deshalb nach GROW oder SHRINK wieder MPASS ausführen, sonst kann es Überraschungen geben!

Spezialoperationen können nun mittels geeigneter FORTH-Wörter und FORTHX (z.B. FORTHX ' N TRANSPOSE', was entsprechend zu dedinieren wäre) schnell und bequem durchgeführt werden. Zugleich kann man die Matrixroutinen des HP 71 und ggf. des Mathematikmoduls von FORTH ausnutzen.

Berichtigung zu Prisma 7/86 (Sonnenephemeride)

Wie schon von Willi Jeschke in Prisma 86.08.12 richtig erkannt, hat mein Programm "SONNENE1" einen kleinen Fehler. Leider bin ich im Augenblick Wehrpflichtableistender, so daß die nötige Korrektur etwas später kommt. Den Weg aus Prisma 8/86 finde ich wenig sinnvoll, da so die ekliptikale Breite in Grad (und damit mit einigen Nullen nach dem Komma) und nicht in Bogensekunden ausgedruckt wird. In Zeile 640 ist aber trotzdem noch ein Fehler. Richtig muß sie folgendermaßen heißen: 640 B=-.021x... Zeile 800 muß dann noch in 800 CALL ETOA (I,L,B/3600,A,D) umgeändert werden.

Stephan Trzeciak (1627)

.END.

```
DECIMAL
: NAME ( PFA --> addr, length )
  7 - DUP -1 TRAVERSE 2+ DUP 2- ROT - 2 / -1 * ;
: MATRIX ( n, m --> )
  CREATE OVER OVER * 16 * 15 + NALLLOT
  LATEST 1 TRAVERSE 7 + DUP 15 + OVER !
  5+ ROT OVER ! 5+ !
  DOES> ( --> PFA, addr, n, m )
  DUP DUP 5+ DUP 5+ @ ROT @ ROT @ ROT ;
: MENTRY ( PFA, addr, n, m, i, j --> PFA, addr )
  1- SWAP 1- ROT * + SWAP DROP 16 * + ;
: PASS ( addr, n, m, PFA, #, #, # --> )
  DROP 2DROP 4 ROLL OVER ! 5+ ROT OVER ! 5+ ! ;
```

```
10 SUB MPASS(A( , ),N$)
20 I=0 @ ON ERROR GOTO 'L2'
30 'L1': I=I+1 @ DISP A(I,1) @ GOTO 'L1'
40 'L2': M=I-1 @ I=0 @ ON ERROR GOTO 'L4'
50 'L3': I=I+1 @ DISP A(1,I) @ GOTO 'L3'
60 'L4': N=I-1
70 FORTHX " @ 5+ @ 11 + DUP @ - ",HTD("ZF5C0")
80 FORTHX " ",M,N
90 FORTHX N$&" PASS " @ END SUB
```

P.S.: Mit Mathematikmodul können die Zeilen 20 bis 60 durch die Zeilen

```
20 M=UBND(A,1)
30 N=UBND(A,2)
```

ersetzt werden.

Ekkehart Schlicht (2943)
Untergasse 8
6144 Zwingenberg

GLSYS

Lösung linearer Gleichungssysteme

Folgendes kleine Programm dient zur Lösung eines linearen Gleichungssystems mit n-Gleichungen und n-Unbekannten. Benutzt wird der verkettete Algorithmus nach Gauss. Das Programm ist in sehr kurzer Zeit entstanden und kann bestimmt noch verbessert werden. Wer hat dafür noch gute Ideen, bzw. wer kennt bessere (schnellere) Algorithmen? Es gibt doch bestimmt noch andere HP-71-Besitzer in unserem Club, die sich mit diesem Problem beschäftigen oder beschäftigt haben.

```
Beispiel:
x+y+z+u+v+w=3
x-y+z-u+v-w=-3
x-y-z-u-v+w=3
x-y-z+u+v+2w=-4
3x+2y-z-u-v+w=16
27x-5y-6z-3u+v-11w=22
```

Lösung:
x=2; y=3; z=1; u=-1; v=-3; w=1

Die Koeffizienten und Ergebnisse werden von selbst abgefragt. Dabei ist A(1,1) in der ersten Zeile das erste Element. Also der Koeffizient von x in der ersten Zeile. E(2) ist dann das Ergebnis der zweiten Zeile. Die Lösungen werden solange angezeigt, bis eine Taste gedrückt wird. Erst dann wird die nächste Lösung angezeigt.

GLSYS

```

10 INPUT "Anzahl d. Gl.= ";N
20 DIM A(N,N),E(N)
30 FOR I=1 TO N
40 FOR J=1 TO N @ DISP "A(";I;",";J;")= "; @ INPUT " ";A(I,J) @ NEXT J
50 DISP "E(";STR$(I);")= "; @ INPUT " ";E(I)
60 NEXT I
70 FOR I=1 TO N
80 FOR J=I TO N
90 FOR L=1 TO I-1 @ A(I,J)=A(I,J)+A(I,L)*A(L,J) @ NEXT L
100 NEXT J
110 FOR K=I+1 TO N
120 FOR L=1 TO I-1 @ A(K,I)=A(K,I)+A(K,L)*A(L,I) @ NEXT L
130 A(K,I)=A(K,I)/-A(I,I)
140 NEXT K
150 FOR J=1 TO I-1 @ E(I)=E(I)+A(I,J)*E(J) @ NEXT J
160 NEXT I
170 FOR I=N TO 1 STEP -1
180 FOR J=N TO I+1 STEP -1 @ E(I)=E(I)-A(I,J)*E(J) @ NEXT J
190 E(I)=E(I)/A(I,I)
200 NEXT I
210 FOR I=1 TO N
220 DISP "x(";I;")= ";E(I)
230 A$=KEY$ @ IF A$="" THEN 230
240 NEXT I
    
```

Stephan Trzeciak (1627)

Grafik

3840 Bytes, FORTH
Beispielprogramm Qaditten (Basic)

Liebe Clubfreunde,
anbei habe ich ein Programm, es hat ein Weil-

chen gereift und ist nun wohl ganz brauchbar.
Sollte das Programm gedruckt werden, muß
bei den FORTH_Worten genau auf die Zwi-
schenräume geachtet werden, sonst gibt es
bei der Anwendung Schwierigkeiten.

FLINTXT - Grafik mit Forthmodul und Think-
Jet
Welcher Druckerbesitzer hatte nicht schon
einmal den Wunsch, für Tabellen usw. ein-
fach und schnell Linien drucken zu können.

Anlage: TEST												20.06.86			
Dat	As	Flwaf	Schw	Dat	As	Flwaf	Schw	Dat	As	Flwaf	Schw	Dat	As	Flwaf	Schw
1	7,1	27,08	1,12	9	7,3	26,87	1,12	17	7,2	26,78	1,15	25	0,0	0,00	0,00
2	7,4	27,15	1,17	10	7,4	26,75	1,13	18	7,6	26,69	1,13	26	0,0	0,00	0,00
3	7,5	26,97	1,21	11	7,0	26,99	1,09	19	7,3	26,90	1,15	27	0,0	0,00	0,00
4	7,2	26,88	1,16	12	7,5	26,95	1,18	20	0,0	0,00	0,00	28	0,0	0,00	0,00
5	7,3	27,19	1,15	13	7,2	26,56	1,18	21	0,0	0,00	0,00	29	0,0	0,00	0,00
6	7,4	26,86	1,13	14	7,2	26,77	1,14	22	0,0	0,00	0,00	30	0,0	0,00	0,00
7	6,6	26,90	1,11	15	7,1	26,48	1,16	23	0,0	0,00	0,00	31	0,0	0,00	0,00
8	7,1	27,00	1,11	16	7,2	26,72	1,16	24	0,0	0,00	0,00	32	0,0	0,00	0,00

Das Programm FLINTXT bietet eine spei-
cherplatzsparende, laufzeitoptimiert und
benutzerfreundliche Lösung. (Darum sieht
das Programm auch so kompliziert aus). Die
Befehle können sowohl in Forth als auch in
Basic-Umgebung aufgerufen werden, eben-
so können die Steuerzeichen leicht an andere
Drucker angepaßt werden..

User-Befehle

Es können die Punkte 0-639 angesprochen
werden, das entsprechende Bit wird in der
Variable BILD gesetzt. Außerhalb dieses Be-
reiches liegende Punkte werden gleich 0 oder
639 gesetzt. Die Rechnervoreinstellung ist
PWIDTH INF.

LOESCHE setzt BILD auf
80*CHR\$(0)
SETZPUNKT (X-) nimmt x vom TOS (= Top of Stack) und setzt

LOESCHPUNKT
(X1 X2 Xn -)

LOESCHPUNKTE
(X1..Xn)

DRUCK
VIELDRUCK (n-)
LINIE (X1 X2-)

LOESCHLINIE
(X1 - X2)

Der Zugriff auf die FORTH-Befehle aus der
Basic-Umgebung ist über den Befehl
FORTHX vorgesehen, z.B. 10 FORTHX "
LOESCHE 1 40 LINIE " oder 20 FORTHX "
SETZPUNKTE ",0,11,12,639. Dieser Basic-

das entsprechende Bit
in BILD

nimmt Zahlen vom
Stack, bis er leer ist!
und setzt die ent-
sprechenden Bits
in BILD

Wie zuvor, löscht je-
doch die entsprechen-
den Bits in BILD

Druckt BILD als Bitlinie
Druckt BILD n-mal
Setzt Bits in BILD
von X1 bis X2

Löscht Bits in Bild
von X1 bis X2

Befehl sendet zuerst die Parameterliste - hier
0, 11, 12, 639 auf den Stack (639 liegt auf dem
TOS) und führt dann die Befehle in den An-
führungszeichen aus. Die Parameterliste darf
maximal 14 Elemente enthalten.

Bild 2 zeigt das Programm für Bild 1, die Fel-
der A, F und S werden ab Zeile 140 in Tabel-
lenform ausgegeben. Wird Text ohne Vor-
schubzeichen gedruckt, überschreibt die
Grafik den Text. Weitere Fragen beantwortet
hoffentlich das Listing.

Die Struktur des Forthprogrammes ist relativ
aufwendig. Wer jedoch einmal versucht, eine
Linie nur mit dem Befehl SETZPUNKT zu er-
zeugen oder gar die Folge 800 1 SYSLINIE
ausprobiert, weiß das danach zu schätzen,
denn ersteres dauert auch in FORTH recht
lange und zweiteres führt zum Systemab-
sturz.

Der Programmkopf enthält die Variablendef-
initionen. Die Strings GRAFIKAN ZEICHEN-

ZEILE und GRAFIKAUS enthalten die entsprechenden ASCII Zeichen zur Druckersteuerung. Die Zahl hinter DEFLNG gibt die aktuelle Länge in den String ein, die danach folgenden Zahlen sind die dezimalen Äquivalente der ASCII Zeichen. Der Befehl DRUCK sendet das alles an den Drucker. Die BEFehle

SCHWARZ und WEISS informieren die SYS-Befehle, ob Punkte gesetzt oder gelöscht werden sollen. Die Kontrolle der Eingangswerte übernehmen die Befehle UNTERGRENZE bis PRUEFX. SYSLINIE zerlegt die in BILD zu setzenden Bitmuster so, daß das "angeknabberte" erste und letzte Byte in

BILD eingelesen werden und dann die dazwischen liegenden vollen Bytes über FILL eingelesen werden. Das geht wesentlich schneller als über SETZPUNKT.
Das wärs, noch Fragen?

FLINTXT

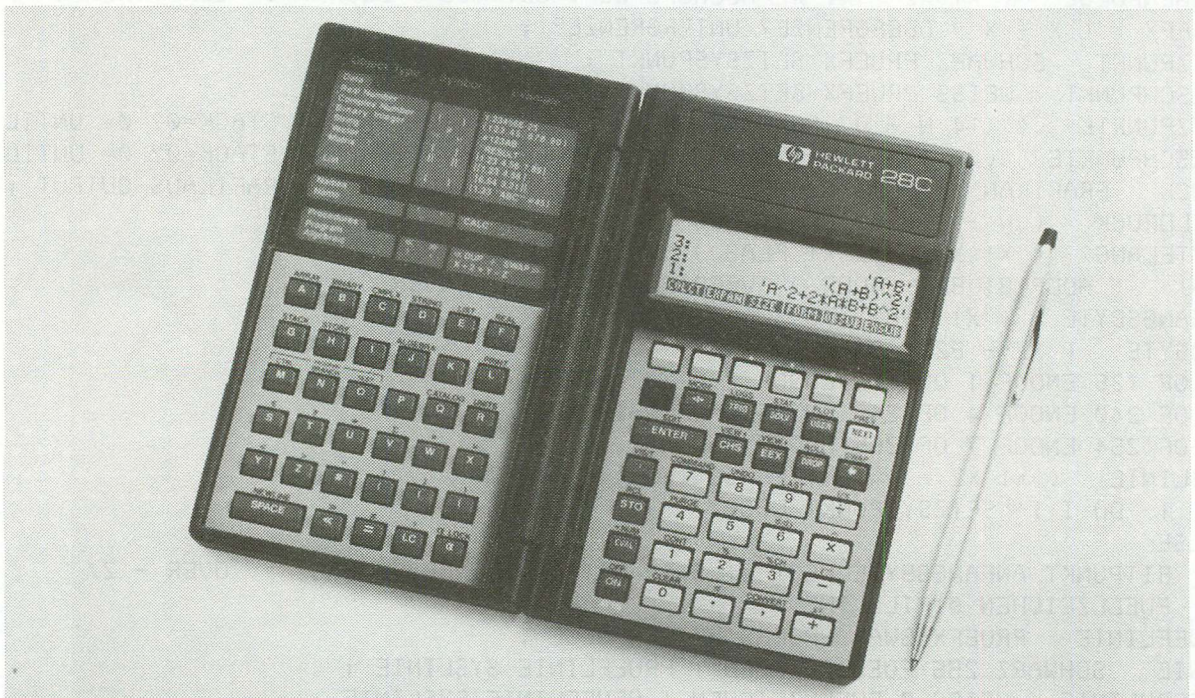
07.07.86

```
( Liniengrafik 19.06.86 von Reinhard Schmidt, 2053 )
( Chemnitzer Str. 111, 4600 Dortmund 1 )
0 CONSTANT UNTERGRENZE
639 CONSTANT OBERGRENZE
VARIABLE BILD 155 NALLOT VARIABLE INCR
VARIABLE 'VERBINDE VARIABLE FUELLZEICHEN
7 STRING GRAFIKAN
6 STRING ZEICHENZEILE
4 STRING GRAFIKAUS
: & SWAP C! 2+ DUP ;
: DEFLNG ( N ADDR - ADDR ADDR-2 ) DROP DUP 2- ;
GRAFIKAN DEFLNG 7 SWAP C! DUP 27 & 42 & 114 & 54 & 48 & 48 SWAP C! 2+ 83 SWAP C!
ZEICHENZEILE DEFLNG 6 SWAP C! DUP 27 & 42 & 98 & 56 & 43 SWAP C! 2+ 87 SWAP C!
GRAFIKAUS DEFLNG 4 SWAP C! DUP 27 & 42 & 114 SWAP C! 2+ 66 SWAP C!
: LOESCHE BILD 80 0 FILL ;
: IOR ( X1 X2 - X ) 255 XOR AND ;
: SCHWARZ ['] OR 'VERBINDE ! ;
: WEISS ['] IOR 'VERBINDE ! ;
: 2^N 2. ITOF Y^X FTOI ;
: BIT 7 SWAP - 2^N ;
: BITPUNKT ( X - ADDR B X ) 8 /MOD 2* BILD + DUP C@ ROT ;
: SETZSYSPUNKT ( X - ) BITPUNKT BIT 'VERBINDE @ EXECUTE SWAP C! ;
: STACK=0? S0 SP@ - 5- ( bei leerem Stack n=0 sonst n>0 ) ;
: UNTERGRENZE? ( X - X oder 0 ) UNTERGRENZE MAX ;
: OBERGRENZE? ( X - X oder 639 ) OBERGRENZE MIN ;
: REIHENFOLGE ( X1 X2 - X1 X2 hoehere Wert auf TOS ) 2DUP > IF SWAP THEN ;
: PRUEFX ( X - X ) OBERGRENZE? UNTERGRENZE? ;
: SETZPUNKT SCHWARZ PRUEFX SETZSYSPUNKT ;
: LOESCHPUNKT WEISS PRUEFX SETZSYSPUNKT ;
: SETZPUNKTE ( <14 N - ) SCHWARZ BEGIN PRUEFX SETZSYSPUNKT STACK=0? 0= UNTIL
: LOESCHPUNKTE ( <14 N - ) WEISS BEGIN PRUEFX SETZSYSPUNKT STACK=0? 0= UNTIL
: DRUCK GRAFIKAN OUTPUT ZEICHENZEILE OUTPUT BILD 80 OUTPUT GRAFIKAUS OUTPUT ;
: VIELDRUCK ( N - ) 0 DO DRUCK LOOP ;
: 1BYTELANG? ( X1 X2 - X2 X1 FLAG ) SWAP 2DUP 8 / SWAP 8 / = ;
: WEG! ( ADDR B1 B2 - ADDR ) 'VERBINDE @ EXECUTE OVER C! ;
: ANFANGSBYTE ( X1 - B1 ) 8 SWAP - 2^N 1- ;
: ENDBYTE ( X2 - B2 ) CASE
  0 OF 128 ENDOF 1 OF 192 ENDOF 2 OF 224 ENDOF
  3 OF 240 ENDOF 4 OF 248 ENDOF 5 OF 252 ENDOF
  6 OF 254 ENDOF 7 OF 255 ENDOF ENDCASE ;
: SYSLINIE ( X1 X2 - ) 1BYTELANG? IF
  1- DO I 1+ SETZSYSPUNKT LOOP
ELSE
  BITPUNKT ANFANGSBYTE WEG! 2+ SWAP BITPUNKT ENDBYTE WEG! OVER - 2/
  FUELLZEICHEN @ FILL THEN ;
: PRUEFLINIE PRUEFX SWAP PRUEFX REIHENFOLGE ;
: LINIE SCHWARZ 255 FUELLZEICHEN ! PRUEFLINIE SYSLINIE ;
: LOESCHLINIE WEISS 0 FUELLZEICHEN ! PRUEFLINIE SYSLINIE ;
```

```

10 ! addiert nach Datum sortierte Qalitten
20 DESTROY ALL
30 DIM T(32),A(32),F(32),S(32),W(32)
40 INTEGER D
50 ON ERROR GOTO 'ERROR'
60 'START': INPUT 'Dat,Tn: ';D,T0
70 INPUT 'Ws,As: ';W0,A0
80 INPUT 'F1,S: ';F0,S0
90 T9=T(D) @ A9=A(D) @ F9=F(D) @ S9=S(D) @ W9=W(D)
100 CALL PRODPLUS(T0,W0,A0,F0,S0,T9,W9,A9,F9,S9,T(D),W(D),A(D),F(D),S(D))
110 GOTO 'START'
120 'ERROR': E=ERRN
130 IF E=40 AND D=0 THEN 'AUS' ELSE IF E=40 THEN BEEP @ GOTO 'START' ELSE DISP E
140 'AUS': INPUT 'An1: ';A$
150 PWIDTH INF
160 FORTHX " @ 639 LINIE 2 VIELDRUCK"
170 FORTHX " LOESCHE 0 1 638 639 SETZPUNKTE 4 VIELDRUCK "
180 PRINT USING "#,' Anlage: ',k,B,71X,K";A$,13,DATUM$
190 FORTHX " 14 VIELDRUCK "
200 FORTHX " @ 639 LINIE DRUCK"
210 FORTHX " LOESCHE SETZPUNKTE 2 VIELDRUCK",0,1,155,315,475,638,639
220 PRINT USING "#,4(' Dat As Flwaf Schw ')"
230 FORTHX " 14 VIELDRUCK "
240 FORTHX " @ 639 LINIE 2 VIELDRUCK"
250 FORTHX " LOESCHE SETZPUNKTE",0,155,315,475,639,638,1
260 FORTHX " 4 VIELDRUCK"
270 FOR I=1 TO 8
280 FOR J=0 TO 3
290 D=I+J*8
300 PRINT USING 310;D,A(D),F(D),S(D)
310 IMAGE #,DDX,DZROX,2(DZRODX)
320 NEXT J
330 FORTHX " VIELDRUCK",14
340 NEXT I
350 FORTHX " @ 639 LINIE 2 VIELDRUCK"
360 PRINT @ PRINT
    
```

Reinhard Schmidt (2053)
 Chemnitzer Str. 111
 4600 Dortmund 1



Der HP-28C hat ein Display mit 4 Zeilen und 23 Zeichen. In der untersten Zeile werden die Softkeys angezeigt. Der Rechner ist zusammenklappbar. Die Alphatastenseite kann man vollständig umklappen und so hinter dem Rechner verschwinden lassen.



**HEWLETT
PACKARD**

Das Leistungspaket

HP Vectra Mod. 50

20 MB · S/P-Schnittstelle
Monochrom-Monitor/
Herkuleskompatibel

DM 10.500,-

HP LaserJet Plus

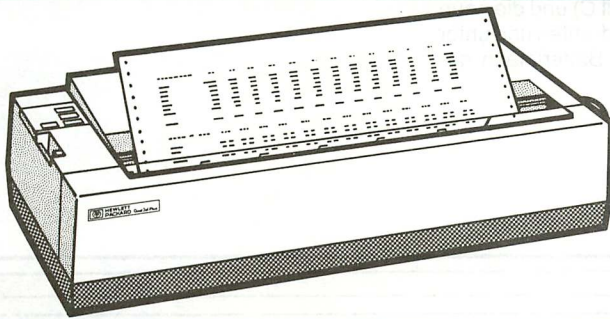
incl. Cartridge Q

DM 6.998,-

HP ColorPro

- 8 Farben

DM 2.698,-



HP QuietJet DM 1.198,-

**HP QuietJet Plus
DM 1.698,-**

HP 18 CD Business Consultant

DM 373,-

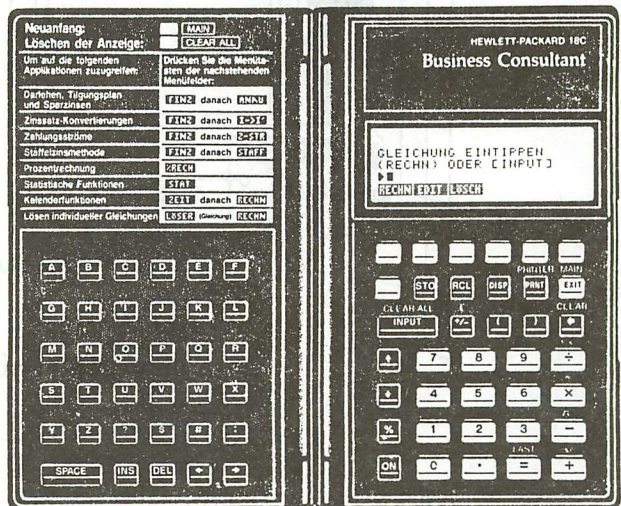
HP 28 CD

mathematisch/
wissenschaftlicher Rechner

DM 499,-

**HP Infrarot-Taschenrechner
Drucker**

DM 285,-



Alle Preise sind incl. Mehrwertsteuer und gelten für CCD-Mitglieder! Versand per UPS – Nachname zuzügl. Versandkosten.

Sind Sie an weiteren HP-Produkten interessiert, so lassen Sie uns das bitte wissen!

Name:

Straße:

Wohnort:

CCD-Nr.:

Taschenrechner-Preisliste

Personal-Computer-Preisliste

GES - COMPUTER

GESELLSCHAFT FÜR EDV UND SOFTWARE mbH

Hartmann-Ibach-Str. 63
6000 Frankfurt 60
Telefon: (069) 462041

Steinheimer Str. 22
6450 Hanau
Telefon: (061 81) 24826

Serie 10 Service

HP-15C Speed-up

Wie alle anderen Taschencomputer von Hewlett Packard, so benutzt auch die CPU des HP 15 einen externen Schwingkreis (LC-Clock) für seine Taktfrequenzgenerierung. Lediglich die Hardwareausführung weicht bei der 10er Serie von früheren Geräten insofern ab, daß hier RAM-ROM- und CPU-Bausteine in hochintegrierter Form eingebaut sind und somit nur noch der LC-Kreis in diskreter Bauweise ausgeführt ist. Weitere Parallelen zu HP 41/HP 67 findet man bei der Tastenmatrixcodierung und bei den Steuer- und Adreßleitungen.

- 1) Vorbereitungen: Um Beschädigungen durch statische Ladungen auszuschließen, sollte man sämtliche Werkzeuge, die benötigt werden (inkl. arbeitende Person), durch Masseverbindung (evtl. an die nächste Wasserleitung) auf Null-Potential legen.
- 2) Öffnen des Gerätes: Unter den Gummfü-

ßen des Rechners (vorsichtig mit spitzem Gegenstand abziehen) sind 4 Kreuzschlitzschrauben; **nachdem** die Batterien des Rechners entfernt wurden, werden diese herausgedreht und der Deckel abgenommen.

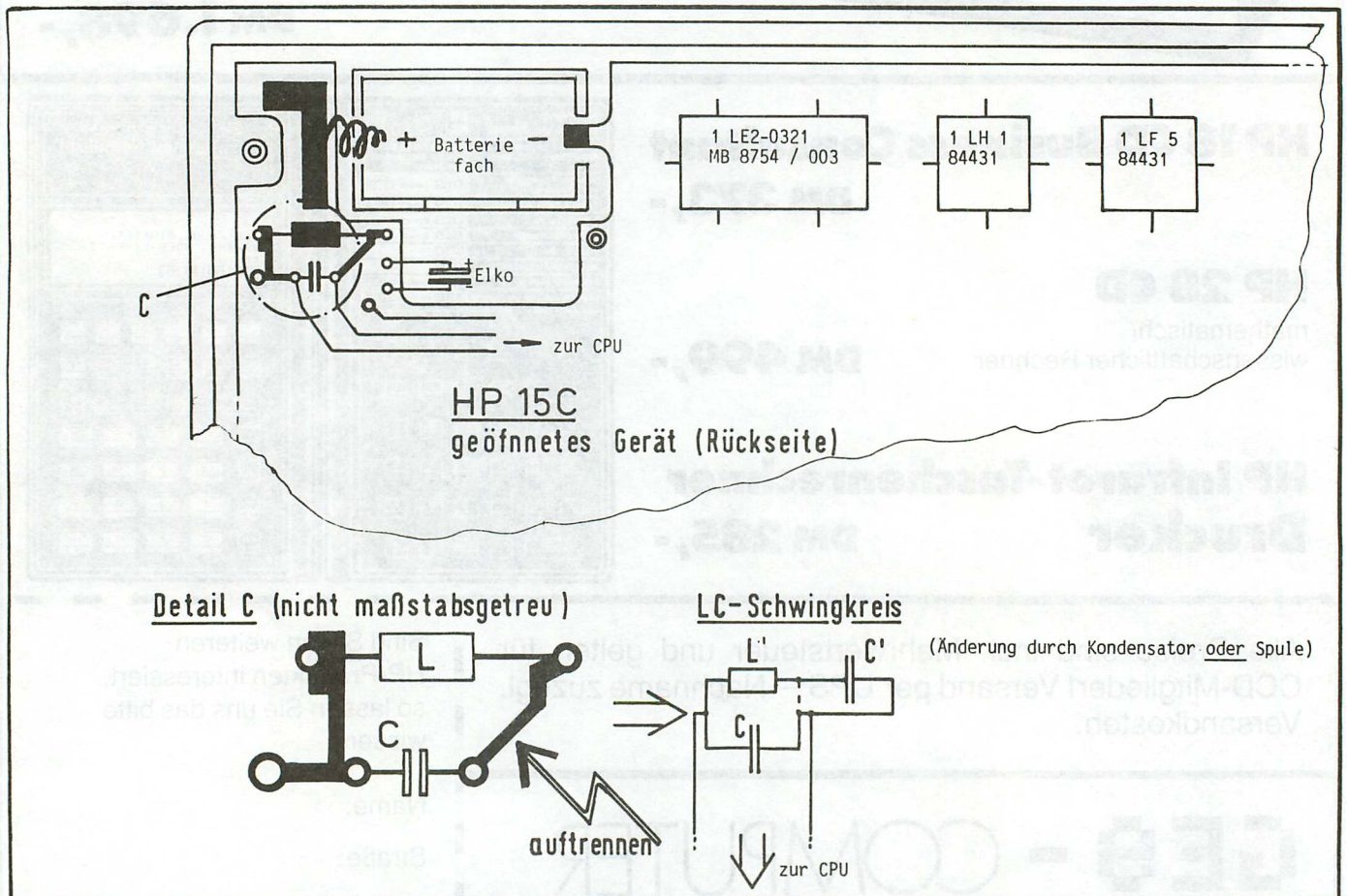
Die Platine des Rechners kann nicht aus dem Gerät entfernt werden. Zu erkennen sind drei hochintegrierte Chips (RAM ROM CPU) auf der Platinenrückseite, der LC-Schwingkreis auf der Platinenvorderseite (siehe Bild: links unter dem Batteriefach) und ein Elektrolytkondensator (22 uF/25V) in der Platinausparung unter dem Batteriefach. Von dem Schwingkreis sind bis auf die eingelöteten Anschlüsse nur die Spule visuell zu erkennen.

- 3) Taktfrequenzänderung: Hier bietet es sich an, statt dem Kondensator die Spule zu ändern, d.h. sie durch eine Spule L' zu ersetzen. Am einfachsten ist es, die Leiterbahn zur Spule aufzutrennen (siehe Detail C) und die neue Spule (verbunden mit Fädeldrähten und unter die Platine links über dem Batteriefach ge-

steckt) einzulöten. Die neue Dimension der Spule soll dabei noch so gewählt werden, daß die Anzeige des Rechners (kritisches Element des Rechners) auch bei Temperaturen von 35° Celsius (Temperatur des Rechners) noch einwandfrei arbeitet. Wird die Taktfrequenz zu weit überschritten (größer 2,5fach), so zeigen die ersten Stellen der Mantisse keine weiteren Zeichen an, bei weiterer Überhöhung werden Anweisungen von der Tastatur nicht mehr ausgeführt.

Richtwerte siehe Zeichnung - (C) Mai 85 1

Jürgen Friebe
Herzogstr. 32
8000 München 40



RAM STORAGE UNIT TEST

Nach 32kB jetzt auch 64kB Speichererweiterung für HP41 verfügbar

Speichererweiterungen (eng. Rambox) bieten den unschätzbaren Vorteil, sämtliche Ports des HP 41 als Direktzugriffsspeicher für persönliche Anwendungen zu nutzen. Auch nach einem MEMORY LOST bleiben alle Daten in der Erweiterung erhalten. Programme können in eigene "Module" geladen werden und sind lauffähig wie Funktionen eines Einsteckmoduls. Somit können auch Projekte verwirklicht werden, die sonst im Hauptspeicher nicht unterzubringen wären.

Kürzlich hatte ich die Möglichkeit, die neuer-schienenene 64kB Speichererweiterung der Firma ERAMCO SYSTEMS (RAM STORAGE UNIT) zu beschnuppern. Jeder von uns kennt ja wohl die o.g. Firma aus Holland, die schon seit geraumer Zeit das ERAMCO-MLDL (Machine Language Development Laboratory) in Ihrem Angebot haben. Von der Firma W&W Software Products GmbH in Bergisch Gladbach wird seit Anfang dieses Jahres ebenfalls eine Speichererweiterung (32kB W&W RAM-BOX) angeboten. Um die obrige Auflistung noch zu vervollständigen, sei noch das MBK-ProfiSET des Mathematischen Büro in Köln erwähnt. Das letzte aber ist unter den Anwendern von Ramerweiterungen weniger verbreitet.

Auf den ersten Blick ist die RAM STORAGE UNIT kaum von der Rambox der Firma W&W zu unterscheiden. Beide Einheiten sind im Kartenlesergehäuse untergebracht. Somit ist wieder der Vorteil der Portabilität des HP-41 Systems gewährleistet.

Da ich bis jetzt nur mit der W&W Rambox gearbeitet habe, möchte ich gegen Ende des Artikels einen Vergleich zwischen beiden Systemen ziehen. Nun soll aber eine kurze Beschreibung der neuen 64kB Speichererweiterung folgen.

Zum Zeitpunkt des Tests stand mir noch ein Prototyp zur Verfügung, der sich aber laut Hersteller nur geringfügig von der Endversion unterscheidet.

Auf der Vorderseite des Kartenlesergehäuses befinden sich in der Vertiefung auf der linken Seite 6 kleine Dip-Schalter mit folgender Bedeutung: Die 4kB-Blöcke (Pages) werden nicht einzeln an/abgeschaltet, sondern mit den Schaltern 1-4 portweise als 8kB-Blöcke. Der Schalter 5 setzt die gesamte Einheit (ein 32kB-Block) überschreibgeschützt, wobei das Betriebssystem, das bei Auslieferung auf Port 4 gelegt ist, von vornherein gegen unbeabsichtigtes Löschen geschützt ist. Schalter 6 ermöglicht den Zugang zum zweiten 32kB-Block. Als Anmerkung sei erwähnt, daß der HP-41 auf Grund seiner Hardwarekonzeption nur eine zusätzliche Speichererweiterung vom max. 32kB adressieren kann. Daher

mußte in der 64kB RSU die Methode des Bankswitching per Schalter gewählt werden. Wo herkömmliche Speicher am Ende ihrer Kfapazität sind, bietet nun die neue Erweiterung ein doppeltes Speicherangebot.

Die RAM STORAGE UNIT (im folgenden RSU genannt) wird standardmäßig mit dem ES-41 DATABASE Betriebssystem ausgeliefert. Bei diesem Betriebssystem handelt es sich um ein 8kB-System mit insgesamt 116 Funktionen. Gewählt wurden die XROM-Nummern 4 und 6.

Leider ist es nun im Rahmen des Testberichtes nicht möglich, auf alle Funktionen im einzelnen einzugehen. Daher werde ich mich auf die wichtigsten auffallenden Neuigkeiten konzentrieren.

Das Handbuch mit zwei Kurzanleitungen in leichtverständlichem Englisch wird in einem Ringbinder geliefert und umfaßt ca. 360 Seiten, die detaillierte Informationen bezüglich der Funktionen des Betriebssystems beinhalten, sowie 8 Anwendungsprogramme, die auf die Funktionen zurückgreifen, einschließlich Barcodes.

Im ersten Teil des Handbuches wird auf alle Funktionen im einzelnen eingegangen. Es finden sich auch nahezu alle Funktionen, die in bereits vorhandenen Betriebssystemen vorhanden sind, darunter z.B. READRAM/WRITERAM (kompatibel zu GET-/SAVEROM des MLDL bzw. zu READPG/WRITEPG der W&W Rambox).

Wichtige zusätzliche Funktionen sind zu nennen:

- DELFL: Löschen eines beliebigen Files an beliebiger Stelle
- RENFL: Umbrennen eines beliebigen Files
- CLRSU: Löschen eines 32kB-Blockes außer dem Betriebssystem
- INIDATA/INIPRGM: Initialisierung eines 4kB-Daten/Programmblockes

Anmerkung: RSU unterscheidet streng zwischen Programm- und Datenblöcken. Es werden viele Datenfile/Datenregister/Zeigeroperationen angeboten, die eigentlich den Schwerpunkt des Moduls ausmachen. Noch nie war Datenregisterarithmetik, Bewegungen innerhalb des Datenregister, Vergleiche von Datenfiles so einfach geworden.

- EXTENDED MEMORY FUNKTIONEN:
CLEM Löschen des ges. X-Memory Bereiches
EXEM Austauschen des ges. X-Mem. mit File in RSU
RCLEM/STOEM Laden und Speichern des EM in RSU
- MAIN MEMORY FUNKTIONEN:
EXALL, RCLALL, STOALL Transfer zwischen Hauptspeicher u. RSU

Ein zweiter Schwerpunkt des DATABASE-Betriebssystems ist die Behandlung von ASCII Files. Ein besonderer Editor (EDT) wird zur Verfügung gestellt. Der Befehlsumfang ist ähnlich wie bei den Datenfiles. Leider kann ich nicht im Rahmen dieses Berichtes darauf eingehen. Das Angebot an so vielen neuen Funktionen erfordert mehrere Wochen Einarbeitungszeit.

Nun aber der langersehnte Vergleich mit der 32kB W&W Rambox:

Vom Speicherumfang ist natürlich die RSU der 32kB W&W Rambox überlegen, jedoch sind Einschränkungen zu machen. In jedem 32-kB Block der RSU muß ein Betriebssystem zur Verwaltung der 4kB-Blöcke vorhanden sein, wodurch sich der verfügbare Speicher von 64kB auf effektiv 48kB verringert. So kann man evtl. im ersten 32kB-Ramblock mit dem Database Rom arbeiten und in der zweiten mit dem MLDL-Betriebssystem. Leider vermisste ich auch in dem umfangreichen Database Rom einen Befehl wie COPYPG (COPYROM) zum Austauschen von 4kB-Blöcken. Positiv muß ich erwähnen, daß die DIP-Schalter nun gut zugänglich sind, auch wenn das IL-Modul eingesteckt ist. Bei der 32kB W&W Rambox wurden auf Grund des optischen Designs die Schalterreihen in die Aussparung des Kartenlesergehäuses gelegt. Da man bei der RSU die 4kB-Blöcke nicht per DIP-Schalter einzeln abschalten kann (s.o.), muß man bei der Vergabe der XROM-Nummern besondere Aufmerksamkeit walten lassen. Besitzen nämlich zwei Pages in einem Port die gleiche XROM-Nummer, so spricht der HP41 nur die niedrigste Page zuerst an, die zweite Page kann man z.B. über den CAT 2, beim 41CX mit CCD-Module über ENTER und XEQ aus dem Katalog erreichen. Einfacher ist es jedoch, die zweite Page in den 32kB-Block zu laden und diesen dann als aktiven Block zu wählen.

Der Befehl LOADP ermöglicht es, im Gegensatz zur W&W Rambox, auch noch bei Platzmangel in einer Page abzuspeichern, da das Programm automatisch in der folgenden leeren Page abgelegt wird.

Resume: Die neue Erweiterung besticht durch ein ausgefeiltes Handbuch, sowie durch einen niedrigen Preis.

Nachtragen möchte ich, daß die RSU auch in einer 32kB-Ausführung erhältlich ist. Die RSU wird in Deutschland ausschließlich durch ULRICH KUNZE COMPUTER SYSTEMS vertrieben.

Wiederum möchte ich nochmals betonen, daß dieser Testbericht keine Reklame für die RAM STORE UNIT der Firma ERAMCO SYSTEMS sein soll und daß der Vergleich mit den Konkurrenzprodukten meine eigene Meinung darstellt. Die Entscheidung für das eine oder andere System muß jeder Anwender selbst treffen. Wer weitere Fragen hat, kann sich ja an mich wenden.

Frank Lehmann (2759)
Humboldtstr. 30/84
7410 Reutlingen
☎/07121/320156 am Wochenende
☎0711/5300807 abends

.END.

Nachtrag zu "Best of PRISMA"

Da zwischen dem Zeitpunkt, zu dem ich meine Unterlagen für das Buch "Best of PRISMA" zur Verfügung stellte, und dem Erscheinungsdatum mehrere Jahre vergingen, erlaube ich mir, folgende Korrekturen anzubringen:

Seite 18: Ich führe den Druckerumbau nicht mehr durch.

Seite 16: Die Umbauanleitung ist auch für den IL-Drucker HP82162A gültig. Bei neueren Druckern wurde der Mikroschalter durch eine Lichtschranke ersetzt. Auch diese Modelle können unter Verwendung eines Mikroschalters umgebaut werden.

Seite 21: "Best of PRISMA" sollte ursprünglich Barcodes zu den abgedruckten Programmen enthalten. Aus diesem Grund und da ich damit rechnete, daß jemand, der Barcodes drucken möchte auch einen Lesestift besitzt, hatte ich darauf verzichtet, die Textketten in decodierter Form wiederzugeben. Dies möchte ich hiermit nachholen.

Winfried Maschke
Ursulakloster 4
5000 Köln 1

Seite 21: Programm ABARP+

Zeile 55: F1 3E
 Zeile 103: F0
 Zeile 112: F3 C0 00 2F
 Zeile 128: F0
 Zeile 166: F0
 Zeile 217: F0
 Zeile 233: F6 52 45 49 48 45 20
 Zeile 235: F3 7F 20 28
 Zeile 237: F2 7F 2D
 Zeile 239: F2 7F 29
 Zeile 283: F6 2D 2D 2D 2D 2D 2D
 Zeile 355: F8 7F 10 00 00 00 00 00 00
 Zeile 358: F8 7F 10 FA 1C 59 34 70 BE
 Zeile 361: F8 7F 10 70 40 81 02 06 08
 Zeile 364: F8 7F 11 FC 18 C6 10 20 BE
 Zeile 367: F8 7F 10 FA 0C 06 08 20 7F
 Zeile 370: F8 7F 10 40 87 F2 45 0C 10
 Zeile 373: F8 7F 10 FA 0C 08 0F C0 FF
 Zeile 376: F8 7F 10 FA 0C 17 E0 41 3C
 Zeile 379: F8 7F 10 08 20 82 08 20 7F
 Zeile 382: F8 7F 10 FA 0C 17 D0 60 BE
 Zeile 385: F8 7F 10 79 04 0F D0 60 BE
 Zeile 388: F8 7F 11 FC 08 13 E0 40 FF
 Zeile 391: F8 7F 11 01 01 02 04 10 40
 Zeile 394: F8 7F 10 04 10 40 81 01 01
 Zeile 397: F8 7F 11 05 09 17 F0 60 BF
 Zeile 400: F8 7F 10 70 40 81 02 04 1C
 Zeile 403: F8 7F 11 06 0C 1F F0 60 C1
 Zeile 405: F8 7F 10 00 00 0F E0 00 00

Seite 23: Programm ABARP

Zeile 55: F1 3E
 Zeile 89: F0
 Zeile 95: F3 C0 00 2F
 Zeile 111: F0
 Zeile 157: F0
 Zeile 211: F4 2D 2D 2D 2D

Seite 25: Programm BARA

Zeile 119: F6 7F 00 00 00 00 02
 Zeile 128: F1 20

Seite 28: Programm Q5X

Zeile 13: F6 01 41 05 00 00 00

Polynomdivision

Zur Berechnung von Integralen gebrochenra-

tionaler Funktionen oder zur Bestimmung schiefer Asymptoten ist es häufig notwendig, Polynome höheren Grades durch solche nie-

dereren Grades zu dividieren. Der allgemeine Fall einer solchen Division stellt sich folgendermaßen dar:

$$\begin{aligned}
 & (a_m x^m + a_{m-1} x^{m-1} + \dots + a_2 x^2 + a_1 x + a_0) : (b_n x^n + \dots + b_1 x + b_0) = \\
 & = \underbrace{E_{m-n} x^{m-n} + E_{m-n-1} x^{m-n-1} + \dots + E_1 x + E_0}_{\text{Ergebnispolynom}} + \underbrace{\frac{R_p x^p + R_{p-1} x^{p-1} + \dots + R_1 x + R_0}{b_n x^n + \dots + b_1 x + b_0}}_{\text{Restglied}}
 \end{aligned}$$

Sind sämtliche Nullstellen des Nenners auch solche des Zählers, so geht die Division auf; das Restglied ist in diesem Fall gleich Null.

Programmaufruf:

EINGABE	ANZEIGE	BEMERKUNGEN
XEQ"POL"	"N ZAEHLER ?"	Grad des Zählers ?
m R/S	"N NENNER ?"	Grad des Nenners ?
n R/S	"Zm= ?"	a _m =?
a _m R/S	"Zm-1= ?"	a _{m-1} =?
⋮	⋮	⋮
a ₁ R/S	"Z1= ?"	a ₁ =?
a ₀ R/S	"Z0= ?"	a ₀ =?
b _n R/S	"Nn= ?"	b _n =?
⋮	"Nn-1= ?"	b _{n-1} =?
⋮	⋮	⋮
b ₁ R/S	"N1= ?"	b ₁ =?
b ₀ R/S	"N0= ?"	b ₀ =?
⋮	⋮	⋮
R/S	...	Rechnung
⋮	"MIT REST"	oder "GEHT AUF"
⋮	"Em-n=..."	Ergebnispolynom
⋮	"Em-n-1=..."	⋮
R/S	⋮	⋮
R/S	"E1=..."	⋮
R/S	"E0=..."	⋮
R/S	"Rp=..."	Restpolynom (nur bei
R/S	"Rp-1=..."	"MIT REST", sonst Wieder-
⋮	⋮	holung des Ergebnispolynoms
R/S	"R1=..."	⋮
R/S	"R0=..."	ENDE. Falls abermals R/S
		gedrückt wird, wird die
		gesamte Ausgabe wiederholt.

FORTSETZUNG POLYNOMDIVISION ②

BEISPIELE :

$$1) (6x^4 - 3x^3 + x - 1) / (3x^3 - x - 1) = 2x - 1 + \frac{2x^2 + 2x - 2}{3x^3 - x - 1}$$

$$\begin{array}{r} 6x^4 - 2x^2 - 2x \\ \hline -3x^3 + 2x^2 + 3x - 1 \\ -3x^3 + x + 1 \\ \hline 2x^2 + 2x - 2 \end{array}$$

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

$$\begin{array}{r} 3x^4 - 3x^3 \\ \hline 3x^3 - x^2 - 3x + 1 \\ 3x^3 - 3x^2 \\ \hline 2x^2 - 3x + 1 \\ 2x^2 - 2x \\ \hline -x + 1 \\ -x + 1 \\ \hline - \end{array}$$

<1> BEISPIEL

<2> BEISPIEL

	XEQ "POL"		XEQ "POL"
N ZAEHLER ?	4 RUN	N ZAEHLER ?	4 RUN
N NENNER ?	3 RUN	N NENNER ?	1 RUN
Z4:	6 RUN	Z4:	3 RUN
Z3:	-3 RUN	Z3:	0 RUN
Z2:	0 RUN	Z2:	-1 RUN
Z1:	1 RUN	Z1:	-3 RUN
Z0:	-1 RUN	Z0:	1 RUN
N3:	3 RUN	N1:	1 RUN
N2:	0 RUN	N0:	-1 RUN
N1:	-1 RUN	GEHT AUF	
N0:	-1 RUN	E3: 3.000000	
MIT REST		E2: 3.000000	RUN
E1: 2.000000		E1: 2.000000	RUN
E0: -1.000000		E0: -1.000000	RUN
R2: 2.000000			
R1: 2.000000			
R0: -2.000000			

BEMERKUNG :

In den Zeilen 93, 99, 145, 159 befinden sich synthetische NOP's, die jedoch auch durch LBL 10 o.ä. ersetzt werden können.

FORTSETZUNG POLYNOMDIVISION ③

01 LBL"POL"	55 "└= ?"	109 X≠0?
02 CF 29	56 PROMPT	110 GTO 05
03 FIX 0	57 STO IND 04	111 ISG 03
04 9	58 ISG 04	112 GTO 04
05 STO 00	59 GTO 01	113 SF 10
06 "N ZAEHLER ?"	60 RCL 01	114 LBL 05
07 PROMPT	61 RCL 02	115 "MIT REST"
08 STO 01	62 -	116 FS?10
09 RCL 00	63 STO 04	117 "GEHT AUF"
10 +	64 RCL 01	118 TONE 9
11 E3	65 RCL 02	119 AVIEW
12 /	66 +	120 LBL 07
13 RCL 00	67 11	121 RCL 03
14 +	68 +	122 FRC
15 STO 03	69 STO Z	123 E3
16 STO 05	70 +	124 *
17 "N NENNER ?"	71 STO Z	125 RCL 03
18 PROMPT	72 E3	126 INT
19 STO 02	73 /	127 -
20 RCL 01	74 +	128 STO 02
21 10	75 STO 07	129 RCL 03
22 +	76 STO 08	130 STO 05
23 STO Z	77 RCL 06	131 STO 01
24 +	78 STO 02	132 RCL 08
25 E3	79 RCL 05	133 STO 07
26 /	80 STO 03	134 RCL 04
27 +	81 STO 01	135 STO 00
28 STO 04	82 LBL 02	136 LBL 06
29 STO 06	83 RCL IND 03	137 "E"
30 LBL 00	84 RCL IND 06	138 FIX 0
31 "Z"	85 /	139 ARCL 00
32 RCL 01	86 STO IND 07	140 "└="
33 RCL 03	87 LBL 03	141 FIX 5
34 -	88 RCL IND 07	142 ARCL IND 07
35 RCL 00	89 RCL IND 02	143 PROMPT
36 +	90 *	144 DSE 00
37 RND	91 ST- IND 01	145 ""
38 ARCL X	92 ISG 01	146 ISG 07
39 "└= ?"	93 ""	147 GTO 06
40 PROMPT	94 ISG 02	148 FS? 10
41 STO IND 03	95 GTO 03	149 GTO 07
42 ISG 03	96 RCL 06	150 LBL 08
43 GTO 00	97 STO 02	151 "R"
44 LBL 01	98 ISG 03	152 FIX 0
45 "N"	99 ""	153 ARCL 02
46 RCL 01	100 RCL 03	154 "└="
47 RCL 02	101 STO 01	155 FIX 5
48 +	102 ISG 07	156 ARCL IND 05
49 10	103 GTO 02	157 PROMPT
50 +	104 RCL 05	158 DSE 02
51 RCL 04	105 STO 03	159 ""
52 -	106 CF 10	160 ISG 05
53 RND	107 LBL 04	161 GTO 08
54 ARCL X	108 RCL IND 03	162 GTO 07
		163 END

Michael Schilli
Daucherstr. 2
8900 Augsburg

Gedächtnistest

85 Zeilen, 186 Bytes, SIZE 002, HP-41 CV

Und noch ein Spielprogramm aus dem Hause Schilli: In der Anzeige des Rechners blinkt für circa eine Sekunde eine Zufallszahl (4-10 Ziffern Länge) auf. Die Aufgabe des Spielers besteht nun darin, eben diese Zahl kurz darauf richtig wiederzugeben. Geschieht dies zweimal hintereinander fehlerlos, so wird die Anzahl der Ziffern der aufblinkenden Zahl um eins erhöht, was natürlich die Aufgabe etwas erschwert. Eine falsch eingetippte Zahl bleibt ohne Folgen, solange die darauffolgende fehlerlos eingegeben wird. Unterlaufen dem Spieler jedoch zwei Fehler hintereinander, so wird das Spiel beendet und eine Bewertung des Spielgeschehens ausgegeben.

Hier ein Beispiel für einen Programmdurchlauf:

XEQ"MEM" "5376"
"?"

5376 R/S

Der Spieler hat die Zahl also richtig wiedergegeben; der Rechner bestätigt dies durch einen hohen Ton. Nun erscheint die nächste Zahl:

"6335"
"?"

6335 R/S

Auch jetzt ist dem Spieler kein Fehler unterlaufen. Da er schon das zweite Mal in Folge fehlerlos arbeitet, erhöht sich nun die Anzahl der Ziffern:

"28463"
"?"

28469 R/S

Diese Eingabe war leider nicht richtig. Diese Tatsache gibt der Rechner durch zwei tiefe Töne bekannt. Noch einmal erscheint eine Zahl mit fünf Ziffern:

"74937"
"?"

74837 R/S

Auch jetzt ertönen zwei tiefe Töne, denn die Eingabe war wieder falsch. Da dies nun schon der zweite Fehler in Folge war, wird das Spiel abgebrochen und der Kommentar erscheint:

"SCHWACH"

Anschließend beginnt ein neues Spiel.

01 LBL"MEM"
02 LBL 00
03 CF 21
04 FIX 0
05 CF 29
06 SF 05
07 SF 06
08 4,01
09 STO 01
10 LBL 01
11 ,1
12 XEQ 03

13 3
14 %
15 -
16 X<=Y?
17 +
18 RCL 01
19 10/X
20 *
21 INT
22 VIEW X
23 PSE
24 " ?"
25 PROMPT
26 X≠Y?
27 GTO 02
28 SF 06
29 TONE 05
30 FS?C 05
31 GTO 01
32 SF 05
33 ISG 01
34 GTO 01
35 "HYPER-GENIUS"
36 AVIEW
37 BEEP
38 BEEP
39 GTO 00
40 LBL 02
41 SF 05
42 TONE 0
43 TONE 0
44 FS?C 06
45 GTO 01
46 XEQ IND 01
47 AVIEW
48 RCL 01
49 4
50 -
51 TONE IND X
52 PSE
53 GTO 00
54 LBL 04
55 "VOLLIDIOT"
56 RTN
57 LBL 05
58 "SCHWACH"
59 RTN
60 LBL 06
61 "MAESSIG"
62 RTN
63 LBL 07
64 "GUT"
65 RTN
66 LBL 08
67 "SEHR GUT"
68 RTN
69 LBL 09
70 "EXCELLENT"
71 RTN
72 LBL 10
73 "GENIAL"
74 RTN
75 LBL 03
76 RCL 00

77 PI
78 +
79 ENTER
80 X/2
81 *
82 FRC
83 STO 00
84 RTN
85 END

186 Bytes
SIZE 002

LEHRSTUHL FÜR
GENIE UND WAHNSINN
Z.H. MICHAEL SCHILLI
DAUCHERSTR.2
8900 AUGSBURG

.END.

Zeilenblockverwaltung

174 Zeilen, 266 Bytes, 38 Regs., SIZE var.
HP 41C/CV, X-F/M
Unterprogramme:
AZBL, CZBL, DZBL, GZBL, IZBL, SZBL

Das hier vorgestellte Programm dient der Verwaltung von Registerblöcken im Hauptspeicher, ähnlich der der ASCII-Files im X-Memory-Bereich.

Es gibt Fälle, bei denen es wünschenswert ist, nicht die Berechnungsergebnisse sondern deren Basisdaten zu speichern, einerseits, um die Eingaberoutine durch Wegfall der Rechen- und Sucharbeit zu kürzen, andererseits, um bei späteren Teiländerungen einen erneuten Rechengang komplett ausführen zu können, ohne alle Basisdaten nochmals eingeben zu müssen.

Bei der Realisierung solcher Programme sind oft sehr große Datenmengen zu verarbeiten, die unbedingt kompakt abgespeichert werden sollten.

Auf die vielfältigen Methoden der Datenverdichtung will ich hier nicht näher eingehen, wer hierüber etwas wissen will, kann u.a. in "Optimales Programmieren mit dem HP 41" von Gerhard Kruse, nachlesen.

Diese Unterprogramme sollen bei der Verwaltung der verdichtet gespeicherten Daten helfen, falls diese mehr als ein Register für eine logische Einheit benötigen. Es wird dem Anwender die Registerrechnung erspart und ihm ermöglicht, über eine Laufnummer auf einen ZEILENBLOCK, (so habe ich die logische Einheit von zusammengefaßten Registern genannt), zugreifen zu können.

FUNKTIONSBESCHREIBUNGEN

CZBL X **bb.lnnn**

Create Zeilen-Blockheader benutzt die Register 06...10 zur Verwaltung des durch die Steuerzahl im X-Register definierten Speicherbereiches im Hauptspeicher.

bb = Basisregister, es gilt:

$$1\emptyset + 1 + hh < bb < 1\emptyset\emptyset$$

wobei hh die vom Hauptprogramm benötigten Register sind. Mit **RCL** **F** ($\emptyset 6$)

kann bb jederzeit abgerufen werden.

l = Länge eines Zeilenblockes,

$$\emptyset < l < 1\emptyset$$

Mit **RCL** **I** ($\emptyset 9$) aufrufbar.

nnn = Anzahl der möglichen Zeilenblöcke,

$$\emptyset < nnn < (319 - bb) : 1$$

Mit **RCL** **H** ($\emptyset 8$) aufrufbar.

Das Programm versucht, mit PSIZE die erforderlichen Register bereitzustellen. Gelingt dies nicht, erscheint: NO ROOM, siehe XF-Modul-Anleitung.

Register 07 wird zurückgesetzt, es dient als Zählregister für die ausgeführten Speicher- oder Löschvorgänge. In Register 10 ist die Nr. des höchsten ansprechbaren Registers abgelegt, über diesem Register ist mindestens ein Block der Länge 1 als Puffer für die REGMOVE- Funktion vorhanden. Ein Überlauf (R07 > R08) muß vom aufrufenden Hauptprogramm überwacht werden.

$$\text{DZBL} \quad X \quad \text{nnn}$$

Delete Zeilen-Block löscht den angegebenen Block, wenn

$$X \leq \text{R } \emptyset 7 \quad \text{und} \\ \emptyset < \text{R } \emptyset 7 ;$$

in diesem Fall wird R 07 um 1 decrementiert, in allen anderen Fällen wird nichts ausgeführt. Die über X evtl. vorhandenen Zeilenblöcke werden alle um je eine Blocklänge nach unten verschoben.

$$\text{IZBL} \quad X \quad \text{nnn}$$

Insert Zeilen-Block fügt an der angegebenen Stelle ein, nachdem zuerst alle evtl. vorhandenen Blöcke mit gleicher oder höherer Nummer um jeweils eine Blocklänge nach oben verschoben wurden. R 07 wird um 1 incrementiert. Bei $X = \emptyset$

wird 1 als Ersatzwert verwendet, bei $X < \text{R } \emptyset 7$ wird AZBL ausgeführt.

$$\text{AZBL}$$

Append Zeilen-Block hängt an den durch R 07 gekennzeichneten Block an und incrementiert R 07 um 1.

$$\text{SZBL} \quad X \quad \text{nnn}$$

Save Zeilen-Block überschreibt den angegebenen Block, wenn $\emptyset < X \leq \text{R } \emptyset 7$

falls jedoch $X = \emptyset$ wird IZBL, falls

$X > \text{R } \emptyset 7$ wird AZBL ausgeführt.

Mit den Funktionen AZBL, IZBL und SZBL wird jeweils der Inhalt der Register

$$bb-1 \dots bb-1$$

übertragen. Diese Register müssen also vom Hauptprogramm zur Ablage der verdichteten Daten benutzt werden.

$$\text{GZBL} \quad X \quad \text{nnn}$$

Get Zeilen-Block ist die Umkehrfunktion von SZBL, sie legt den Inhalt der durch X spezifizierten Register in die Register

$$bb-1 \dots bb-1$$

zurück und incrementiert anschließend X um 1, wenn vorher

$$\emptyset < nnn \leq \text{R } \emptyset 7$$

war. In allen anderen Fällen erscheint NONEXISTENT.

Dieser Fehler muß ggfs. vom Hauptprogramm abgefangen werden.

PRP ""

LBL "CZBL"	40 RCL 06
02 INT	41 +
03 STO 06	42 RCL X
04 LASTX	43 RCL 09
05 FRC	44 -
06 EI	45 E3
07 *	46 /
08 INT	47 +
09 STO 09	48 RCL 07
10 LASTX	49 RCL Z
11 FRC	50 -
12 E3	51 RCL 09
13 *	52 *
14 STO 08	53 E6
15 DSE X	54 /
16 CLD	55 +
17 *	56 REGMOVE
18 +	57+LBL 00
19 PSIZE	58 DSE 07
20 RCL 09	59 CLD
21 -	60 RTN
22 STO 10	61+LBL 99
23 CLST	62+LBL "IZBL"
24 STO 07	63 ABS
25 RTN	64 INT
26+LBL "DZBL"	65 X=0?
27 ABS	66 E
28 INT	67 RCL 07
29 X=0?	68 X=0?
30 RTN	69 GTO 98
31 RCL 07	70 X<Y
32 X<Y?	71 X>Y?
33 X>Y?	72 GTO 98
34 RTN	73 DSE X
35 X=Y?	74 CLD
36 GTO 00	75 RCL X
37 RCL X	76 RCL 09
38 RCL 09	77 *
39 *	78 RCL 06

79 +	120 +
80 RCL X	129 RCL 06
81 RCL 09	130 RCL 09
82 +	131 -
83 E3	132 +
84 /	133 REGMOVE
85 +	134 RTN
86 X<Y	135+LBL "GZBL"
87 CHS	136 ABS
88 RCL 07	137 INT
89 +	138 X=0?
90 RCL 09	139 GTO 00
91 *	140 RCL 07
92 E6	141 X=0?
93 /	142 X<Y?
94 +	143 FS? 30
95 REGMOVE	144 GTO 01
96 RDN	145 X<Y
97 ISG X	146+LBL 00
98 CLD	147 E3
99 ISG 07	148 X<Y
100 CLD	149 STO IND Y
101 GTO 00	150 RTN
102+LBL 98	151+LBL 01
103+LBL "AZBL"	152 X<Y
104 ISG 07	153 RCL X
105 CLD	154 ISG Y
106 RCL 07	155 CLD
107+LBL 00	156 DSE X
108+LBL "SZBL"	157 CLD
109 ABS	158 RCL 09
110 INT	159 *
111 X=0?	160 RCL 06
112 GTO 99	161 +
113 RCL 07	162 LASTX
114 X<Y?	163 RCL 09
115 GTO 98	164 -
116 X<Y	165 E3
117 DSE X	166 /
118 CLD	167 +
119 RCL 09	168 RCL 09
120 *	169 E6
121 RCL 06	170 /
122 +	171 +
123 E3	172 REGMOVE
124 /	173 RDN
125 RCL 09	174 RTN
126 E6	175 .END.
127 /	

NACHWORT

Meine Hauptprogramme sind durchwegs länger als 2...4 KBytes, weswegen ich Modulsimulatoren wie Profi-Set oder RAMBOX benutze und deshalb den gesamten Hauptspeicher ohne weiteres zur Datenspeicherung verwenden kann. Diese Hilfsfunktionen laufen bei mir natürlich in XROM- Version in der Box ab.

An einem Anwendungsbeispiel will ich kurz aufzeigen, warum ich diese Programmierung bevorzuge.

Es sollen für ein Wohnhaus 80 Heizkörper berechnet werden. Zur Berechnung eines Heiz-

körpers sind bis zu 12 Daten erforderlich. Einzelnen abgespeichert sind somit $12 \times 80 = 960$ Register von Nöten, dies geht nur mit Cassetten-Laufwerk und ständigem Zugriff oder ähnlichem.

Nach meiner Methode verdichte ich die Daten eines Heizkörpers auf zwei Register und brauche insgesamt ca. 200 Register für das ganze Objekt. Zur Ablage auf Band ist nur ein einziger Zugriff nötig, während der ganzen Berechnungsdauer muß das Laufwerk nicht präsent sein. Wenn ich nach Monaten etwas ändern muß, (Änderungen kommen immer vor!) erreiche ich bequem z.B. den 65. Heizkörper unter seiner Nummer, ändere die Baulänge und speichere ihn wieder unter 65 mit SZBL ab. Sollte eine Fabrikatänderung vorgekommen sein, wird ein Index konvertiert und das gesamte Rechenprogramm läuft, über eine Schleife von 1...80 gesteuert, erneut ab. In welchem Datenregister welche Daten stehen, interessiert mich dann nicht mehr.

Uwe Triesberger
Harrößenstr. 23
7990 Friedrichshafen 5

.END.

Elektrotechnik

"HF - Tapete"

f-L - C - X Nomogramm

171 Zeilen, 282 Bytes, 41 Regs., SIZE 007
HP 41C

Wer mit Wechselstromschaltungen arbeitet, kennt es, das f-L-C-X Nomogramm. Im Laborjargon wird es oft "HF-Tapete" genannt. Es stellt den Scheinwiderstand (X) von Induktivitäten (L) und Kapazitäten (C) in Abhängigkeit von der Frequenz (f) grafisch dar. Für jeweils zwei gegebene Variablen (uV) können die

beiden anderen Variablen (aV) mehr oder weniger genau aus dem Nomogramm abgelesen werden. Den Vorteilen großer Übersichtlichkeit und der Vermeidung lästiger Rechenarbeit stehen Nachteile gegenüber:

Für eine ausreichende Ablesegenauigkeit sind recht sperrige Abmessungen nötig. Der Ablesefehlerteufel schlägt gelegentlich zu, besonders, wenn's eilt.

Ein Taschenrechnerprogramm kann leicht die Nachteile der HF-Tapete bei einer alle Ansprüche zufriedenstellenden Rechengenauigkeit vermeiden. Es muß aber mindestens ebenso benutzerfreundlich sein, wie sein Vorbild, damit es auch verwendet wird:

- Das Programm sollte wie andere Rechnerfunktionen immer verfügbar sein. Weil es dazu dauernd im Programmspeicher gehalten werden muß, soll sein Platzbedarf gering sein.

- Die Zahl der für die Programmsteuerung zu betätigenden Tasten sollte so klein wie möglich, die hierfür benötigten Tasten sollten leicht zu merken und räumlich zusammengefaßt sein.

- Eingegebene Werte sollen zur Kontrolle angezeigt und wie die berechneten Ergebnisse zur weiteren Bearbeitung im Speicher gehalten werden.

"HF" vereint die teilweise widersprüchlichen Forderungen zu einem brauchbaren Kompromiß. Die alphanumerische Anzeige und die Flagindikatoren dienen der eindeutigen Wertausgabe und Variablenkennzeichnung. Die

Programms Ausführungszeiten sind dabei ausreichend kurz. Hier könnte ein Spezialist für synthetische Programmierung vermutlich noch Verbesserungen erreichen.

Alle Größen werden in den Grund-SI-Einheiten Hertz, Henry, Farad, Ohm, also ohne Dezimalvorsätze wie k, m, M, n, p eingegeben. Das Programm stellt beim Start mit [XEQ] "HF" das Anzeigeformat ENG 3 und den USER-Modus ein. Damit sind die Tasten [A](Frequenz f), [B] (Induktivität L), [C] (Kapazität C) und [D] (Blindwiderstand) für die Eingabe und Ausgabe der Variablen belegt. Der Rechner unterscheidet (Flag 22) Zahleneingaben von Ergebnisanforderungen; es genügt, die zur jeweiligen Variablen gerechnete bzw. gespeicherten Werte als unabhängige Variable (uV) eingegeben werden, ist vorher Taste [E] (SF22) zu drücken. Das berechnete Ergebnis steht in Form einer zugeschnittenen Größengleichung in der Anzeige, der Zahlenwert im X-Register und zugehörigen Datenspeicherregister. Ändert sich, was häufig vorkommt, bei aufeinanderfolgenden Berechnungen nur eine der beiden uV, so muß nur der geänderte Wert eingegeben werden. Die Flagindikatoren in der Anzeige zeigen immer, mit welchem uV das Programm rechnet. Die Zuordnung entspricht der Spaltennummer der Funktionstaste.

Die aV werden abhängig von den eingegebenen uV wie folgt berechnet

$$\begin{aligned}
 f &= 1 / (2 * \pi * C * X) = X / (2 * \pi * L) = 1 / (2 * \pi * \sqrt{L * C}) \\
 L &= X / (2 * \pi * f) = X^2 * C = 1 / (4 * \pi^2 * f^2 * C) \\
 C &= 1 / (2 * \pi * f * X) = L / X^2 = 1 / (4 * \pi^2 * f^2 * L) \\
 X &= -1 / (2 * \pi * f * C) = 2 * \pi * f * L = \sqrt{L / C}
 \end{aligned}$$

AUS UNSEREM ANGEBOT:

HP-41 RAM STORAGE UNIT 32k	im Cardr.geh.	DM 796.86
HP-41 RAM STORAGE UNIT 64k	im Cardr.geh.	DM 1252.86
HP-41 Module:		
-AEC-ROM	DM 299.-	GR7AG-IL Graphik-
-MATH/STAT 8k	DM 129.-	Video Controller DM 1659.-
-ADVANTAGE ROM 12k	DM 139.-	Komf. Systemkoffer
-X-MEMORY	DM 199.-	für HP-41,71,75 DM 399.-
-EXT. I/O	DM 199.-	NEC P6 Color DM 1819.-
Infrarot-Drucker für HP 18C	DM 329.-	NEC MultiSync DM 1998.-
NEU!! HP 28C	DM 569.-	Portables versch. Hersteller

ERAMCO SYSTEMS - HP - NEC - TANDON - TOSHIBA - ZENITH

Preise zuzügl. Porto u. Verpackung. Versand nach Vorkasse.

ULRICH KUNZE COMPUTER SYSTEMS Postfach 300667 7000 Stuttgart 30
Tel.: 0711/886159

Bedienungsanweisung

Nr	Operation	Variable	Taste(n)	Anzeige	F
1	Programm eingeben				
2	Datenregisterbereich festlegen		SIZE 007		
3	Programm starten		[XEQ] "HF"		
4	Eingabe von zwei der vier Variablen in beliebiger Reihenfolge: Frequenz in Hertz Induktivität in Henry Kapazität in Farad Blindwiderstand in Ohm Die beiden für die Berechnung benutzten Größen sind durch die entsprechenden Zahlen im Flagindikator am unteren Rand des Anzeigefeldes markiert.	f/Hz L/H C/F X/Ω	[A] [B] [C] [D]	F/Hz=(f/Hz) L/H=(L/H) C/F=(C/F) X/O=(X/Ω)	1 2 3 4
5	Berechnen der gesuchten Größe(n): Frequenz Induktivität Kapazität Blindwiderstand		[A] [B] [C] [D]	F/Hz=(f/Hz) F/H=(L/H) C/F=(C/F) X/O=(X/Ω)	
6	Um eine beliebige Variable erneut zu berechnen: nach Nr.5				
7	Um eine oder alle Variablen zu ändern: nach Nr.4.				
8	Um ein in der Anzeige stehendes Ergebnis einer zuvor durchgeführten Rechnung als neue Variable einzugeben: anschließend nach Nr.7.		[E]		

Registerbelegung

- R00: Index
- R01: f/Hz
- R02: L/H
- R03: C/F
- R04: X/Ω
- R05: letzte Kennziffer
- R06: vorl. Kennziffer

01*LBL "HF"	27 XEQ 11	53 RCL 03	79 GTO 02	105*LBL D	168 RTN
02 CF 22	28 GTO 02	54 /	80 GTO IND 00	106 FS?C 22	169*LBL E
03 SF 27	29*LBL 08	55 GTO 02	81*LBL 03	107 GTO 01	170 SF 22
04 CLX	30 RCL 03	56*LBL 06	82 E	108 RCL 04	171 END
05 STO 05	31 RCL 04	57 RCL 04	83 XEQ 10	109 FS? 04	
06 ENG 3	32 ABS	58 ABS	84 X↑2	110 GTO 02	
07 STOP	33 *	59 1/X	85 RCL 02	111 GTO IND 00	
08*LBL A	34 XEQ 11	60 XEQ 10	86 /	112*LBL 03	
09 FS?C 22	35 GTO 02	61 GTO 02	87 GTO 02	113 RCL 02	
10 GTO 01	36*LBL 01	62*LBL 08	88*LBL 06	114 XEQ 10	
11 RCL 01	37 E	63 RCL 04	89 RCL 04	115 1/X	
12 FS? 01	38 XEQ 09	64 X↑2	90 ABS	116 GTO 02	
13 GTO 02	39*LBL 02	65 RCL 03	91 XEQ 10	117*LBL 04	
14 GTO IND 00	40 "F/HZ="	66 *	92 GTO 02	118 RCL 03	
15*LBL 05	41 GTO 15	67 GTO 02	93*LBL 07	119 CHS	
16 RCL 02	42*LBL B	68*LBL 01	94 RCL 02	120 XEQ 10	
17 RCL 03	43 FS?C 22	69 2	95 RCL 04	121 GTO 02	
18 *	44 GTO 01	70 XEQ 09	96 X↑2	122*LBL 05	
19 SQRT	45 RCL 02	71*LBL 02	97 /	123 RCL 02	
20 XEQ 11	46 FS? 02	72 "L/H="	98 GTO 02	124 RCL 03	
21 GTO 02	47 GTO 02	73 GTO 15	99*LBL 01	125 /	
22*LBL 07	48 GTO IND 00	74*LBL C	100 3	126 SQRT	
23 RCL 02	49*LBL 04	75 FS?C 22	101 XEQ 09	127 GTO 02	
24 RCL 04	50 E	76 GTO 01	102*LBL 02	128*LBL 01	
25 ABS	51 XEQ 10	77 RCL 03	103 "C/F="	129 4	
26 /	52 X↑2	78 FS? 03	104 GTO 15	130 XEQ 09	
				131*LBL 02	
				132 "X/O="	
				133*LBL 15	
				134 ARCL X	
				135 AVIEW	
				136 STOP	
				137*LBL 10	
				138 RCL 01	
				139 *	
				140*LBL 11	
				141 PI	
				142 *	
				143 2	
				144 *	
				145 1/X	
				146 RTN	
				147*LBL 09	
				148 RCL 05	
				149 X=Y?	
				150 GTO 00	
				151 CF 00	
				152 CF 01	
				153 CF 02	
				154 CF 03	
				155 CF 04	
				156 SF IND X	
				157 SF IND Y	
				158 X<>Y	
				159 STO 05	
				160 +	
				161 STO 00	
				162 E	
				163 FS? 04	
				164 ST+ 00	
				165*LBL 00	
				166 RCL Z	
				167 STO IND 05	

Beispiele

1.1 Gegeben: L=100H; f=50Hz

Gesucht: C und Kennwiderstand bei Resonanz

Tastenfolge	Anzeige	Flagindikator
[XEQ] [α] HF [α]	0.000 00	
100 [B]	L/H=100.0E 0	0 2
50 [A]	F/Hz=50.00E 0	1 2
[C]	C/F=101.3E-9	1 2
[D]	X/O=31.42E 3	1 2
Kontrolle der für f und L eingegebenen Werte		
[A]	F/Hz=50.00E 0	1 2
[B]	L/H= 100.0E 0	1 2

1.2 Gegeben: f=50Hz wie bei 1.1 und C=220nF

Tastenfolge	Anzeige	Flagindikator	X _c
220 [EEX] 9 [CHS] [C]	C/F=220.0E-9	1 3	
[D]	X/O=-14.47E 3	1 3	

1.3 Gegeben: Der noch in dr Anzeige stehende bei 1.2 berechnete Blindwiderstand X_c

Tastenfolge	Anzeige	Flagindikator	für f=1kHz die entsprechende Induktivität und Kapazität
[E] [D]	X/O=-14.47E 3	3 4	
[EEX] 3 [A]	F/Hz=1.000E 3	1 4	
[B]	L/H=2.303E 0	1 4	
[C]	C/F=11.00E-9	1 4	

2 Gegeben: C=12nF, Kennwiderstand X=15Ω

Tastenfolge	Anzeige	Flagindikator	für L	Resonanz
12 [EEX] 9 [CHS] [C]	C/F= 12.00E-9	1 3		
15 [D]	X/O=15.00E 0	3 4		3 4
[B]	L/H=2.700E-6	3 4		

3 Gegeben: Schwingkreis L=0,6mH C₁=2,51nF und C₂=0,251nF

Tastenfolge	Anzeige	Flagindikator	f ₁	und	f ₂
.6 [EEX] 3 [CHS] [B]	L/H= 600.0E-6	2 4			
2.51 [EEX] 9 [CHS] [C]	C/F= 2.510E-9	2 3			
[A]	F/Hz= 129.7E 3	2 3			
.251 [EEX] 9 [CHS] [C]	C/F=251.0E-12	2 3			
[A]	F/Hz= 410.1E 3	2 3			

Anmerkung:

Zur Anzeige kommen immer die beiden Flags die die zuletzt eingegebenen, für die Berechnung verwendeten uV kennzeichnen. Der Rumpf des Programms kann natürlich auch zur Ablaufsteuerung anderer, ähnlicher Aufgaben verwendet werden.

Enzian J. Schneider
Eulenweg 6
7416 Trochtelfingen

.END.

AECROM Testbericht

Nach der ersten Ankündigung dieses neuen Moduls im letzten PRISMA '86 habe ich inzwischen ein solches Wunderwerk der Technik erhalten und möchte mich etwas detaillierter darüber auslassen.

Erst einmal ein paar einleitende Worte:

Dem Modul liegt ein DIN-A5 Ringbuch von gut 100 Seiten Stärke bei, mit jeweils einem Beispiel dazu, ganz hinten sind noch die 6 Beispielprogramme als Barcodes abgedruckt. Die letzte Seite schmückt ein Bestellformular für weitere Module, das nur am Rande erwähnt.

Nach einigen anfänglichen Grußworten stößt man auf ein ordentliches Inhaltsverzeichnis. Darauf folgen Hinweise zu folgenden Unverträglichkeiten des AECROMs:

1) Das SURVEYING PAC muß in einem Port

mit einer höheren Nummer als das AECROM stecken.

2) Es wird auf eine "Inkompatibilität" zum CCD-Modul und zum ZENROM hingewiesen, allerdings ohne Details über die Ursache.

3) Bezüglich des CCD-Moduls kann ich von folgenden Erfahrungen berichten: hat das AECROM eine höhere Portnummer als das CCD-Modul, so laufen bei mir bis jetzt alle beide Module anstandslos. Hat das AECROM allerdings eine niedrigere Nummer als das CCD-Modul, so erscheint das CCD-Modul zwar im CAT 2 und es lassen sich alle Funktionen ausführen, was aber nicht funktioniert sind seine betriebssystemorientierten Fähigkeiten wie die erweiterten CAT-Funktionen und die erweiterte ALPHA-Tastatur.

Das liegt daran, das beim Einschalten des Rechners nach betriebssystemorientierten Funktionen in Modulen gesucht wird und wer zuerst gesehen wird, malt. Das AECROM enthält nämlich auch Funktionen dieser Art. So, nach der langen Einleitung nun zu den Tatsachen; das 1. Kapitel befaßt sich mit den

Längeneinheitenmodi

a) Dezimal Fuß Modus ">DF"

In diesem Modus ist der HP 41 standartmäßig, hier ist die Anzeige ganz normal, die "Fuße" vor dem Komma, nach dem Komma die "Füße" in zehnteln, hundertsteln usw.: Format -> F.ffffff. In diesem Modus wird auch in all den anderen Modi gerechnet, auch wenn

das Rechnerdisplay im entsprechenden Modus anders anzeigt.

b) Dezimal Inch Modus ">DI"

In diesem Modus zeigt das Display ganze Inches vor dem Komma und Bruchteile davon nach dem Komma an, die Eingaben erfolgen im selben Format und werden intern umgerechnet, sodaß im Stack immer Zahlen im Dezimal Fuß stehen. Format -> I.iiiiiii. Geht man von einem Modus in den anderen, so wird immer der korrekte Wert angezeigt; ein Beispiel: Ich gehe in den Dezimal Fuß Modus und tippe 12.75 Fuß ein, schalte in den Dezimal Inch Modus, siehe da, das Display zeigt mir 153.00 Inch an.

c) Dezimal Meter Modus ">M"

Dieser Modus zeigt uns unser gewohntes Bild, Meter vor dem Komma und Zentimeter und Millimeter nach dem Komma. Format -> M.mmmmmmmmm.

d) Bruchmodus (Fraction Mode) ">F"

Hat man in diesen Modus geschaltet, so ändert sich das Display drastisch: Ich habe 145.214789 Fuß im Dezimal Fuß Modus eingegeben und schalte nun in den Bruch Modus um >> das Display zeigt 1452-9/16", was soviel bedeutet wie 145 Fuß, zwei neun sechszehntel Inch, bei allen Maschinenbauern lacht jetzt wahrscheinlich das Herz, schöner geht es kaum. Die Dateneingabe erfolgt in dieser Betriebsart im Format -> F.Illfff. F sind ganze Füße, ll ganze Inches, ffff der maximal vierstellige Bruchanteil.

Soweit die vier Eingabe/Displaymodi, man wechselt sie entweder durch Aufrufen ihres Namens oder indem man auf die ON Taste drückt. Hält man diese fest, so wandern die vier Modi im Klartext durch das Display; beim gewünschten Modus läßt man dann die ON Taste los (das ist eine von den betriebssystemzugreifenden Funktionen).

In dieses Kapitel gehören noch zwei weitere Funktionen, "/CONV" und "*CONV", sie dienen der Korrektur von Berechnungen im Dezimal Inch und Dezimal Fuß Modus, da hierbei durch das interne Fuß-Format falsche Anzeigen entstehen, über Details kann man dann im Handbuch nachlesen.

Das 2. Kapitel befaßt sich mit dem Programmgenerator

Diese Funktion erlaubt das Eingeben einer Formel im Klartext in nahezu beliebiger Länge; man ist nicht auf die Länge des ALPHA-Registers angewiesen (es wird ohnehin nicht benutzt). Aus dieser eingegebenen Formel wird dann in den Programmspeicher hinein ein sofort lauffähiges Programm geschrieben, das im Klartext die Variablen erfragt und im Prinzip aussieht wie jedes andere handgeschriebene Programm. Zur Eingabe der zu programmierenden Formel wird die Tastatur zum Teil neu belegt, aber das sieht man am besten auf dem Bild 1, dort ist die neue Tastaturbelegung zu sehen (die Buchstaben entsprechen jeweils einer Variablen). Negativ anzumerken wäre die fehlende Tastaturschablone, man braucht deshalb immer das Handbuch in Reichweite. Vielleicht war das bei dem relativ niedrigen Preis nicht mehr drin.

So, nun wollen wir doch einmal in's Detail gehen: Der Programmgenerator wird mit dem XEQ"PROG" gestartet und meldet sich mit einem prompt zur Eingabe des Namens, den das zu programmierende Programm haben soll, dem Anfangslabel. Nach dessen Eingabe

be erscheint kurz PACKING in der Anzeige, dem letzten Programm im Speicher wird, falls nötig, noch ein END verpaßt. Nun steht im Display groß und breit "ENTER:FORMULA", was wohl eine unmißverständliche Aufforderung zur Eingabe der Formel darstellt. Ab jetzt gilt die neue Tastatur (siehe Bild 1).

Wir nehmen mal ein einfaches Beispiel für den Anfang: Obst = Äpfel + Birnen.

Mit XEQ"PROG" rufen wir den Programmgenerator auf, tippen "OBST" für den Programmnamen ein und werden mit "ENTER:FORMULA" zur Eingabe der Formel aufgefordert. Nun tippen wir A als Variable für Äpfel, SHIFT +, UNSHIFT B für Birnen ein, ja, ihr habt richtig gelesen, SHIFT muß man wieder zurücknehmen wie beim CAT'2 mit dem CCD-Modul. Das hat durchaus seinen Sinn,

wenn man Konstanten eingibt müßte man sonst dauernd SHIFT drücken.

So, im Display steht nun A+B, mit R/S geht's weiter. Es erscheint für etwa 1s die Aufforderung "ENTER:LBL,CONS", danach möchte das Programm den Namen der Variable A haben, im Display steht jetzt "A..=". Wir tippen jetzt AEPFEL ein, die Aufforderung bleibt im Display, es steht dort also jetzt "A..=AEPFEL".

R/S beendet diese Eingabe, es erscheint "B..=" in der Anzeige, nun nehmen wir BIRNEN und drücken R/S, es erscheint die Frage nach der zu errechnenden Größe mit "ANS =" (wie answer oder Antwort), hier tippen wir OBST ein und drücken R/S.

Jetzt hüpf das Herz, im Display erscheint die

Meldung PROGRAMMING..., was soviel heißt wie "Bitte etwas Geduld". Nach etwa 2,5 s kommt ein kurzes PACKING und danach die Meldung RUN SIZE >= 02, d.h. zum Starten des jetzt mit "OBST" oder einfach auch mit R/S zu startenden Programms wird mindestens SIZE 002 benötigt.

Unter Beispiel 1 findet ihr das Listing der Eingaben im TRACE-Modus und darunter das vom Programmgenerator erzeugte Programm.

Beispiel 1

PROG "OBST"

```
PACKING
ENTER:FORMULA
A+B
ENTER:LBL,CONS
A . . =AEPFEL
B . . =BIRNEN
ANS=OBST
PROGRAMMING. .
PACKING
RUN SIZE)>=02
```

```
11:10 02.01 08 RCL 00
01+LBL "OBST" 09 RCL 01
02 "AEPFEL=?" 10 +
03 PROMPT 11 "OBST= "
04 STO 00 12 AVIEW
05 "BIRNEN=?" 13 PSE
06 PROMPT 14 CLD
07 STO 01 15 .END.
```

Das 2. Beispiel ist etwas komplizierter, es ist eine Gleichung aus der Schallwelldynamik:

$$dp = \frac{4\pi^2}{T^2} \sigma \uparrow \sin 2\pi(T \frac{z}{\delta} - \frac{x}{\delta})$$

Wir gehen nach demselben Schema wie in Beispiel 1 vor: XEQ "PROG", als Programmnamen nehmen wir BESCHL für Schallbeschleunigung. Die Eingaben entnehmt bitte dem Listing Beispiel 2.

Beispiel 2

PROG "BESCHL"

```
PACKING
ENTER:FORMULA
((4*pi^2)/T^2)01 SIN(2
pi(Z/T-X/P))
ENTER:LBL,CONS
I . . =I
O . . =0
P . . =P
T . . =T
X . . =X
Z . . =Z
ANS=BESCHL
PROGRAMMING. .
PACKING
RUN SIZE)>=08
```

Tastenbelegung im Programmgenerator

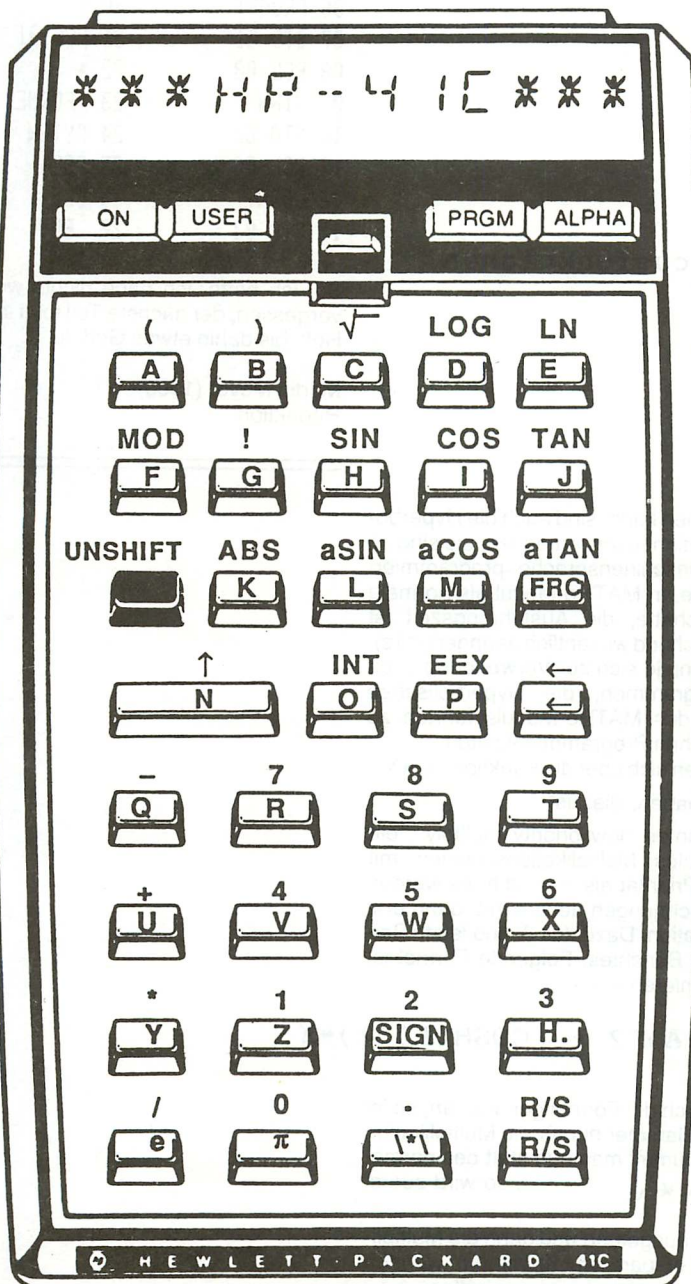


Bild 1

```

15:55 05.01 30 /
01+LBL "BESCHL" 31 STO 06
02 "I=?" 32 RCL 05
03 PROMPT 33 RCL 03
04 STO 00 34 /
05 "0=?" 35 RCL 04
06 PROMPT 36 RCL 02
07 STO 01 37 /
08 "P=?" 38 -
09 PROMPT 39 STO 07
10 STO 02 40 2
11 "T=?" 41 PI
12 PROMPT 42 *
13 STO 03 43 RCL 07
14 "X=?" 44 *
15 PROMPT 45 STO 07
16 STO 04 46 RCL 06
17 "Z=?" 47 RCL 01
18 PROMPT 48 *
19 STO 05 49 RCL 00
20 4 50 *
21 PI 51 RCL 07
22 2 52 SIN
23 Y↑X 53 *
24 * 54 "BESCHL= "
25 STO 06 55 AVIEW
26 RCL 06 56 PSE
27 RCL 03 57 CLD
28 2 58 .END.
29 Y↑X
    
```

Wie aus dem Ausdruck der Formel zu ersehen ist, muß man sich im Rahmen der Grundrechenarten um die Klammersetzung kümmern, wobei einem hierbei ein wenig geholfen wird. Es gelten die Regeln Punkt-vor-Strichrechnung, davor Funktionen wie SIN und Potenzen, insgesamt also die übliche Prioritätenhierarchie. Das Multiplikationszeichen ist optimal, da für Variablen nur ein Buchstabe zulässig ist. Bei Funktionen wie SIN braucht man auch nur dann Klammern, wenn ein Ausdruck als Argument verwendet wird. PI ist auf der Tastatur vorhanden und wird auf den Thermodruckern auch korrekt ausgedruckt, Besitzer des Thinkjets haben hier schlechte Karten. Wie an dem aufgelisteten Beispiel zu erkennen ist, erfolgt die Eingabe der Variablenamen alphabetisch, die Programmierdauer dieser Formel betrug etwa 9 Sekunden. Was der Programmgenerator leider zum Schluß des erzeugten Programms vergißt ist der Befehl VIEW X; im NORM Modus werden nämlich auf dem Thermodrucker während der Programmabarbeitung schön brav alle Eingaben aufgelistet, das Endergebnis fehlt aber. Man muß eben von Hand die PRINT Taste drücken, ein unnötiger Schönheitsfehler.

Mit etwas Übung kann man also innerhalb weniger Sekunden eine Formel selbst größeren Umfangs programmieren lassen, selbst für absolute Programmierverweigerer oder frische Besitzer eines HP41 ist diese Funktion des Moduls Gold wert. Vor einem Fehler des Programmgenerators muß ich allerdings an dieser Stelle warnen: Drückt man an der Stelle ENTER:FORMULA aus Versehen die R/S Taste um z.B. den Programmgenerator abzubrechen (normal die Backarrow Taste), so sucht der Programmgenerator auf Nimmerwiedersehen nach der

nicht eingegebenen Formel, da hilft nur noch Batterie raus und wieder rein, für Besitzer des TIME-Moduls besonders schmerzlich, da die Uhrzeit nach dieser Aktion nicht mehr korrekt ist.

Nun der Übersicht zuliebe eine Liste aller verfügbaren Funktionen des Programmgenerators:

- ()
- √
- LOG
- LN
- MOD
- ! (FACT)
- ABS
- INT
- FRC
- SIGN
- ^ (Potenz)
- e
- π
- EEX
- SIN
- SINH
- COS
- COSH
- TAN
- TANH

alle Arcusfunktionen

- *
- * \
- /
- +

Wie man sehen kann, sind auch die Hyperbolischen Funktionen implementiert, sie sind im Modul in Maschinensprache programmiert und nicht wie im MATHe Modul als normale Programmschritte, die Ausführungszeit ist dementsprechend wesentlich geringer (<1s). Damit eignen sie sich zur Verwendung in eigenen Programmen, die Hyperbolischen Funktionen des MATHe-Moduls führten zu astronomischen Programmlaufzeiten.

Einige werden sich über die Funktion * \ gewundert haben, dies ist ein sogenanntes "low priority multiply", ein ganz normales Multiplikationszeichen mit niedrigerer Priorität als + -, d.h. es werden erst Strichrechnungen ausgeführt, dann erst die Multiplikation. Dazu das 3. und letzte Beispiel dieses Berichtes. Folgende Formel ist zu programmieren -->>

$$((\text{SINH } A)^2 + (\text{COSH } A)^2) * X$$

Sieht man sich die Formel genauer an, so ist die äußere Klammer nur für die Multiplikation notwendig. Nimmt man nun statt des normalen * das * \ , so wird zuerst

die Addition ausgeführt und dann die Multiplikation. Dies erspart zwei Klammern, wonach wir folgendes zu programmieren hätten -->>

$$(\text{SINH } A)^2 + (\text{COSH } A)^2 \ \backslash * \ X$$

Beispiel 3

```

PROG "FORMEL"

PACKING
ENTER:FORMULA
( SINH.A)↑2+( COSH.A
)↑2\*\X
ENTER:LBL,CONS
A . .=A
X . .=X
ANS=FORMEL
PROGRAMMING.

18:17 05.01 14 RCL 02
01+LBL "FORMEL" 15 2
02 "A=?" 16 Y↑X
03 PROMPT 17 RCL 03
04 STO 00 18 2
05 "X=?" 19 Y↑X
06 PROMPT 20 +
07 STO 01 21 RCL 01
08 RCL 00 22 *
09 SINH 23 "FORMEL= "
10 STO 02 24 AVIEW
11 RCL 00 25 PSE
12 COSH 26 CLD
13 STO 03 27 .END.
    
```

So, ich hoffe, ich habe nichts wesentliches vergessen, der nächste Teil folgt im nächsten Heft, bis dahin etwas Geduld.

Martin Meyer (1000)
Redaktion

.END.

M?

Memory String

**61 Zeilen, 111 Bytes, 16 Regs., SIZE 000
HP 41C, X-F/M, CCD-ROM-Handbuch**

Bei der Arbeit mit dem HP 41 interessiert oft - wegen der doch eher beengenden Speichergröße - der Belegungszustand des Hauptspeichers. Ein Bild sagt mehr als tausend Zahlen. In diesem Fall soll eine Zeichenkette den Belegungszustand des Hauptspeichers repräsentieren. Länge des Strings 10 Character, d.h. jedes Zeichen steht für 32 Register (beim CX). Angezeigt wird, ob ein solcher Registerblock durch Daten, Programme, Buffer belegt oder frei ist. Z.B. repräsentiere

'&' = 32 Datenregister
'+' = 32 Programmregister
'/' = 32 Bufferregister
'.' = 32 freie Register

dann sagt
&&++++----/
alles.

Aus Gründen der Programmkürze steht bei M? allerdings - entgegen der Speicherstruktur - der Bufferstring vor den unbelegten Registern. Ansonsten stehen links die hohen Adressen und rechts die niedrigen.

Das Programm M? liefert auf Knopfdruck den Belegungszustand in der Stringform. Leider ist es mir nicht gelungen das Programm noch zu verkürzen. Hilfreich wäre der direkte Zugriff auf die Zahl der noch unbelegten Register (wie sie z.B. bei GTO.. angezeigt wird). Aber wo steht die? Also muß diese Zahl mühsam berechnet werden (unter anderem wird deshalb A? aus dem CCD-Modulhandbuch verwendet).

Durch einfache Änderung des Programmes können auch andere als die von mir gewählten Symbole genommen werden, wenn diese nicht zusagen (z.B. ist das Zeichen ' nicht optimal, weil es kein volles Displayelement belegt und dadurch das Bild verzerrt; aber bei mir werden sowieso kaum mehr als 15 Bufferregister belegt).

Es treten unvermeidbare Rundungsfehler auf. So können sowohl nur 16 als auch 47 Register zu einem Zeichen führen. Dieser Effekt läßt sich durch einen längeren String oder durch zusätzliche Zeichen (50%-Belegung o.ä.) minimieren. Aber Grundidee von M? war nicht die blutleere Exaktheit sondern des HP-Speichererlebens pralle Fülle (oder besser Gefülltheit).

Bei mir hat M? schon unzählige, erfolgreiche Einsätze hinter sich. Meines Erachtens ist M? aus dem Holz geschnitzt aus dem man benutzerfreundliche Grundfunktionen macht. Aber diese Holzart scheint in Silicon-valley unbekannt zu sein. Oregon jedoch sollte noch genug Bäume haben.

Happy programming

01+LBL "M?"	11 *
02 13	12 R†
03 PEEKB	13 +
04 X<>Y	14 CHS
05 A+	15 512
06 PEEKB	16 +
07 16	17 XEQ "A?"
08 MOD	18 X<>Y
09 LASTX	19 SIZE?
10 X†2	20 ST- Y

21 32	42 ISG X
22 ST/ Y	43 XEQ 14
23 ST/ Z	44 +
24 ST/ T	45 ISG X
25 CLX	46 XEQ 14
26 1,5	47 11
27 ST+ Y	48 ALENG
28 ST+ Z	49 -
29 ST+ T	50 X<>Y
30 RDN	51 ISG X
31 INT	52 XEQ 14
32 X<>Y	53 PROMPT
33 INT	54+LBL 14
34 X<> Z	55 DSE Y
35 INT	56 GTO 13
36 STO T	57 RTN
37 RDN	58+LBL 13
38 CLA	59 XTOA
39 42,9	60 GTO 14
40 XEQ 14	61 END
41 +	

Laßt den HP 41 nicht sterben. Dieser lebt in einer ökologischen Werkzeugnische, die ihn und seine Nachfolger auch in Zukunft unentbehrlich machen. Nur sollte er dort eingesetzt werden, wo er stark ist. Eine unangemessene Aufmotzung (Diskette für den 41-er halte ich für Blödsinn, schon der Drucker ist eine Zumutung) schadet. Durch die sinkenden PC-preise wird er sich nur durch Besinnung auf seine Grundtugenden auf Dauer halten können. Ich jedenfalls werde noch lange meine Berechnungen mit dem HP 41 durchführen. HP-Besitzer Deutschlands vereinigt euch im CCD!!

Erich Neubacher (3085)
Gotenstr. 120
5300 Bonn 2

Einkommensteuer

STEUER 86/88

**EST86, 130 Zeilen, 284 Bytes, 41 Regs., SI-
ZE 000
HP 41C**

Lieber Vati!

Du hast ja recht, wenn Du sagst, was soll ich mit den alten Steuertarifen, wenn aber 1986 alles viel besser wird und wir weniger Steuern zahlen müssen. Überlastet wie ich nun mal bin, habe ich den Tarif bis 1985 im Prisma 86.8.25 verwendet. Hier ist nun das Programm mit dem Tarif für 1986 (dieser Tarif gilt auch für das Jahr 1987.

Durch einen Fehler beim Überspielen des Programms entsprach das Barcodeprogramm nicht dem gelisteten Programm. Bei ersterem steht in der Anzeige noch die für Dich bedeutungslose Einkommensteuertarifbasis.

Zum Trost habe ich gleich noch ein Programm für den Tarif ab 1988 geschrieben.

01+LBL "EST86"	64 INT
02 1	65 RTN
03 SF 02	66+LBL 02
04 "SPLIT?J:N N=0"	67 35
05 TONE 7	68 -
06 PROMPT	69 X<>Y
07 X=0?	70 80027
08 CF 02	71 X<Y?
09 "EINKUENFTE?="	72 GTO 03
10 TONE 7	73 X<>Y
11 PROMPT	74 ENTER†
12 GTO 00	75 R†
13+LBL "ESTG"	76 -
14 CF 02	77 E2
15 GTO 00	78 /
16+LBL "ESTG"	79 21 E2
17 SF 02	80 %
18+LBL 00	81 INT
19 ,5	82 56020
20 X<>Y	83 -
21 FS? 02	84 %
22 *	85 INT
23 XEQ E	86 6 E5
24 FC? 02	87 +
25 GTO 00	88 %
26 2	89 INT
27 ST* Z	90 22 E5
28 *	91 +
29+LBL 00	92 *
30 FIX 2	93 E5
31 SF 29	94 /
32 "EST("	95 2962
33 FS? 02	96 +
34 "†S"	97 INT
35 FC? 02	98 RTN
36 "†G"	99+LBL 03
37 "†T)= "	100 27
38 ARCL X	101 -
39 AVIEW	102 X<>Y
40 BEEP	103 130032
41 RTN	104 X<=Y?
42+LBL E	105 GTO 04
43+LBL "EST32a"	106 X<>Y
44 INT	107 ENTER†
45 RCL X	108 R†
46 54	109 -
47 MOD	110 E2
48 -	111 /
49 4536	112 42 E3
50 X<Y?	113 %
51 GTO 01	114 518 E4
52 CLX	115 +
53 RTN	116 *
54+LBL 01	117 E5
55 CLX	118 /
56 10035	119 29417
57 X<Y?	120 +
58 GTO 02	121 INT
59 CLX	122 RTN
60 22	123+LBL 04
61 %	124 CLX
62 998	125 56
63 -	126 %
	127 16433
	128 -
	129 INT
	130 END

01*LBL "EST88"	17 SF 02	33 FS? 02	58 GTO 02	83 -	99*LBL 03	115 +
02 1	18*LBL 00	34 "FS"	59 CLX	84 %	100 27	116 *
03 SF 02	19 ,5	35 FC? 02	60 22	85 INT	101 -	117 E5
04 *SPLIT?J:N N=0"	20 X<>Y	36 "FG"	61 %	86 452 E3	102 X<>Y	118 /
05 TONE 7	21 FS? 02	37 "FT)= "	62 990	87 +	103 130032	119 27790
06 PROMPT	22 *	38 ARCL X	63 -	88 %	104 X<=Y?	120 +
07 X=0?	23 XEQ E	39 AVIEW	64 INT	89 INT	105 GTO 04	121 INT
08 CF 02	24 FC? 02	40 BEEP	65 RTN	90 22 E5	106 X<>Y	122 RTN
09 "EINKUENFTE?="	25 GTO 00	41 RTN	66*LBL 02	91 +	107 ENTER↑	123*LBL 04
10 TONE 7	26 2	42*LBL E	67 35	92 *	108 R↑	124 CLX
11 PROMPT	27 ST+ Z	43*LBL "EST32a"	68 -	93 E5	109 -	125 56
12 GTO 00	28 *	44 INT	69 X<>Y	94 /	110 E2	126 %
13*LBL "ESTG"	29*LBL 00	45 RCL X	70 00027	95 2962	111 /	127 10502
14 CF 02	30 FIX 2	46 54	71 X<Y?	96 +	112 6 E4	128 -
15 GTO 00	31 SF 29	47 MOD	72 GTO 03	97 INT	113 %	129 INT
16*LBL "ESTS"	32 "EST("	48 -	73 X<>Y	98 RTN	114 5 E6	130 END
		49 4536	74 ENTER↑			
		50 X<Y?	75 R↑			
		51 GTO 01	76 -			
		52 CLX	77 E2			
		53 RTN	78 /			
		54*LBL 01	79 790			
		55 CLX	80 %			
		56 10035	81 INT			
		57 X<Y?	82 30020			

TV ROBOT

Wie ich zu meinem Bedauern feststellen mußte, waren in Heft 8/86 die Barcodes zu dem Programm TVROBOT nicht mit abgedruckt. Ihr findet sie am Ende dieses Heftes; bitte entschuldigt dieses elementare Versehen, da das Programm ja nicht zu listen war.

Martin Meyer (1000)

Ansonsten werden die Programme genauso bedient wie in Prisma 8/86 beschrieben.

Bitte entschuldige wenn ich Dich etwas verwirrt habe.

Ich gelobe Besserung.
Dein Söhnlein.

Hans Jürgen Hübner
Kranichsteiner Str. 4
6000 Frankfurt 70

R o b o t e r f a l l e

Roboter= R	9													
Fallen= @	8		@							@				
Android= A	7													
Androidbewegungen:	6								R					
	5		@											
1=runter & links	4													
2=runter	3								R					
3=runter & rechts	2		R											
4=links	1		R						A					
5=keine Bewegung	0				@			@						
6=rechts			0 1 2 3 4 5 6 7 8 9											
7=rauf & links														
8=rauf														
9=rauf & rechts														

EINGABE: <-

SPLINE

421 Zeilen, 611 Bytes, 88 Regs., SIZE 7+5n
HP 41C, 2 MM

YOFX

29 Zeilen, 45 Bytes, 7 Regs.,
Hp 41C

Ich bin zwar erst vor Kurzem (Nr 2969!!), mit dem CCD-Modul, zu Euch gestoßen, doch das weite Spektrum der vorgestellten Programme gab mir bereits gute Gelegenheit dieses Versäumnis lebhaft zu bedauern. Besonders interessierten mich im Prisma 7/85 die Splines und die Binomialkoeffizienten, weil ich vergleichbare eigene Programme besitze, die ich Euch heute gerne mitteilen würde:

1) SPLINE

Dieses Programm berechnet kubische Splines ohne Randbedingungen durch eine vorgegebene Stützstellenfolge. Daseinsberechtigung neben dem fulminanten Programm von Wilfried Mlaker hat es vor allem aufgrund seiner Kürze (es benötigt - einschließlich I/O - nur einen Bruchteil an Platz und Rechenzeit) und seiner Universalität (geschrieben in grauer Vorzeit, als es noch gar keine XF/M's geschweige denn 41 CX's gab, läuft es auf allen 41ern)

Zum Programm:

Benötigt werden 88 Prgm Register und 7+5mal (Stützstellenzahl) Data Register. Das Vorhandensein letzterer wird in Zeile 13 überprüft. Die (mindestens3) Stützstellen müssen streng monoton steigende X-Werte besitzen (wird beides nicht überprüft, führt später aber sonst zu DATA ERROR).

Verwendet werden die Flags 5 (Merkflag bei Eingabe korrigierter Punkte), 6 (Unterbrechung beim 'Plotten') und 7 (Einzelinterpolation if set). Die Speicherplätze 7+kn bis 7+(k+1)n enthalten für k=0,1,2,3,4 die Felder A,B,C,D respektive F des, für Interessierte im leichter lesbaren BASIC angegeben (allererster Uralt Fassung auf einem HP&!) Tischrechner von '76(!), Algorithmus.

Zur Bedienung:

XEQ'SPLINE (oder XEQ A) promptet die Anzahl der Stützstellen, die darauf per 'x-Wert', ENTER, 'y-Wert', R/S eingegeben werden. Im Anschluß an die Eingabe des letzten Punktes werden die Arbeitsfelder berechnet.

XEQ B wiederholt alle eingegebenen Punkte. Eingabefehler - oder Änderungen - können während der Anzeige des entsprechenden Punktes durch Eingabe von 'neuer X-Wert', ENTER, 'neuer Y-Wert', R/S korrigiert werden (worauf der korrigierte Punkt zur Kontrolle nochmals angezeigt wird). Falls nichts korrigiert wird / werden soll, kann (muß nicht) durch R/S unterbrochen werden; falls korrigiert wurde, werden die Arbeitsfelder nach Anzeige aller Punkte automatisch neu errechnet.

XEQ D 'plottet' den Interpolationsverlauf von START bis END in Schritten von STEP (größer 0). Eine Antwort von 1, R/S auf die Frage "Cont? 1:0" bewirkt eine kontinuierliche Anzeige der Funktion (z.B. zur Übersicht), eine Antwort von 0, R/S bewirkt ein Anhalten nach Berechnung des jeweils nächsten Wertepaares (z.B. zum Mitschreiben / -Zeichnen) (Vorteil des verzögerten Stops: nach restart (durch R/S) sofortige Anzeige des nächsten Paares). Auch hier kann unbeschadet unterbrochen werden.

XEQ E schließlich errechnet mit der Folge 'X-Wert', XEQ E (auf ganz R) den zum Eingabewert gehörenden Interpolationswert.

Literaturhinweis:

Späth, Spline Algorithmen, Oldenburg
Stoer, Numerische Mathematik, Springer

YOFX

verwendet fast den gleichen Algorithmus wie das Programm von Norbert Benz, ist im Gegensatz dazu aber numerisch korrekt (bis 10 hoch 10) und total, soweit darstellbar. (und ist kürzer und schneller - aber nur wenig und verwendet außerdem ausschließlich den Stack) Zum Programm:

Benötigt werden 7 Prgm Register. Die auf den ersten Blick vorhandene Einschränkung auf ABS(y-x) kleiner 70 ist in Wahrheit keine, da der kleinste Binomialkoeffizient bei der sie greift (70 aus 140) sowieso nicht mehr normal darstellbar ist.

Um auch so exotische Koeffizienten wie z.B. 2 aus 10 hoch 50 noch berechnen zu können, muß die Division vorgezogen werden, was wiederum eine treue Rundung des Ergebnisses nach sich zieht, um sicher im ganzzahligen Bereich zu bleiben. Mathematiker der alten Schule, die gewöhnt sind statt '6 aus 49' '49 über 6' zu sprechen, können Zeile 3 durch 'x>y' ersetzen (und den Programmnamen vielleicht durch YOYX wie 'y über X?') und vor Zeile 5 ein 'X<>Y' einfügen.

Zur Bedienung:

'k-Wert', ENTER, 'n-Wert', XEQ'YOFX (oder XEQ A) berechnet den Binomialkoeffizienten 'k aus n' ($\binom{n}{k}$) falls k kleiner gleich n und k, n aus 10^{100} kleiner ($\binom{n}{k}$ kleiner 10^{100}). Von diesen Zusicherungen wird nur erstere abgeprüft (in den Zeilen 3/4): sie entspricht dem häufigsten Eingabefehler (verwechseln von x und y). Zu große Eingaben führen zu Überlauf (sollte, weil völlig unsauber, nicht durch SF24 markiert werden), nichtnatürliche Eingaben zu total falschen Ergebnissen - wenn überhaupt.

Ich hoffe, Ihr könnt beide Programme nutzbringend einsetzen. Übrigens: Um Euch zu beweisen, daß ich nicht nur nachtarocken kann, schreibe ich Euch bald noch andere Programme aus meinem Interessensgebiet (und Studiennebenfach), der Numerik. Nachdem Prisma ja kaum Forum für Microde sein kann, möchte ich aber schon jetzt jeden MCODE-Numeriker anregen, sich mit mir direkt auszutauschen. Ich denke, es würde sich lohnen!

```
01*LBL "SPLINE" 19 7
02 CF 29        20 +
03*LBL A       21 STO 00
04 CLRG        22 1
05 "SP?"      23 STO 02
06 TONE 7     24 FIX 0
07 PROMPT     25*LBL 00
08 STO 01     26 "P"
09 5          27 ARCL 02
10 *          28 "f (X,↑,Y)"
11 6          29 TONE 7
12 +          30 PROMPT
13 RCL IND X  31 RCL 01
14 RCL 01     32 RCL 00
15 6          33 +
16 +          34 X<>Y
17 E3        35 STO IND Y
18 /         36 RCL Z
```

```
37 STO IND 00 100 RCL 01
38 1          101 2
39 ST+ 02    102 *
40 ISG 00    103 RCL 00
41 GTO 00    104 +
42 FIX 4     105 RCL X
43*LBL 20    106 1
44 CLX       107 -
45 STO 02    108 RCL IND X
46 RCL 01    109 RCL 03
47 E-3       110 *
48 +         111 ST+ IND Z
49 ST- 00    112 RCL 00
50*LBL 01    113 1
51 RCL 01    114 +
52 3         115 RCL 00
53 *         116 1
54 RCL 00    117 -
55 +         118 RCL 01
56 RCL 00    119 4
57 1         120 *
58 +         121 RCL 00
59 RCL IND X 122 +
60 RCL IND 00 123 RCL IND Y
61 -         124 RCL IND T
62 STO IND Z 125 -
63 RCL 01    126 2
64 ST+ Z     127 *
65 RCL 00    128 RCL 03
66 +         129 RCL 02
67 RCL IND Z 130 *
68 RCL IND Y 131 -
69 -         132 STO IND Y
70 RCL T     133 RCL 01
71 /         134 3
72 STO 03    135 *
73 RCL 02    136 RCL 00
74 -         137 +
75 RCL 01    138 RCL IND X
76 2         139 STO 02
77 *         140 RCL Z
78 RCL 00    141 /
79 +         142 STO 03
80 X<>Y     143 ISG 00
81 STO IND Y 144 GTO 02
82 RCL 03    145 RCL 01
83 STO 02    146 5,007
84 ISG 00    147 +
85 GTO 01    148 STO 00
86 RCL 01    149*LBL 03
87 2         150 RCL 01
88 *         151 3
89 7         152 *
90 +         153 RCL 00
91 0         154 +
92 STO 02    155 RCL 01
93 STO 03    156 2
94 STO IND Y 157 *
95 RCL 01    158 1
96 2         159 +
97 -         160 RCL 00
98 ST- 00    161 +
99*LBL 02    162 RCL X
```

163 1	226 RCL 00	289*LBL 06	352 +	415 ST+ 02
164 -	227 +	290 STO 03	353 RCL 04	416 ISG 00
165 RCL IND Z	228 RCL IND Z	291 RCL 01	354 *	417 GTO 12
166 RCL IND Z	229 RCL IND Z	292 5	355 RCL 01	418 CLD
167 *	230 -	293 +	356 RCL 00	419 FS? 05
168 RCL IND Y	231 RCL 02	294 X<>Y	357 +	420 GTO 20
169 -	232 /	295 RCL IND Y	358 X<>Y	421 .END.
170 STO IND Y	233 RCL IND Y	296 X>Y?	359 RCL IND Y	
171 RCL 01	234 3	297 GTO 07	360 +	
172 4	235 *	298 -	361 "P("	
173 *	236 STO IND Z	299 STO 04	362 ARCL 03	01*LBL "YOFX"
174 RCL 00	237 RCL 03	300 RCL 01	363 "+, "	02*LBL A
175 +	238 +	301 5	364 ARCL X	03 X<Y?
176 RCL IND X	239 RCL 02	302 +	365 "+)"	04 SF 99
177 ST/ IND T	240 *	303 STO 00	366 FS?C 07	05 ENTER↑
178 DSE 00	241 -	304 GTO 10	367 GTO 11	06 RDN
179 GTO 03	242 RCL 01	305*LBL 07	368 FS? 06	07 X<>Y
180 RCL 01	243 4	306 CLX	369 STOP	08 -
181 2	244 *	307 RCL 07	370*LBL 11	09 LASTX
182 -	245 RCL 00	308 X<=Y?	371 AVIEW	10 X>Y?
183 E3	246 +	309 GTO 08	372 RTN	11 X<>Y
184 /	247 X<>Y	310 -	373*LBL B	12 STO Y
185 ST+ 00	248 STO IND Y	311 STO 04	374 RCL 01	13 FACT
186*LBL 04	249 ISG 00	312 7	375 6	14 X<>Y
187 RCL 01	250 GTO 04	313 STO 00	376 +	15 R↑
188 3	251 0	314 GTO 10	377 E3	16 SIGN
189 *	252 TONE 5	315*LBL 08	378 /	17*LBL 01
190 RCL 00	253 RTN	316 6	379 7	18 LASTX
191 +	254*LBL D	317 STO 05	380 +	19 *
192 RCL IND X	255 SF 07	318*LBL 09	381 STO 00	20 DSE L
193 STO 02	256 "START?"	319 RCL 05	382 1	21 DSE Y
194 RCL 01	257 TONE 7	320 STO 00	383 STO 02	22 GTO 01
195 2	258 PROMPT	321 1	384 CF 05	23 RCL Z
196 *	259 STO 03	322 +	385 CF 22	24 /
197 RCL 00	260 "END?"	323 STO 05	386*LBL 12	25 ENTER↑
198 +	261 TONE 7	324 RCL 03	387 "P"	26 FRC
199 RCL X	262 PROMPT	325 RCL IND 05	388 RCL d	27 +
200 1	263 STO 06	326 X<=Y?	389 FIX 0	28 INT
201 +	264 "STEP?"	327 GTO 09	390 ARCL 02	29 END
202 RCL IND X	265 TONE 7	328 CLX	391 STO d	
203 RCL IND Z	266 PROMPT	329 RCL IND 00	392 "+ ("	
204 -	267 STO 02	330 -	393 ARCL IND 00	
205 STO 03	268 CF 06	331 STO 04	394 "+, "	
206 RCL 02	269 "CONT? 1:0"	332*LBL 10	395 RCL 01	
207 /	270 CLX	333 RCL 01	396 RCL 00	
208 RCL 01	271 TONE 7	334 2	397 +	
209 3	272 PROMPT	335 *	398 ARCL IND X	
210 *	273 X=0?	336 RCL 00	399 "+)"	
211 RCL 00	274 SF 06	337 +	400 AVIEW	
212 +	275*LBL 05	338 RCL X	401 PSE	
213 X<>Y	276 RCL 03	339 RCL 01	402 FC?C 22	
214 STO IND Y	277 XEQ 06	340 +	403 GTO 13	
215 RCL 01	278 RCL 06	341 RCL X	404 RCL 01	
216 1	279 RCL 03	342 RCL 01	405 RCL 00	
217 +	280 RCL 02	343 +	406 +	
218 RCL 00	281 +	344 RCL IND Z	407 X<>Y	
219 +	282 STO 03	345 RCL 04	408 STO IND Y	
220 RCL X	283 X<=Y?	346 RCL IND T	409 RCL Z	
221 1	284 GTO 05	347 *	410 STO IND 00	
222 -	285 TONE 5	348 +	411 SF 05	
223 RCL 01	286 RTN	349 RCL 04	412 GTO 12	
224 2	287*LBL E	350 *	413*LBL 13	
225 *	288 SF 07	351 RCL IND Y	414 1	

```

10 REM SPLINE INTERPOLATION FUER N STUETZSTELLEN
20 DISP "WIEVIELE STUETZSTELLEN";
30 INPUT N
40 IF (N-INT(N))<0 OR ABS(N-13)>12 THEN 550
50 DIM Aä25ö,Bä25ö,Cä25ö,Dä25ö,Eä25ö,Fä25ö
60 FOR I=1 TO N
70 DISP "EINGABE DES" I ".STUETZPUNKTES!";
80 INPUT AäIö,BäIö
90 IF I=1 THEN 110
100 IF AäIö <= AäI-1ö THEN 580
110 NEXT I

```

```

120 S=0
130 FOR I=1 TO (N-1)
140 DäIö=AäI+1ö-AäIö
150 R=(BäI+1ö-BäIö)/DäIö
160 CäIö=R-S
170 S=R
180 NEXT I
190 S=R-CäIö=Cänö=0
200 FOR I=2 TO (N-1)
210 CäIö=CäIö+R*CäI-1ö
220 FäIö=(AäI-1ö-AäI+1ö)*2-R*S
230 S=DäIö
240 R=S/FäIö
250 NEXT I
260 FOR I=(N-1) TO 2 STEP -1
270 CäIö=(DäIö*CäI+1ö-CäIö)/FäIö
280 NEXT I
290 FOR I=1 TO (N-1)
300 S=DäIö
310 R=CäI+1ö-CäIö
320 DäIö=R/S
330 CäIö=CäIö*3
340 FäIö=(BäI+1ö-BäIö)/S-(CäIö+R)*S
350 NEXT I

```

Bem.: $\begin{matrix} \text{ä} & \hat{=} & \neq \\ \text{ä} & \hat{=} & [\\ \text{ö} & \hat{=} &] \end{matrix}$

```

360 DISP "EINGABE DER INTERPOLATIONSSTELLE";
370 INPUT Aän+1ö
380 IF (Aän+1ö-Aän-1ö)<0 THEN 420
390 Q=Aän+1ö-Aän-1ö
400 I=N-1
410 GOTO 510
420 IF (Aän+1ö-Aä1ö) >= 0 THEN 460
430 Q=Aän+1ö-Aä1ö
440 I=1
450 GOTO 510
460 K=0
470 I=K
480 K=K+1
490 IF (Aän+1ö-AäKö) >= 0 THEN 470
500 Q=Aän+1ö-AäIö
510 S=((DäIö*Q+CäIö)*Q+FäIö)*Q+BäIö
520 WRITE (15,1000)Aän+1ö,S
530 WAIT 1000
540 GOTO 360
550 DISP "UNZULAESSIG!"
560 WAIT 2000
570 GOTO 20
580 DISP "UNZULAESSIGE STUETZSTELLENFOLGE!"
590 WAIT 2000
600 GOTO 70
1000 FORMAT F10.2,F12.2

```

Christian Martini
Am Reichswald 6
8503 Altdorf

GRAFIK auf ThinkJet mit dem HP 41

Die Graphik können wir nun mit dem PRAF-XIO Programm verhältnismäßig schnell - ca. 40 Sekunden für das Beispiel - ausdrucken.

CCD

Bedienung FILE-Name ins Alpha Register und starten.

Nun habe ich das gleiche im Alternativmodus also mit Spaltengraphik aufgebaut. Hierbei ergeben sich 4 Reihen je 72 Grafikbytes. Die Bytewerte müssen wieder berechnet werden. Da im Alternativmodus CR und LF explizit gesendet werden muß, stellen wir zunächst den Zeilenabstand 8 Punkte pro Zeile ein.

Sequenz: E_CA(8)

Die erste Grafikreihe erhält dann die Sequenz E_CK(72)(0)(15)(63)(63)(127) usw.

nach jeder Grafikreihe müssen wir eine CR/LF, Bytes (13)(10) als Record abspeichern.

Der Ausdruck erfolgt dann mit dem gleichen Programm PRAF-XIO. Jetzt braucht der Rechner nur noch ca. 20 Sekunden.

Noch schneller kann man den Ausdruck erzeugen, wenn man den ASCII-File mit SAVEAS auf Band speichert und die Graphikbytes direkt vom Band auf den Drucker sendet. Hierzu dient das folgende Programm.

FERDAS

Hans Jürgen

Für einen Briefkopf oder zur Illustration eines Berechnungsformulars ist es wünschenswert, eine Grafik mit auszudrucken. Da häufig Nullbytes für Grafiken verwendet werden müssen und der etwas komplizierten Ausgabe von Nullbytes mit dem IL-Modul ist die Beschaffung des X-I/O-Moduls dringend anzuraten.

Anhand des CCD-Signets soll hier die Verfahrensweise geschildert werden.

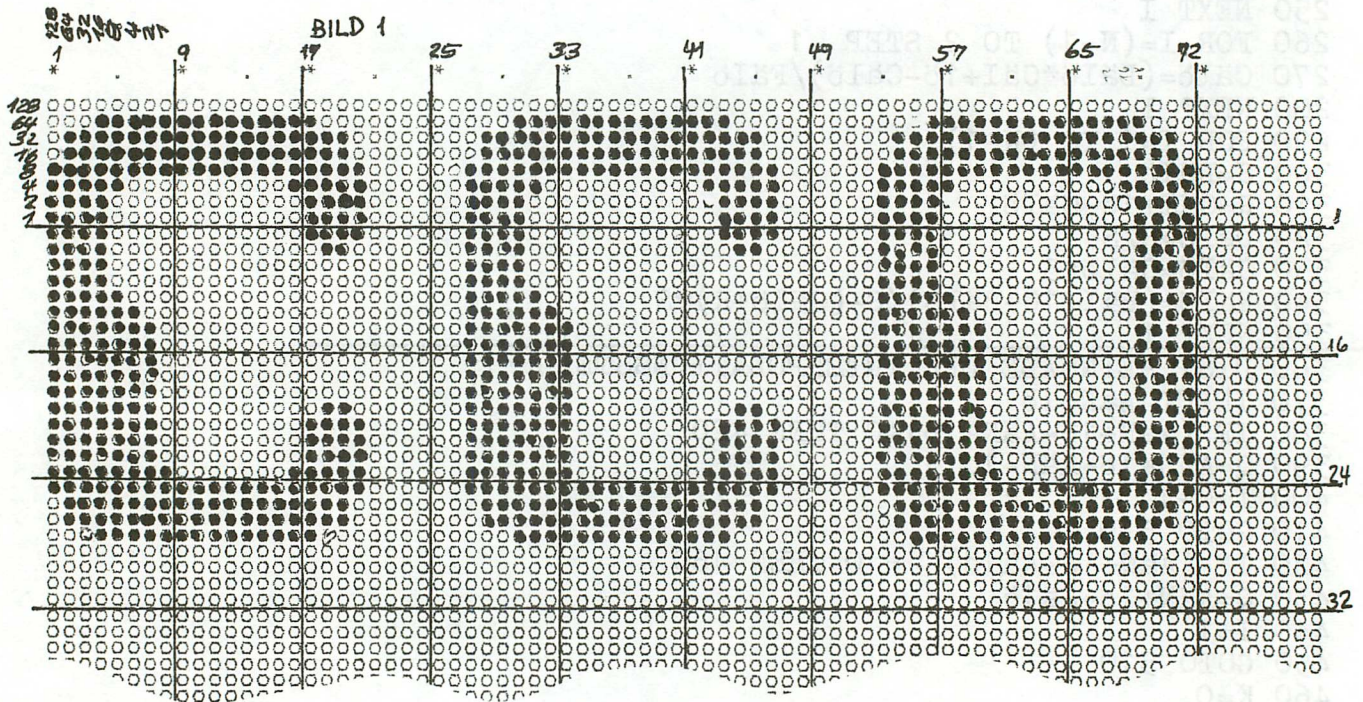
Zunächst benötigen wir ein Rasterformular auf dem wir unsere gewünschte Grafik zeichnen.

Jede Grafiksequenz legen wir in einem X-Memory ASCII-File als Record ab. Es ergeben sich hier 32 Zeilen a 72 Punkte also 72:8=9 Graphikbytes pro Zeile.

Nun müssen die Byte-Werte berechnet werden. Die erste Reihe enthält keine Punkte und somit wäre die zu sendende Sequenz E_C*b1W(Ø) (Die Sequenz E_C*bØW könnte einen hier unerwünschten Punktzeilenvor...bewirken)

Für die zweite Zeile ergeben sich 9 Graphikbytes mit den Werten 31, 255, 128, 7, 255, 224, 1, 255, 248. Die zu sendende Sequenz wäre also:

E_C*b9W-
(31)(255)(128)(7)(255)(224)(1)(255)(248)
usw.



Ausdruck von ASCII Files des X-Memory

PRAF-XIO

Print ASCII File mit X-I/O

25 Zeilen, 53 Bytes, 8 Regs., SIZE 000
HP 41C, X-F/M, TIME, X-I/O, IL, Printer

Das Programm gibt den Inhalt von ASCII-Files des X-Memory recordweise aus. Es werden auch Null-Bytes gesendet.

Bedienung:

FILE-Name ins Alpharegister schreiben und Programm starten. Der Filename bleibt erhalten, ebenso Y, Z und T. Für andere IL-Druk-

kertypen als den ThinkJet muß Zeile 2 entsprechend geändert werden.

```
01*LBL "PRAF-XIO"  11 "D"
02 35              12*LBL 01
03 FINDAID        13 OUTAN
04 SELECT         14 FC?C 17
05 CLX            15 ADV
06 SEEKPTA       16 SF 25
07 CLX            17 "D"
08 RCL I          18 ARCLREC
09 SIGN           19 FS?C 25
10 CLX            20 GTO 01
```

```
21 CLA
22 X<> L
23 STO I
24 CLX
25 END
```

Hans Jürgen Hübner (1990)
Kranichsteiner Str. 4
6000 Frankfurt 70

Schneller Datentransfer von ASCII-Files

FERAS

Transfer ASCII-File

105 Zeilen, 192 Bytes, 28 Regs., SIZE 003
HP 41C, X-F/M, X-I/O, IL, DRIVE, Printer

Für die Gestaltung eines persönlichen Computerausdruckes möchte man gerne Grafik und Text mischen. Hierfür wird verhältnismäßig viel Programmspeicherplatz benötigt und viel Zeit um das Programm abzuarbeiten.

Die Datensequenzen als ASCII-File aus dem X-Memory zu senden, benötigt ebenfalls sehr viel Zeit.

Im X-I/O-Modul haben wir jedoch die Funktion XFERN. zur Verfügung. Benutzen wir also den Massenspeicher zur Ablage der Graphiksequenz in der Form wie sie der Drucker benötigt und transferieren sie bei Bedarf mit dem Befehl XFERN. Hierfür dient das folgende Programm.

Bedienung:

Filename ins Alpharegister schreiben und Programm starten. Der Filename bleibt erhalten. Für andere Drucker als den ThinkJet bzw. andere Ausgabeinheiten muß Zeile 2 entsprechend und evtl. Zeile 93 geändert werden.

```

01*LBL "FERAS"      38 OUTXB
02 35                39 RCL 01
03 FINDAID          40 TAD
04 SELECT            41 E
05 STO 02            42 DDT
06 16                43 2
07 FINDAID          44 INAN
08 SELECT            45 RCL 01
09 STO 01            46 LAD
10 RCL I              47 4
11 "AS"              48 DDL
12 ASTO T            49 2
13 STO I              50 OUTAN
14 SIGN              51 LASTX
15 R↑                 52 STO I
16 FLTYPE            53*LBL 05
17 X*Y?              54 INSTAT
18 GETAS              55 FS? 05
19 MANIO              56 GTO 05
20 ADROFF             57 RCL 01
21 RCL 01             58 TAD
22 TAD                59 2
23 3                  60 DDT
24 DDT                61 AUTOIO
25 INAN               62 ADRON
26 CLX                63 RCL 02
27 256                64 LAD
28 ATOXR              65 FC? 04
29 X=0?               66 GTO 06
30 X<>Y               67 LASTX
31 18                 68 X<>Y
32 ST- Y              69 "DRIVE ERR"
33 R↑                 70 GTO 09
34 LAD                71*LBL 06
35 3                  72 INXB
36 DDL                73 X=0?
37 R↑                 74 GTO 07
    
```

```

75 CLX                91 X<>Y
76 RCL I              92 XFERN
77 "END OF FILE"     93 ADV
78 X<>Y               94 GTO 06
79*LBL 09             95*LBL 10
80 UNL                96 CF 21
81 RCL 01             97 AVIEW
82 UNT                98 SF 21
83 GTO 10             99 CLA
84*LBL 07             100 R↑
85 CLX                101 R↑
86 INXB              102 STO I
87 RCL X              103 CLST
88 2                  104 BEEP
89 MOD                105 END
90 +
    
```

Hans Jürgen Hübner

Laufwerk-Hilfsprogramme (Drive Utilities)

AS - DA

Write ASCII into Datafile

60 Zeilen, 108 Bytes, 16 Regs., SIZE 001
HP 41C, X-F/M, X-I/O, IL, DRIVE

Das Programm schreibt den X-M/F ASCII-Arbeitsfile in einem Spezialformat auf das Laufwerk. Dieses Format ermöglicht später eine sehr schnelle Ausgabe auf einen Drucker oder ein anderes IL-Gerät mit dem Programm XFERDA.

Bedienung:

Den zu schreibenden ASCII-File zum Arbeitsfile machen (z.B. mit SEEKPTA) Filename für das Laufwerk in das Alpharegister eingeben. Programm starten. Nach Ablauf des Programmes steht die Zahl der übertragenen Zeichen im X-Register, der Filename bleibt im Alpha-Register erhalten R00 wird restauriert.

```

01*LBL "AS-DA"      23 "D"
02 16                24 SF 17
03 FINDAID          25*LBL 01
04 SELECT            26 ALENGIO
05 RCL I              27 DSE X
06 CLA                28 OUTAN
07 FLSIZE            29 ST+ 00
08 ISG X              30 FS?C 17
09 ""                 31 GTO 02
10 X<>Y               32 2
11 STO I              33 "D"
12 SIGN              34 OUTACL
13 RDN                35 ST+ 00
14 CREATE             36*LBL 02
15 CLX                37 SF 25
16 SEEKR              38 "D"
17 SEEKPT             39 ARCLREC
18 WRTRX              40 FS?C 25
19 STO 00             41 GTO 01
20 MANIO              42 8
21 6                  43 DEVL
22 DEVL               44 CLA
    
```

```

45 LASTX              53 X<> 00
46 STO I              54 ,
47 CLX                55 SEEKR
48 SEEKR              56 WRTRX
49 RCL 00             57 X<>Y
50 X<>Y               58 X<> 00
51 READRX             59 AUTOIO
52 X<>Y               60 END
    
```

Hans Jürgen Hübner

Laufwerk/Printer Hilfsprogramme (Drive/Printer Utilities)

XFERDA

Transfer Datafile

19 Zeilen, 42 Bytes, 6 Regs., SIZE 001
HP 41C, X-I/O, IL, DRIVE, PRINTER

Das Programm überträgt den Inhalt eines Spezialdatenfiles vom Laufwerk direkt auf den (hier ThinkJet) Drucker. Der Datafile muß im entsprechenden Format auf dem Laufwerk vorhanden sein.

Bedienung:

Den Filenamen des zu übertragenden Files ins Alpha-Register schreiben. Programm starten. Nach Ablauf des Programms steht die Zahl der übertragenen Zeichen im X-Register, Alpha bleibt erhalten. R00 wird restauriert. Für andere Ausgabeinheiten muß Zeile 03 entsprechend geändert werden.

```

01*LBL "XFERDA"      11 READRX
02 RCL 00             12 DEVT
03 35                 13 R↑
04 FINDAID            14 X<> 00
05 SELECT             15 R↑
06 16                 16 XFERN
07 FINDAID            17 R↑
08 SELECT             18 X<> Z
09 ,                  19 END
10 SEEKR
    
```

Hans-Jürgen Hübner

POL (Schilli)

Benötigte Programmregister: 41

Zeile 1	(1-5)	CCD-Barcodes
Zeile 2	(6-7)	CCD-Barcodes
Zeile 3	(8-17)	CCD-Barcodes
Zeile 4	(17-21)	CCD-Barcodes
Zeile 5	(22-31)	CCD-Barcodes
Zeile 6	(31-39)	CCD-Barcodes
Zeile 7	(39-47)	CCD-Barcodes
Zeile 8	(48-55)	CCD-Barcodes
Zeile 9	(55-64)	CCD-Barcodes
Zeile 10	(65-73)	CCD-Barcodes
Zeile 11	(74-84)	CCD-Barcodes
Zeile 12	(85-92)	CCD-Barcodes
Zeile 13	(93-102)	CCD-Barcodes
Zeile 14	(102-110)	CCD-Barcodes
Zeile 15	(111-115)	CCD-Barcodes
Zeile 16	(115-117)	CCD-Barcodes
Zeile 17	(117-127)	CCD-Barcodes
Zeile 18	(128-138)	CCD-Barcodes
Zeile 19	(139-145)	CCD-Barcodes
Zeile 20	(146-152)	CCD-Barcodes
Zeile 21	(153-159)	CCD-Barcodes
Zeile 22	(160-163)	CCD-Barcodes

Zeile 8	(46-55)	CCD-Barcodes
Zeile 9	(55-58)	CCD-Barcodes
Zeile 10	(58-61)	CCD-Barcodes
Zeile 11	(61-67)	CCD-Barcodes
Zeile 12	(67-70)	CCD-Barcodes
Zeile 13	(70-73)	CCD-Barcodes
Zeile 14	(73-84)	CCD-Barcodes
Zeile 15	(85)	CCD-Barcodes

CZBL (Triesberger)

Benötigte Programmregister: 38

Zeile 1	(1-6)	CCD-Barcodes
Zeile 2	(6-16)	CCD-Barcodes
Zeile 3	(17-26)	CCD-Barcodes
Zeile 4	(26-34)	CCD-Barcodes
Zeile 5	(35-44)	CCD-Barcodes
Zeile 6	(45-54)	CCD-Barcodes
Zeile 7	(55-62)	CCD-Barcodes
Zeile 8	(62-69)	CCD-Barcodes
Zeile 9	(69-77)	CCD-Barcodes
Zeile 10	(78-88)	CCD-Barcodes
Zeile 11	(89-98)	CCD-Barcodes
Zeile 12	(99-103)	CCD-Barcodes
Zeile 13	(103-108)	CCD-Barcodes
Zeile 14	(108-115)	CCD-Barcodes
Zeile 15	(116-126)	CCD-Barcodes
Zeile 16	(126-135)	CCD-Barcodes
Zeile 17	(135-142)	CCD-Barcodes
Zeile 18	(143-151)	CCD-Barcodes
Zeile 19	(152-161)	CCD-Barcodes
Zeile 20	(162-172)	CCD-Barcodes
Zeile 21	(172-175)	CCD-Barcodes

MEM (Schilli)

Benötigte Programmregister: 27

Zeile 1	(1-5)	CCD-Barcodes
Zeile 2	(5-11)	CCD-Barcodes
Zeile 3	(12-22)	CCD-Barcodes
Zeile 4	(22-29)	CCD-Barcodes
Zeile 5	(30-35)	CCD-Barcodes
Zeile 6	(35-38)	CCD-Barcodes
Zeile 7	(39-45)	CCD-Barcodes

HF (Schneider)

Benötigte Programmregister: 41

Zeile 1	(1-6)	CCD-Barcodes
Zeile 2	(6-13)	CCD-Barcodes
Zeile 3	(14-22)	CCD-Barcodes
Zeile 4	(23-32)	CCD-Barcodes
Zeile 5	(33-40)	CCD-Barcodes
Zeile 6	(40-44)	CCD-Barcodes
Zeile 7	(44-51)	CCD-Barcodes
Zeile 8	(52-61)	CCD-Barcodes
Zeile 9	(61-70)	CCD-Barcodes
Zeile 10	(71-75)	CCD-Barcodes
Zeile 11	(76-83)	CCD-Barcodes
Zeile 12	(83-92)	CCD-Barcodes
Zeile 13	(92-101)	CCD-Barcodes
Zeile 14	(102-106)	CCD-Barcodes
Zeile 15	(107-114)	CCD-Barcodes
Zeile 16	(114-122)	CCD-Barcodes
Zeile 17	(123-132)	CCD-Barcodes
Zeile 18	(132-139)	CCD-Barcodes
Zeile 19	(140-151)	CCD-Barcodes
Zeile 20	(151-157)	CCD-Barcodes
Zeile 21	(158-167)	CCD-Barcodes
Zeile 22	(167-171)	CCD-Barcodes

M? (Neubacher)

Benötigte Programmregister: 16

Zeile 1	(1-5)	CCD-Barcodes
Zeile 2	(6-15)	CCD-Barcodes
Zeile 3	(15-21)	CCD-Barcodes
Zeile 4	(22-28)	CCD-Barcodes
Zeile 5	(28-37)	CCD-Barcodes
Zeile 6	(38-43)	CCD-Barcodes
Zeile 7	(43-50)	CCD-Barcodes

Zeile 8	(51-58)	CCD-Barcodes
Zeile 9	(59-61)	CCD-Barcodes

EST86 (Hübner)

Benötigte Programmregister: 42

Zeile 1	(1-4)	CCD-Barcodes
Zeile 2	(4)	CCD-Barcodes
Zeile 3	(5-9)	CCD-Barcodes
Zeile 4	(9-13)	CCD-Barcodes
Zeile 5	(13-16)	CCD-Barcodes
Zeile 6	(16-21)	CCD-Barcodes
Zeile 7	(22-29)	CCD-Barcodes
Zeile 8	(30-34)	CCD-Barcodes
Zeile 9	(34-38)	CCD-Barcodes
Zeile 10	(38-43)	CCD-Barcodes
Zeile 11	(43-49)	CCD-Barcodes
Zeile 12	(49-56)	CCD-Barcodes
Zeile 13	(57-65)	CCD-Barcodes
Zeile 14	(66-72)	CCD-Barcodes
Zeile 15	(73-81)	CCD-Barcodes
Zeile 16	(82-88)	CCD-Barcodes
Zeile 17	(89-95)	CCD-Barcodes
Zeile 18	(95-103)	CCD-Barcodes
Zeile 19	(103-112)	CCD-Barcodes
Zeile 20	(112-117)	CCD-Barcodes
Zeile 21	(118-125)	CCD-Barcodes
Zeile 22	(126-134)	CCD-Barcodes
Zeile 23	(135)	CCD-Barcodes

EST88 (Hübner)

Benötigte Programmregister: 41

Zeile 1	(1-4)	CCD-Barcodes
Zeile 2	(4)	CCD-Barcodes
Zeile 3	(5-9)	CCD-Barcodes

Zeile 4	(9-13)	CCD-Barcodes
Zeile 5	(13-16)	CCD-Barcodes
Zeile 6	(16-21)	CCD-Barcodes
Zeile 7	(22-29)	CCD-Barcodes
Zeile 8	(30-34)	CCD-Barcodes
Zeile 9	(34-38)	CCD-Barcodes
Zeile 10	(38-43)	CCD-Barcodes
Zeile 11	(43-49)	CCD-Barcodes
Zeile 12	(49-56)	CCD-Barcodes
Zeile 13	(57-65)	CCD-Barcodes
Zeile 14	(66-72)	CCD-Barcodes
Zeile 15	(73-82)	CCD-Barcodes
Zeile 16	(82-87)	CCD-Barcodes
Zeile 17	(88-95)	CCD-Barcodes
Zeile 18	(95-103)	CCD-Barcodes
Zeile 19	(103-111)	CCD-Barcodes
Zeile 20	(112-119)	CCD-Barcodes
Zeile 21	(119-127)	CCD-Barcodes
Zeile 22	(127-130)	CCD-Barcodes

Zeile 13	(40-41)	CCD-Barcodes
Zeile 14	(41-46)	CCD-Barcodes
Zeile 15	(47-52)	CCD-Barcodes
Zeile 16	(53-55)	CCD-Barcodes
Zeile 17	(55-59)	CCD-Barcodes
Zeile 18	(60-63)	CCD-Barcodes
Zeile 19	(63-66)	CCD-Barcodes
Zeile 20	(66-68)	CCD-Barcodes
Zeile 21	(68-71)	CCD-Barcodes
Zeile 22	(71-75)	CCD-Barcodes
Zeile 23	(75)	CCD-Barcodes
Zeile 24	(76-80)	CCD-Barcodes
Zeile 25	(81-82)	CCD-Barcodes
Zeile 26	(82-86)	CCD-Barcodes
Zeile 27	(86-89)	CCD-Barcodes
Zeile 28	(89-91)	CCD-Barcodes
Zeile 29	(91-94)	CCD-Barcodes
Zeile 30	(94-97)	CCD-Barcodes
Zeile 31	(98-100)	CCD-Barcodes
Zeile 32	(100-103)	CCD-Barcodes
Zeile 33	(103-106)	CCD-Barcodes
Zeile 34	(106-109)	CCD-Barcodes
Zeile 35	(109-112)	CCD-Barcodes
Zeile 36	(113-116)	CCD-Barcodes
Zeile 37	(117-118)	CCD-Barcodes
Zeile 38	(118-122)	CCD-Barcodes
Zeile 39	(122)	CCD-Barcodes
Zeile 40	(122-129)	CCD-Barcodes
Zeile 41	(129-133)	CCD-Barcodes
Zeile 42	(133-134)	CCD-Barcodes
Zeile 43	(134-138)	CCD-Barcodes
Zeile 44	(138-147)	CCD-Barcodes
Zeile 45	(147-154)	CCD-Barcodes

TVROBOT (Meyer)

Benötigte Programmregister: 180

Zeile 1	(1-2)	CCD-Barcodes
Zeile 2	(3-5)	CCD-Barcodes
Zeile 3	(6-9)	CCD-Barcodes
Zeile 4	(9-11)	CCD-Barcodes
Zeile 5	(11-12)	CCD-Barcodes
Zeile 6	(13-16)	CCD-Barcodes
Zeile 7	(16-21)	CCD-Barcodes
Zeile 8	(21-24)	CCD-Barcodes
Zeile 9	(24-27)	CCD-Barcodes
Zeile 10	(27)	CCD-Barcodes
Zeile 11	(28-33)	CCD-Barcodes
Zeile 12	(33-40)	CCD-Barcodes

Zeile 46 (154-161) **CCD-Barcodes**
 Zeile 47 (162-169) **CCD-Barcodes**
 Zeile 48 (169-177) **CCD-Barcodes**
 Zeile 49 (178-185) **CCD-Barcodes**
 Zeile 50 (185-192) **CCD-Barcodes**
 Zeile 51 (193-201) **CCD-Barcodes**
 Zeile 52 (202-210) **CCD-Barcodes**
 Zeile 53 (210-214) **CCD-Barcodes**
 Zeile 54 (214-221) **CCD-Barcodes**
 Zeile 55 (221-226) **CCD-Barcodes**
 Zeile 56 (227-230) **CCD-Barcodes**
 Zeile 57 (230-231) **CCD-Barcodes**
 Zeile 58 (231-237) **CCD-Barcodes**
 Zeile 59 (237-245) **CCD-Barcodes**
 Zeile 60 (245-252) **CCD-Barcodes**
 Zeile 61 (252-257) **CCD-Barcodes**
 Zeile 62 (258-263) **CCD-Barcodes**
 Zeile 63 (264-271) **CCD-Barcodes**
 Zeile 64 (271-274) **CCD-Barcodes**
 Zeile 65 (274-275) **CCD-Barcodes**
 Zeile 66 (275-281) **CCD-Barcodes**
 Zeile 67 (281-288) **CCD-Barcodes**
 Zeile 68 (288-294) **CCD-Barcodes**
 Zeile 69 (294-302) **CCD-Barcodes**
 Zeile 70 (303-310) **CCD-Barcodes**
 Zeile 71 (311-318) **CCD-Barcodes**
 Zeile 72 (318-324) **CCD-Barcodes**
 Zeile 73 (325-330) **CCD-Barcodes**
 Zeile 74 (330-340) **CCD-Barcodes**
 Zeile 75 (341-351) **CCD-Barcodes**
 Zeile 76 (352-360) **CCD-Barcodes**
 Zeile 77 (360-366) **CCD-Barcodes**
 Zeile 78 (367-368) **CCD-Barcodes**

Zeile 79 (368-370) **CCD-Barcodes**
 Zeile 80 (370-373) **CCD-Barcodes**
 Zeile 81 (373-377) **CCD-Barcodes**
 Zeile 82 (378-383) **CCD-Barcodes**
 Zeile 83 (383-385) **CCD-Barcodes**
 Zeile 84 (385-388) **CCD-Barcodes**
 Zeile 85 (388) **CCD-Barcodes**
 Zeile 86 (388-393) **CCD-Barcodes**
 Zeile 87 (393-399) **CCD-Barcodes**
 Zeile 88 (399-402) **CCD-Barcodes**
 Zeile 89 (402-405) **CCD-Barcodes**
 Zeile 90 (405-409) **CCD-Barcodes**
 Zeile 91 (410-418) **CCD-Barcodes**
 Zeile 92 (418-427) **CCD-Barcodes**
 Zeile 93 (428-436) **CCD-Barcodes**
 Zeile 94 (436-441) **CCD-Barcodes**
 Zeile 95 (442-450) **CCD-Barcodes**
 Zeile 96 (451-455) **CCD-Barcodes**
 Zeile 97 (455-461) **CCD-Barcodes**

SPLINE (Martini)

Benötigte Programmregister: 88

Zeile 1 (1-3) **CCD-Barcodes**
 Zeile 2 (3-11) **CCD-Barcodes**
 Zeile 3 (12-22) **CCD-Barcodes**
 Zeile 4 (23-28) **CCD-Barcodes**
 Zeile 5 (28-35) **CCD-Barcodes**
 Zeile 6 (35-42) **CCD-Barcodes**
 Zeile 7 (42-50) **CCD-Barcodes**
 Zeile 8 (51-61) **CCD-Barcodes**
 Zeile 9 (62-70) **CCD-Barcodes**
 Zeile 10 (70-81) **CCD-Barcodes**
 Zeile 11 (82-92) **CCD-Barcodes**
 Zeile 12 (93-103) **CCD-Barcodes**

Zeile 13 (104-113) CCD-Barcodes



Zeile 14 (114-124) CCD-Barcodes



Zeile 15 (125-136) CCD-Barcodes



Zeile 16 (137-145) CCD-Barcodes



Zeile 17 (146-154) CCD-Barcodes



Zeile 18 (155-165) CCD-Barcodes



Zeile 19 (166-175) CCD-Barcodes



Zeile 20 (176-183) CCD-Barcodes



Zeile 21 (184-194) CCD-Barcodes



Zeile 22 (195-204) CCD-Barcodes



Zeile 23 (205-216) CCD-Barcodes



Zeile 24 (217-228) CCD-Barcodes



Zeile 25 (228-237) CCD-Barcodes



Zeile 26 (238-249) CCD-Barcodes



Zeile 27 (249-256) CCD-Barcodes



Zeile 28 (256-260) CCD-Barcodes



Zeile 29 (260-265) CCD-Barcodes



Zeile 30 (266-269) CCD-Barcodes



Zeile 31 (269-277) CCD-Barcodes



Zeile 32 (278-287) CCD-Barcodes



Zeile 33 (288-297) CCD-Barcodes



Zeile 34 (298-309) CCD-Barcodes



Zeile 35 (309-320) CCD-Barcodes



Zeile 36 (321-330) CCD-Barcodes



Zeile 37 (331-341) CCD-Barcodes



Zeile 38 (342-351) CCD-Barcodes



Zeile 39 (352-361) CCD-Barcodes



Zeile 40 (362-366) CCD-Barcodes



Zeile 41 (367-376) CCD-Barcodes



Zeile 42 (377-386) CCD-Barcodes



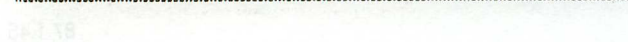
Zeile 43 (387-392) CCD-Barcodes



Zeile 44 (392-399) CCD-Barcodes



Zeile 45 (399-408) CCD-Barcodes



Zeile 46 (408-415) CCD-Barcodes



Zeile 47 (416-421) CCD-Barcodes



YOFX (????)

Benötigte Programmregister: 7

Zeile 1 (1-4) CCD-Barcodes



Zeile 2 (5-16) CCD-Barcodes



Zeile 3 (17-25) CCD-Barcodes



Zeile 4 (26-29) CCD-Barcodes



PRAFIO (Hübner)

Benötigte Programmregister: 8

Zeile 1 (1-2) CCD-Barcodes



Zeile 2 (3-11) CCD-Barcodes



Zeile 3 (11-18) CCD-Barcodes



Zeile 4 (19-25) CCD-Barcodes



FERAS (Hübner)

Benötigte Programmregister: 28

Zeile 1 (1-3) CCD-Barcodes



Zeile 2 (4-11) CCD-Barcodes



Zeile 3 (11-18) CCD-Barcodes



Zeile 4 (19-26) CCD-Barcodes



Zeile 5 (27-34) CCD-Barcodes



Zeile 6 (34-42) CCD-Barcodes



Zeile 7 (43-51) CCD-Barcodes



Zeile 8 (52-59) CCD-Barcodes



Zeile 9 (60-66) CCD-Barcodes



Zeile 10 (67-70) CCD-Barcodes



Zeile 11 (70-77) CCD-Barcodes



Zeile 12 (77-80) CCD-Barcodes



Zeile 13 (81-89) CCD-Barcodes



Zeile 14 (90-98) CCD-Barcodes



Zeile 15 (99-105) CCD-Barcodes



AS-DA (Hübner)

Benötigte Programmregister: 16

Zeile 1 (1-3) CCD-Barcodes



Zeile 2 (4-11) CCD-Barcodes



Zeile 3 (12-20) CCD-Barcodes



Zeile 4 (20-27) CCD-Barcodes



Zeile 5 (28-34) CCD-Barcodes



Zeile 6 (35-41) CCD-Barcodes



Zeile 7 (42-51) CCD-Barcodes



Zeile 8 (51-59) CCD-Barcodes



Zeile 9 (59-60) CCD-Barcodes



XFERDA (Hübner)

Benötigte Programmregister: 6

Zeile 1 (1-3) CCD-Barcodes



Zeile 2 (4-10) CCD-Barcodes



Zeile 3 (11-18) CCD-Barcodes



Zeile 4 (19) CCD-Barcodes



Der HP-28C steuert den Drucker HP 82240A über einen Infrarotstrahl an. Kein Kabelsalat mehr, keine Verbindung herzustellen. Einfach aufstellen, einschalten und los gehts.

Postvertriebsstück
Gebühr bezahlt

D 2856 E

Rolf Hansmann
Computerclub Deutschland e.V.
Limburger Straße 15
6242 Kronberg 2

CCD

ISSN 0176-8735

PRISMA

Nr. 1 Januar/Februar 1987

**NEU
BEI H & G**

**NEU
BEI H & G**

HP 28 C DM 487,00

Geben Sie uns rechtzeitig Ihre Bestellung, die Nachfrage wird groß sein.

H & G 50 x HP 41 CV

je DM 297,50 (nur solange Vorrat reicht)

50 x HP 18 CD je DM 387,50

Für CCD-Mitglieder unter Angabe der Mitgliedsnummer.

Wir sind HP Vertragshändler.

H & G Hansen & Gieraths
EDV Vertriebsges. mbH

Münsterstraße 1
Postfach 16 05
D-5300 Bonn 1

☎ (02 28) 72 90 80
Telex 17 228 3665
Teletex 228 3665=ahabo

Volksbank Bonn eG
Konto-Nr. 201 0145 018
BLZ 380 601 86