

PRISMA

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

Juli/August 1988 Nr. 4

D 2856 F



Die HP-IL RAM-Disk von CMT ist ein kleines und nützliches Gerät für unterwegs. Sie kann bis zu 512 KByte an Daten speichern. Auf Reisen ist diese somit ideal als Backup-Medium einzusetzen, damit unterwegs erfaßte Daten oder geschriebene Programme nicht durch "Memory Lost" unwiederbringlich verschwinden. Als Option gibt es auch eine RS232-Schnittstelle. Dann kann man auch von unterwegs mit einem zusätzlichen Modem in die ClubMailbox gelangen.

Clubnachrichten

Clubbörse

"To live and let die"

USA Handheld Conference '88

Neue HP Taschenrechner

Grundlagen

QuietJet RAM-Nachrüstung

MS-DOS

Referenzblätter

Serie 70

CCDUTIL

KEYCTL

CUSTUTIL

STRINGLX

Serie 40

Partialbruchzerlegung

Steuer

Widerstandsberechnung

Gehrungsprogramm

Neue Display Character

FLCS

XROM Version 2

Datenbarcodes zu BANNER

Serie 10/20

Programmsammlung Power Pack

CCD-MODUL Version A für 110,- DM, zusätzl. Literatur Nachfragen, ☎ 0421 / 6589199

Verkaufe serielle Interfacekarte **EPSON 8145** für LX/RX/FX/JX/EX/LQ-Drucker von EPSON. Originalverpackung. Preis VHS.

Biete **Einsteckdrucker HP-82143A**. Neuer Akkusatz. Schaltbarer Halbzweilenvorschub. Suche **IL-Drucker HP-82162A**. Am liebsten wäre mir Tausch! Sebastian von Borries (2184), Geniner Straße 35a, D-2400 Lübeck, ☎ 0451 / 581456

Suche preisgünstigen HP-IL Konverter (HP82166) oder HP-IL/GPIO Interface (HP82165). Wolfgang Bürgel, ☎ 0511 / 562430 ab 20 Uhr.

Verkaufe

HP-75C Handheld HPIL-Schnittstelle Magnetkartenleser DM 900,-
HP-82161A HPIL-digital-cassette-drive + 10 Kassetten DM 500,-
HP-82166A HPIL-Parallelinterface (z.B. für Epsondrucker) DM 190,-
HP-41CX Taschenrechner, Zeit + X-Funktionen Textedit. DM 320,-
HP-28CD Taschenrechner, Symbolisches Ableiten + Integr. DM 290,-
75 Zubehör
 RAMerweiterung um 8K Mathe-Modul (16K ROM), Komplexe Zahlen, Matrizen DM 190,-
 I/O ROM (24K), I/O Funktionen, Privateauth. DM 240,-
 Textformater (8K ROM), Textverarbeitung DM 140,-
75 Programmsammlungen
 Forth 1 Kassette eigenes Betriebssystem DM 240,-
 Math 1-3 Magnetkarten, Dreiecksber., Matrizen DM 90,-
41 Zubehör
 Magnetkartenleser 1F + 120 Magnetkarten DM 190,-
 WAND 1F + fünf Bücher mit Barcodes DM 190,-
 HPIL-Modul DM 150,-
 Advantage Modul, deutsches Handbuch DM 90,-
 IL-Development-Modul DM 90,-
 HP 41 Akkusatz + HP 220V Netzteil DM 90,-
 Varta Lady-Akku Ladegerät + 4 Lady-Akku DM 30,-
 Alle Preise sind Ihre Endpreise incl. Porto und Verpackung (z.T. Originalverpackung)
 Peter Habicht, ☎ 06008 / 7235 von 17.00-20.00 Uhr

HP71B + I/O 900,- DM, Digitallaufwerk 600,- DM, 2 x 4 K Ram Erw. 100,- DM, Forth 1A (Assembler zu HP41) 200,- DM, Forth 2A (Texteditor, Lex...) 200,- DM, 2 Netzteile 100,- DM, auf Wunsch im Aktenkoffer! Alle Preise VB, Gesamt 1999,- DM, N. Maier, Beethovenstraße 24, 6350 Bad Nauheim

Verkaufe wegen Systemerweiterung folgende Komponenten für **HP-41**:

Grabau Video-Interface GR7 AG-IL mit 12" Monitor für DM 800,-, Bar-Code-Leser HP82153A für DM 200,-, HP-IL Modul HP82160A für DM 100,-, Funktions/Doppelspeichermodul HP82180A mit HP82181A für DM 100,-, HP-IL Development Module HP00041-15043 für DM 60,-, Extended I/O Modul HP82183A für DM 100,-, CT Port Extender 4100 für DM 100,-, HEPAX Modul Std für DM 400,-, und folgende Literatur:

The HP-IL System - Kane/Harper/Ushijima für DM 15,-, Control the World with HP-IL - Friedman für DM 15,-, HP-41 M-Code for Beginners - Emery für DM 15,-, Plotten und Drucken auf dem Thermodrucker - Meschede für 15,-, Erweiterte Funktionen des HP41 - leicht gemacht - Dal-kowski für DM 10,-, HP41 Barcodes mit dem HP-IL System - Albers für DM 20,-, Zusammenfassung der Bedienungsanleitungen HP80 - Stroinski für DM 15,-. Alle Komponenten und Bücher in Bestzustand. Wer mehrere Teile nimmt, kann handeln.
 W. Meschede (CCD2670) abends ☎ 0911 / 808756.

Verkaufe ThinkJet 2225 B mit Acrylständer, 2. Batterie und ca. 1000 Blatt Endlospapier 550,- DM, HP-IL Interface HP 82401 A 100,- DM, HP 11 C 50,- DM
 Walter Schneider, ☎ 0221 / 431898

Verkaufe HP71 B mit HPIL-Modul u. Akkus Preis VHS IL/RS232-Interface 82164A ungebr., org. verp. DM 400,-
 Suche IBM/HP-IL-Card 82973A.
 Ulrich Bunse, 4630 Bochum, ☎ 0234 / 705052

HP 41 CX und **CCD MODUL** zusammen 450,- DM.
 Christoph Klug, Trockener Kamp 59, 3200 Hildesheim

Verkaufe: HP-71B mit IL-Modul DM 750,-, HP 2225B Think-Jet (neuwertig) DM 750,-,
 Volker Lang (130), ☎ 07138 / 5349 ab 18 Uhr

HP-71B mit IL-Modul, Magnetkartenleser, 4K Speichererweiterung und HP-41 Translator PAC für 1100 DM zu verkaufen. ☎ 05726 / 281

Verkaufe HP71B + Kartenleser + IL-Modul + Translator Pac + Zubehör (Magnetkarten, Literatur HP71, FORTH) DM 1000,- nur komplett! Hans Meier, ☎ 089/3231716, falls ich nicht da bin, Nachricht hinterlassen, ich rufe zurück!

Verkaufe: HP 2228 AD QUIETJET Tintenstrahldrucker + Papier + 2 Druckköpfe + 18K RAM 1 1/2 Jahre alt, Preis: 850,- DM; ☎ 06196/87-2051 tagsüber, 23013 abends, Martin Meyer, Robert-Stolz-Str. 5, 6232 Bad Soden

Suche leere Modulgehäuse mit Federleiste f. HP-41. Bin für jedes dankbar. ☎ 06150 / 7594, Friedrich-Ebert-Str. 93, 6106 Erzhäusen.

HP 71 B mit IL-Modul zu verkaufen: 800,- DM, Forth/Assembler Modul und Mathematik-Modul je 150,- DM, 32 kB-RAM-Modul (CMT) 200,- DM und EPROM-Halter für Kartenleserschacht 150,- DM, Diskettenlaufwerk 9114A mit Disketten und Netzteil 1000,- DM, Preise V.B. Michael Hartmann, Hainstraße 52, 5300 Bonn, ☎ 0228 / 624428

Suche für HP-41 nachstehende Einsteckmodule:
 # Aviation [00041-15018]
 # Circuit Analysis [00041-15006]
 # Clinical Lab and Nuclear Medicine [00041-15024]
 # Finanzmathematik [00041-15014]
 # Navigation [00041-15017]
 # Petroleum Fluids Pac [00041-15039]
 # Stress Analysis für Mechanical Engineering [00041-15027]
 # Structural Analysis für Civil Engineering [00041-15021]
 # Thermal and Transport Science [00041-15019]
 # Surveying [00041-15005]
 W. Knell, Prozeptionsweg 29, D 4720 Beckum

Verkaufe HP ThinkJet 2225 D mit RS 232/V24 Interface. Preis 510,- DM, Auch Tausch gegen IL-, IB oder Centronics-Version möglich. Klaus Kaiser, Mainzer Landstr. 561, 6230 Frankfurt a. M. 80, ☎ 069/397852

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

Regionalgruppe Berlin

Der Ansprechpartner ist:

Jörg Warmuth
 Wartburgstr. 17
 D-1000 Berlin 62

Regionalgruppe Karlsruhe

Der Ansprechpartner ist:

Stefan M. Schwall
 Rappenwörthstr. 42
 D-7500 Karlsruhe 21
 Tel. (0721) 576756
 E-Mail: GE01:S.SCHWALL

Regionalgruppe München

Der Ansprechpartner ist:

Victor Lecoq
 Seumestr. 8
 D-8000 München 70
 Tel. (089) 789379

Regionalgruppe Hamburg

Die Ansprechpartner sind:

Alfred Czaya
 An der Bahn 1
 D-2061 Sülfeld
 Tel. (040) 433668 Mo.-Do. abends

und

Horst Ziegler
 Schüslerweg 18 b
 D-2100 Hamburg 90
 Tel. (040) 7905672

Regionalgruppe Rhein-Main

Der Ansprechpartner ist:

Andreas Eschmann
 Lahnstr. 2
 D-6096 Raunheim
 Tel. (06142) 46642

Regionalgruppe Köln

Der Ansprechpartner ist:

Frank Ortmann
 Okerstr. 24
 D-5090 Leverkusen

Impressum

Titel:

PRISMA

Herausgeber:

CCD-Computerclub Deutschland e.V.

Postfach 11 04 11

Schwalbacher Straße 50

6000 Frankfurt am Main 1

Verantwortlicher Redakteur:

Alf-Norman Tietze (ant)

Redaktion:

Hans Jürgen Hübner (hjh)

Klaus Kaiser (kk)

Michael Krockner (mik)

Martin Meyer (mm)

Henry Schimmer (hs)

Dieter Wolf (dw)

Herstellung:

CCD e.V.

Manuskripte:

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

Druck und Weiterverarbeitung:

Reha Werkstatt Rödelsheim

Biedenkopfer Weg 40 a, 6000 Frankfurt

Anzeigenpreise:

Es gilt unsere Anzeigenpreisliste 3 vom Juni 1987

Erscheinungsweise:

PRISMA erscheint jeden 2. Monat.

Auflage:

3000

Bezug:

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

Urheberrecht:

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

"To live and let die"

engl.: "Leben und sterben lassen"

Diese Schlagzeile - in Anlehnung an einen Kinofilm mit Geheimagent James Bond 007 - fiel mir spontan zu der neuesten Nachricht über den HP-71B Handheld Computer ein. Es war in Corvallis, bereits ein paar Tage vor der Konferenz, als die neue Meldung wie eine Bombe bei uns einschlug: Der HP-71B wird NICHT mehr weitergebaut!

Hewlett-Packard hat sich entschieden, den HP-71B nicht weiter zu produzieren. Bei HP in Corvallis gibt man an, daß dies aus Kostengründen geschehe, da die Herstellung vom HP-71B doch so aufwendig sei. Ein schwaches Argument. Schließlich wurde weder in den USA noch in Europa richtig versucht, diesen Rechner zu verkaufen. Es gab so gut wie keine Werbemaßnahmen für dieses Produkt. Auch das HP-IL System wird seit etwa einem Jahr nicht mehr weiterentwickelt. Es werden keine neuen IL-Geräte mehr von HP auf den Markt kommen. Und jetzt haben wir Anwender die Folgen zu tragen.

Auf der Konferenz - wo der "Tod des HP-71B" offiziell verkündet wurde - schwebte diese Nachricht wie ein böser Geist in allen Räumen und lähmte spürbar den Enthusiasmus aller Teilnehmer. Der HP-71B ist der beste Taschencomputer und IL-Controller von HP. Ein solcher Schritt ist aus unserer Sicht völlig unverständlich. Auch Bob Puckette, ein ehemaliger HP-Ingenieur aus dem 71'er Entwicklungsteam, bedauerte zutiefst, daß HP nun diesen "feinsten" Taschencomputer aus der Produktion herausnimmt. Aber was soll's - es ist nun mal so. Was bleibt ?!

Der HP-71B ist tot, es lebe der HP-71B !

Die Aktivitäten mit und um den HP-71B werden weltweit in den Userclubs weitergehen. Zwar mit etwas gedämpften Zuwachsraten - aber immerhin. Auch wird es noch 2 bis 3 Jahre dauern, bis man im Computerladen keinen 71'er mehr kaufen kann. Das Zubehör wird ohnehin noch eine Weile im Programm bleiben. Der HP-71B ist noch immer seiner Zeit voraus. Ein Vergleich der Leistungsfähigkeit mit Taschencomputern anderer Fabrikate hinterläßt die Konkurrenten Sharp, Casio - und wie sie alle heißen - in einem sehr düsteren Licht. Auch der Preis darf bezüglich der Leistung als angemessen betrachtet werden. Es beklagt sich ja auch kein Mensch darüber, daß

ein Mercedes 300 TE mehr kostet als ein Honda Civic aus Japan.

Die hervorstechenden Merkmale des HP-71B sind die hohe Rechengenauigkeit und eine hohe Rechengeschwindigkeit. Zusammen mit dem Mathe-Modul ist der 71'er sogar ein mathematischer "Kraftzweig" in Schokoladentafelgröße mit Leistungsmerkmalen, wie man sie sonst nur von Großrechnern an Universitäten oder in Forschungslabors kennt. Die Dateiverwaltung, globale Unterprogramme mit Parameterübergabe, FORTH und Lexfiles sind ebenfalls überdurchschnittliche Pluspunkte, die woanders nicht zu finden sind. Mit JPCEPROM, CCDUTIL (das neue Club-Lexfile), HP-IL Modul und weiteren RAM- oder EPROM-Modulen (von CMT) gerät der HP-71B zum Power Pack ersten Grades. Und Hewlett-Packard nimmt ihn aus der Produktion heraus. Das ist echtes Marketing! Man bedenke: HP ist eine Weltfirma mit dem Anspruch, qualitativ höchstwertige Produkte zu verkaufen - und dort weiß man, was man tut. Vielleicht finden wir bald einen echten HP Taschenrechner bei Tschibo oder Eduscho im Schaufenster zwischen Reiseweckern und Badehosen. Für nur 49,50 DM einschl. Infrarot-Druckerchen?! Aber programmierbar ist der nicht mehr.

Innerhalb des CCD und in anderen europäischen Clubs geht es nun trotzdem munter weiter mit dem HP-71B. Viele neue und sehr nützliche BASIC-Befehle werden entwickelt, so daß man zukünftig noch komfortabler mit dem 71'er programmieren kann. Interessante und vielseitige Anwendungsprogramme sind derzeit zur Veröffentlichung im PRISMA geplant. Unter anderem auch ein Datenbankprogramm, daß einen Datenaustausch mit dBASE III PLUS (auf dem Personal Computer) ermöglicht. Die Perspektiven für den mobilen Einsatz des HP-71B als Datenaufnahme- und Informationssystem sehen gut aus. Und alles ist datenkompatibel für Programme auf den stationären PC's.

Ich werde mir noch einen zweiten 71'er kaufen, der immer in meinem Auto "mitfährt" und mir die Reisekostenabrechnung für das Finanzamt erledigt. Der läßt sich sogar ohne Tricks (!) von der Steuer absetzen. Außerdem: bald wird der HP-71B wahrscheinlich billiger bzw. ausverkauft werden. Und wer außer mir will schon ein Auslaufmodell ?!

Happy Programming Alf-Norman Tietze
(Chefredakteur)



Inhalt

Clubnachrichten

Clubbörse	2
"To live and let die"	3
Impressum	3
USA Handheld Conference 1988	4
Aktive Werbung für den CCD	46
Neue HP Taschenrechner	48

Grundlagen

QuietJet RAM-Nachrüstung	5
--------------------------	---

MS-DOS

MS-DOS Referenzblätter	17
------------------------	----

Serie 70

Lexfile: CCDUTIL	7
Lexfile: KEYCTL	8
Lexfile: CUSTUTIL	14
Lexfile: STRINGLX	39
Korrektur zu SPLINE30	16

Serie 40

Partialbruchzerlegung	19
Steuer	24
E-Technik: Widerstandsberechnung	27
Gehrungsprogramm	29
Neue Display Character	31
FLCS	31
XROM Version 2	32
Datenbarcodes zu BANNER	33
Korrektur zu "Division o. Grenzen"	40

Serie 10/20

Programmsammlung Power Pack	37
-----------------------------	----

Barcodes	41
----------	----

Clubadressen	46
--------------	----

International Conference of Hand-Held Computer Users

vom 4. bis 6. August 1988 in Corvallis, Oregon (USA)

"Vergangenheit, Gegenwart und Zukunft der Handheld Computer" - so lautete das Motto zur diesjährigen Konferenz. Die ursprüngliche Idee zur diesem globalen Zusammentreffen wurde bereits kurz nach der "Handheld"-Konferenz August 1987 in Kopenhagen geboren. Und zwar von Clubmitglied Sherman Lowell (jaja - der CCD hat auch in den USA einige Fans) und Clubfreund David Lin, der "Boss" von Corvallis MicroTechnology Incorporation.

Bereits 1981 hatte in Corvallis (Oregon) eine PPC-Tagung über den HP-41C stattgefunden, da für die Teilnehmer die Möglichkeit bestand, das dortige Werk von Hewlett-Packard zu besichtigen. In Corvallis entwickelt HP alle Taschenrechner und Taschencomputer. So bot sich dieser Ort auch 1988 für die internationale Konferenz an.

Im LaSells Stewart Center der Oregon State University fand sich genügend Platz für das Auditorium. Es wurde einfach ein Hörsaal zur Verfügung gestellt. In einigen Seminarräumen konnten Clubs, Händler und Hersteller ihre Angebote darstellen. Alle Konferenzteilnehmer konnten auch diesmal das HP Werk besichtigen. Nachfolgend nun die Zusammenfassung einiger Themen.

Geschichte und Gegenwart der HP Handheld Computer

Nach der offiziellen Eröffnung der Konferenz wurden am ersten Tag von heutigen und ehemaligen HP Mitarbeitern über die Entwicklungsgeschichten der HP Taschencomputer referiert: Bernie Musch (HP-41), Bob Puckette (HP-71), Don Morris (HP-75) und Bill Wickes (HP-28). So entstand ein interessanter Überblick und die historisch/technischen Zusammenhänge der einzelnen Entwicklungen wurden deutlich.

Noch am Abend desselben Tages bis in die Nacht gab es eine "Round-Table Discussion" zwischen den Konferenzteilnehmern und HP-Vertretern. Hauptgegenstand dieser Diskussion war selbstverständlich der "Tod des HP-71B", der die ganze Konferenz mit seinem "Schatten" überlagerte. Wie zu erwarten war, gab HP keine präzisen Auskünfte über einen möglichen Ersatz. Wichtig war allerdings, daß HP auf diesem Wege - international (!) - deutlich gemacht werden konnte, wie groß der Bedarf an einem leistungsfähigen Taschencomputer mit Ein-/Ausgabe-Möglichkeiten zu Peripheriegeräten ist.

Anwendungen von Taschencomputern in der Praxis

Eine interessante Verbreitung erfährt der HP-28 zur Zeit in Schulen und Universitäten der USA als Lehrinstrument für höhere Mathematik. Thomas Dick vom "Mathematics Department Oregon State University Corvallis" berichtete über eine umfangreiche Studie seines Departments. Die Schüler/Studenten bauen ihre Abneigung gegen kompliziert scheinende Taschencomputer ab und erleben mehr Spaß beim Lernen mathematischer Themenbereiche. Durch die Grafikfähigkeiten des HP-28 entwickelt sich bei ihnen ein besseres "grafisches Verständnis" für Funktionen.

Die enorme Leistungsfähigkeit des HP-28 dürfe allerdings nur begleitend zum eigentlichen Erlernen der mathematischen Lösungswege hinzugezogen werden. Andernfalls mache der Taschencomputer als "Black-Box" nur wenig pädagogischen Sinn. Zusammenfassend aber mache der Einsatz solcher "Werkzeuge" im Mathematikunterricht einen vielversprechenden Eindruck.

Zahlreiche andere Anwendungen machen deutlich, daß es für den Einsatz eines Taschencomputers zur mobilen Datenerfassung keinen adäquaten Ersatz durch unhandliche Laptop-PC's gibt, solange eine sehr wesentliche Eigenschaft vorhanden ist: Der Datenaustausch zwischen Handheld und Personal Computer.

Die Beispiele während der Konferenz reichten von der mobilen Meßdatenerfassung - Temperaturen, Ultraschallvermessung von Rohrleitungen, geodätische Vermessung - bis hin zu Datenbanken und Informationssystemen im "Handheld-Format". Hier gibt es eine besonders nennenswerte Neuheit für den MC-II von CMT. Seit wenigen Wochen gibt es das MBASE-Modul, ein dBASE III Plus kompatibles Datenbanksystem - jetzt erstmalig in einem Handheld Computer. Zusammen mit der neuen RAM-Disk (1 MByte + HP-IL Schnittstelle) kann man nun einen beachtlichen Informationsumfang "auf der Hand" mit sich herumtragen. Die Programmfiles (.PRG), die Formatmasken (.FMT) und natürlich auch die Datenbanken (.DBF) sind sowohl für den PC als auch für den MC-II kompatibel.

Ebenfalls neu ist das MC-II-BASIC-Modul. Damit koennen jetzt auch GWBASIC-Programme auf dem MC-II ablaufen. Darüber hinaus ist das neue

BASIC in einigen wichtigen Dingen erweitert (I/O über RS-232, HP-IL, Alarmfunktionen und eine höhere Rechengenauigkeit).

Der Datenaustausch zwischen Taschencomputern und PC's via HP-IL wird durch eine neue LINK-Software aus Texas noch komfortabler. Sobald das Programm auf dem Markt ist (Herbst 88 ist vorgesehen) werden wir einen Testbericht darüber im PRISMA bringen.

Zukünftige Entwicklung der Handheld Computer

Durch sinkende Preise bei Prozessor und besonders bei RAM-Chips werden zukünftige Handheld Computer immer leistungsfähiger werden. Auch neue Technologien zur höheren Auflösung bei Bildschirmen (hier: LCDs) werden dazu beitragen. CPUs mit 16 oder 32 Bit sind ebenso absehbar wie Arbeitsspeicher von 1 MByte Größe. Vieles wäre bereits technisch zu realisieren, allerdings noch zu einem Preis, den zur Zeit niemand dafür bezahlen würde.

Infrarot/RS-232 Interface

Als erste Reaktion auf die neue Infrarot-Übertragungstechnik der HP-Taschencomputer brachte die Firma RUSH Systems ein Infrarot/RS-232 Interface für PC's auf den Markt. Damit ist es möglich, alles was auf den Infrarot-Drucker zu bringen ist auch in den PC zu übertragen - allerdings nur in dieser Richtung. Denn bei dem HP Infrarotsystem handelt es sich ja leider nicht um ein Ein-/Ausgabe-System, sondern es ist lediglich die Ausgabe von Daten möglich. Der Preis für das Hook-up Infrarot/RS-232 Interface beträgt 95.- US Dollar.

JPC-EPROM jetzt von CMT

Das JPC-EPROM für den HP-71 wird jetzt auch von CMT in einem 32 KByte EPROM-Modul (Frontport) vertrieben. Dazu kommt ein sehr gut gedrucktes Handbuch und sogar eine Kurzanleitung im gewohnten Format. Der Preis ist günstig und beträgt 185.- US Dollar. Die CMT-Produkte sind in Deutschland bei der Firma W&W in Bergisch-Gladbach zu beziehen.

1989 hat HP übrigens sein 50-jähriges Bestehen. Man darf gespannt sein!

(ant)

QUIETJET RAM-Nachrüstung

Wer in der Bedienungsanleitung seines Druckers gelesen hat, der weiß, daß man im RAM einzelne Zeichen des Standardzeichensatzes gegen Eigendefinierte ersetzen kann, es lassen sich auch ganze Zeichensätze vom Rechner aus in dieses RAM laden, ähnlich wie dies bei den Laserdruckern gemacht wird. Mit 2 kByte im Grundausbau läßt sich dies alles aber nur in kleinem Rahmen oder garnicht realisieren, das RAM muß ja noch den Puffer spielen.

Im Handbuch steht auch, daß man dieses RAM mit Hilfe eines Kits von HP nachrüsten kann, wie wir wissen, ist dies kein sonderlich billiges Unterfangen. Als Techniker dachte ich mir, das müßte doch auch auf preiswertere Art und Weise zu schaffen sein, also Schraubendreher zur Hand und auf Los geht's los:

Zu allererst ziehen wird den Drehknopf mit einem sanften Ruck von seiner Achse.

Wie schon gesagt nehmen wir jetzt einen mittleren Kreuzschlitzschraubendreher zur Hand und drehen unseren Drucker auf den Kopf, keine Angst, er tut sich dabei nicht weh, man muß nur den hinteren Deckel für den Traktor festhalten.

An allen vier Ecken sind große Löcher zu sehen, in denen sich tief unten die Kreuzschlitzschrauben befinden, die wir jetzt erst einmal herausdrehen.

Nun fassen wir den Drucker so an, daß er beim nochmaligen Umdrehen in seine normale Lage nicht auseinanderfällt. Klappt man jetzt den vorderen Deckel aus, so fällt er einem entgegen, er ist einfach zwischen dem Gehäuseober- und -unterteil eingeklemmt; einfach, aber funktionell.

Wir drehen den Drucker jetzt so, daß wir die hintere Seite vor uns haben und heben das Oberteil leicht an, dann nach links um die Achse des Drehknopfes herum nach oben zu uns hin, das Oberteil hängt an den Verbindungskabeln für die Drucktaster, also nicht zu heftig ziehen. Dieses Kabel muß man nach dem Wegklappen des Deckels von der vor uns liegenden Platine ziehen.

Die Platine ist auf einem Abschirmblech montiert, das an der Vorderseite 90° nach oben gebogen ist, siehe dazu Bild 1, man sieht hier von hinten auf den Drucker ohne Oberteil.

Die Platine ist an mehreren Stellen auf das Abschirmblech geschraubt, die Einheit Platine/Abschirmblech zusammen ist aber lediglich gesteckt. Dazu seht Euch mal die Kunststoffhaken rechts und links des Bleches (der Platine) an. Diese Haken muß man jetzt, am besten mit den Fingern gegen die Platine hebelnd, etwas nach links bzw. rechts drücken, eben soweit, daß man das Blech über die Raste des jeweiligen Hakens heben kann.

Nun hebt man die ganze Einheit soweit an, daß das Blech mit der Unterseite den Rand des unteren Gehäusebodens erreicht und zieht, vielleicht mit etwas Nachdruck, an der linken Seite mehr als an der rechten, die ganze Einheit aus ihrer Verankerung der Gummivalze. Hier befinden sich zwei weiße Kunststoffstützen, in die das Blech einfach eingesteckt ist, keine Angst, die Dinger (Bild 2) sind recht flexibel und brechen nicht, soweit kann man die Platineneinheit garnicht nach oben heben.

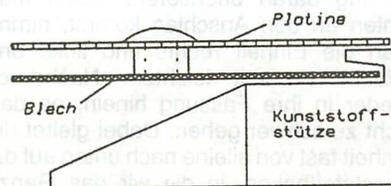


Bild 2

Beim Ziehen muß man lediglich darauf achten, daß man das Ganze nicht zu weit herauszieht, in der Mitte befindet sich nämlich das Flachkabel für den Druckkopf. Wir ziehen also die Einheit links soweit wie möglich heraus und ziehen jetzt die Kabel der beiden Motoren von der Platine. Jetzt stört uns nur noch das Flachkabel zum Druckkopf, dieses ist vorne unter einem hellen Kunststoffteil um 90° nach links geknickt, dieses Kunststoffteil kann mit einem Schraubendreher

oder einer kurzen kräftigen Zange nach oben abziehen, es ist auch nur gesteckt.

Damit können wir nun die Platine soweit aus dem Drucker herausziehen, daß wir alle 4 Schrauben des über der linken Hälfte befindlichen Abschirmblechs erreichen und aufschrauben können, dafür können wir wieder denselben Kreuzschlitzschraubendreher wie für das Gehäuse verwenden.

Das Abschirmblech kann man jetzt einfach hochheben, und schon sind wir am Ziel unserer Wünsche, die gesamte Intelligenz des Druckers liegt, in mehrere ICs gepackt, vor uns, siehe Bild 3.

Was wir jetzt auch sehen können ist, daß sich auf der Platine zwei leere 28 polige IC-Sockel befinden. Diese sind für die beiden Speicherbausteine gedacht, die man laut Bedienungsanleitung mit Hilfe des Kits nachrüsten kann, wobei dieses Kit wahrscheinlich aus zwei ganz gewöhnlichen 8 kByte großen RAMs besteht, die im Laden zusammen rund 20,- DM kosten können und von HP dann für vielleicht 100,- DM verkauft werden.

Als Hilfe für absolute Computerlaien:

Wir benötigen für die Speichererweiterung minimal 1, maximal 2 8 kByte (64 kBit) große statische RAMs, sie stecken jeweils in einem 28 poligen Dual In Line (=DIL) Gehäuse, d.h. ein schwarzer Käfer mit je 14 Beinchen auf jeder Seite.

In der Regel steht auf dem IC: fff 2064 ... fff ist irgend ein Firmenkürzel, das den Hersteller angibt, wichtig sind die 2064, das steht auf fast allen statischen RAMs mit 8k*8 großem Speicher, wobei die 20 für statisches RAM steht und die 64 die Größe des Speichers in kBit, bei 8 kByte zu je 8 Bit pro Byte ergibt sich nach Adam Riese $8 \cdot 8 = 64$, so einfach ist das!

Die Zugriffsgeschwindigkeit der RAMs ist im Prinzip unwichtig, man kann so gut wie alle auf dem Markt verfügbaren Bausteine verwenden, der Prozessor ist kein Geschwindigkeitsrekorder.

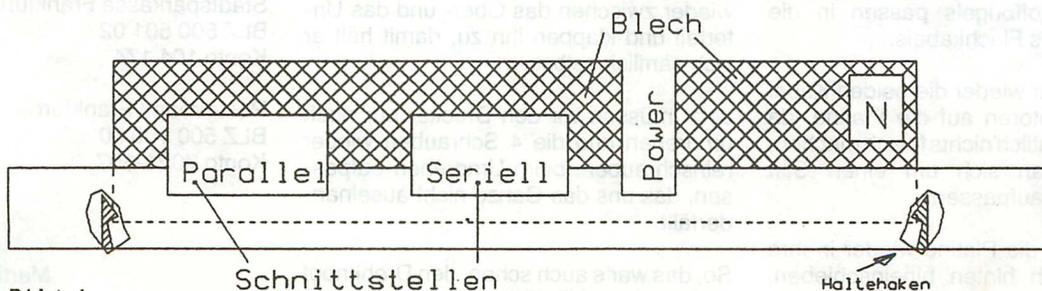


Bild 1

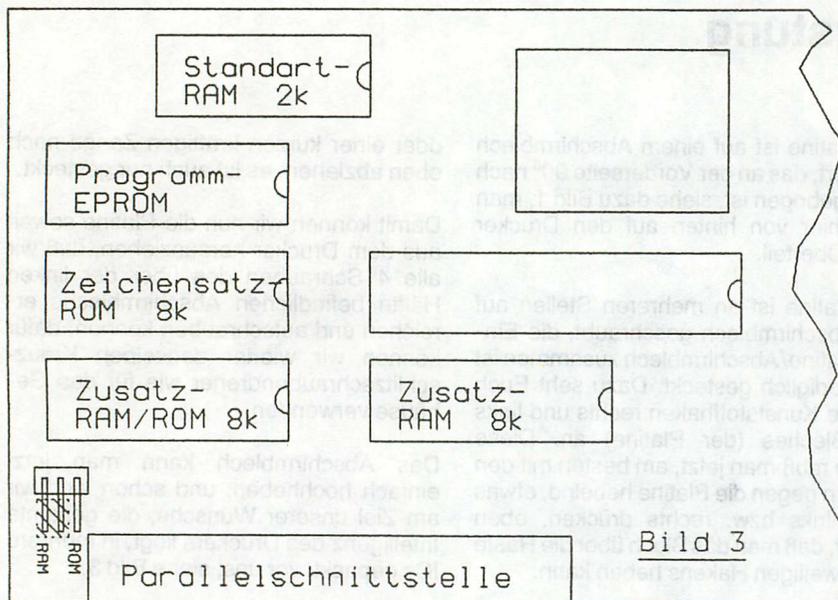


Bild 3

Diese beiden Speicherbausteine stecken wir jetzt gleichsinnig zu den übrigen ICs in ihre Fassungen, eventuell muß man vorher die Beinchen ein wenig zusammenbiegen, damit man problemlos in die Fassung kommt. Mit gleichsinnig meine ich in derselben Richtung, d.h. die Markierung, egal ob wie auf der Zeichnung eine halbrunde Einkerbung oder ein Punkt am Pin Nr. 1, zeigt immer nach rechts und die Schrift steht somit auf dem Kopf wie bei den anderen ICs auch.

In der linken Ecke sehen wir die Kodiermöglichkeit für RAM/ROM, diese stecken wir ganz nach links auf die Stellung RAM, das ist lediglich ein Kurzschlußbügel über die jeweiligen Stifte. Man kann nämlich statt des linken RAMs auch ein ebenso großes ROM oder EPROM einstecken, das einen benutzerdefinierbaren Zeichensatz enthält, wer Unterlagen über die Kodierung dieses Speichers hat kann mir diese zukommen lassen, es würde mich sehr interessieren.

Jetzt wenden wir uns wieder dem Zusammenbau unseres Lieblings zu:

Das Abschirmblech wird wieder mit seinen 4 Schrauben auf der Platine festgeschraubt, dann schieben wir die Platine etwa zur Hälfte wieder in den Drucker hinein, ziehen wieder das Flachkabel des Druckkopfes durch seine Öffnung vorne und können das vorhin entfernte helle Kunststoffteil wieder in seine alte Lage zurückstecken, die linken Haken dieses hellen Kunststoffbügels passen in die Aussparung des Flachkabels.

Nun stecken wir wieder die beiden Kabel der Druckermotoren auf die Platine, da kann man eigentlich nichts falsch machen, außer daß man sich um einen Stift versteckt, also aufpassen.

Bevor wir jetzt die Platine wieder in ihre Halterung nach hinten hineinschieben, müssen wir noch den Bügel für die Pa-

piererkennung hochheben. Dieser befindet sich genau dort unterhalb der Gummiwalze, wo diese auf gut 1/2 cm eingekerbt ist, etwas rechts von der Mitte, wenn man von hinten schaut.

Jetzt schieben wir mit dem hochgehobenen Bügel (schwarz) die Platine/Abschirmblecheinheit sanft in den Drucker hinein, wobei die Blech Aussparungen für die Schnittstellen genau zu den Aussparungen im Gehäuseboden für dieselben passen, man kann sich für die Positionierung daran orientieren. Wenn man hinten an den Anschlag kommt, nimmt man die Einheit rechts und links und schiebt sie mit leichtem Nachdruck wieder in ihre Fassung hinein, es darf nicht zu schwer gehen. Dabei gleitet die Einheit fast von alleine nach unten auf die Kunststoffhaken, in die wir das Ganze nun hineindrücken. Dabei sollte man die Haken mit den Fingern etwas wegdrücken, da das Blech etwas scharfkantig ist und sonst am Kunststoff hängen bleibt.

Jetzt nehmen wir uns das Gehäuseoberenteil, stecken dessen Anschlußkabel wieder auf die Platine und setzen das Oberenteil am besten links zuerst über die Achse des Papiertransports auf den Drucker, wobei man den Umschalthebel rechts eventuell ein wenig nach links biegen muß, damit man darüber kommt.

Haben wir das Oberenteil glücklich aufgesetzt, so nehmen wir den herumliegenden vorderen Deckel und stecken ihn gerade wieder zwischen das Ober- und das Unterteil und klappen ihn zu, damit hält er sich nämlich selbst.

Jetzt müssen wir den Drucker nur noch umdrehen und die 4 Schrauben wieder reinschrauben, beim Umdrehen aufpassen, das uns das Ganze nicht auseinanderfällt.

So, das war's auch schon, den Drehknopf dürfen wir natürlich nicht vergessen.

Wenn wir jetzt den Drucker mit gedruckter ONLINE und FF einschalten und damit einen Selbsttest auslösen, dann müßte nach dem Ausdruck der Schalterstellung nicht mehr 2k RAM sondern 18k RAM stehen, damit hätten wir dann auch die Bestätigung für den erfolgreichen Einbau bekommen und können von nun an nach Herzenslust auch mit eigenen Zeichensätzen drucken, viel Spaß dabei.

Sollte jemand Erfahrungen mit dem Konstruieren und Laden von Zeichensätzen auf diesem Drucker haben, einen Artikel ist dies allemal wert.

Martin Meyer (1000) Redaktion

Nachbestellungen

Es kommen immer wieder Fragen zu Leistungen des Clubs an seine Mitglieder, ich möchte an dieser Stelle mal wieder einige herausgreifen, es gehört einfach ab und zu, vor allem für Neulinge, noch einmal erwähnt.

Nachbestellungen

1. von BEST OF PRISMA (ein Buch der Besten Programme und Beiträge der Jahre 82-85 zum HP41): man schicke einen Brief an die Clubadresse mit der Bitte um Zusendung dieses Buches, Unkosten 30,- DM, entweder in Form eines Verrechnungsschecks oder einem Überweisungsbegleib auf eines der Konten des Clubs, siehe unten.
2. von alten Heften: man schicke einen Brief mit 5,- DM pro Heft in Form von Briefmarken an die Clubadresse, die gewünschten Hefte natürlich nicht vergessen anzugeben, bei mehreren Heften ist das mit den Briefmarken etwas umständlich, Zahlungswise dann besser wie oben.

Bestellung des Inhaltsverzeichnisses von PRISMA 1982-86, 1987 noch in Arbeit:

Brief an die Redaktion mit SAFU (selbst-adressierter frankierter DIN A4 Umschlag) mit 1,60,- DM in Briefmarken.

Konten:

Stadtparkasse Frankfurt
BLZ 500 501 02
Konto 104 174

Postgiroamt Frankfurt
BLZ 500 100 60
Konto 7021-607

Martin Meyer (1000) Redaktion

HP-71 Utilities

CCDUTIL

Ein nützliches HP-71 LEX-File

Nun will ich euch endlich ein nicht mehr so ganz junges Projekt vorstellen: ein CCD-LEX-File für den HP-71. Wie im letzten Heft bereits von Alf-Norman Tietze erwähnt, sind dem CCD von HP die letzten 48 Token der ID#E1 reserviert worden (an dieser Stelle sei Andreas Wolpers für seine Mühe um den Schriftverkehr mit HP gedankt). Aber keine Angst! Wenn das nicht reicht, können wir noch reichlich nachkriegen. Damit wir auch in diese Verlegenheit kommen, sind ALLE dazu aufgerufen, sich an diesem Projekt zu beteiligen. Nicht nur Programmierer sind willkommen, sondern auch jede Anregung wird dankend angenommen. Ihr braucht euch keinerlei Restriktionen aufzuerlegen, ob etwas auch verwirklicht werden kann, oder nicht. Es ist fast alles möglich. Im Einzelfall werden die Programmierer das schon herausfinden.

Um Assembler-Anfängern den Einstieg zu erleichtern, werde ich von Zeit zu Zeit Listings aus unserem Projekt im Prisma veröffentlichen.

Da das Projekt, wie bereits erwähnt, nicht mehr das Jüngste ist, hat es auch schon einige Früchte getragen: Dreizehn Befehle sind bereits fertig und einige andere geplant bzw. schon in der Mache. Dazu gehören Befehle zur formatierten Eingabe, zum Durchsuchen und Bearbeiten von Datenbeständen, zur binären Bearbeitung von Strings, zum Auslesen, Zurückschreiben und Setzen diverser Rechnerstati, ein dem des HP-75 entsprechender Appointment Mode (Alarmer), etc. Die bereits fertigen Befehle sind in dem LEX-File CCDUTIL in der Clubbibliothek erhältlich. Diese Befehle haben bereits fest zugewiesene, von HP abgesegnete Token, so daß keine Konflikte mit irgendwelchen anderen LEX-Files auftreten können (wenn doch, dann ist das nicht meine Schuld).

Es folgt nun eine Kurzanleitung zu diesen Befehlen.

Alle Anweisungen sind über das Tastenfeld ausführbar und nach "IF...THEN...ELSE" erlaubt (die Funktionen sowieso).

DDELAY Anweisung
Ist eine DISPLAY IS Einheit deklariert, wird das Delay auf 0,0, ansonsten auf .5, 125 gesetzt.

PDELAY Anweisung
Ist eine PRINTER IS Einheit deklariert, wird das Delay nicht beeinflusst.
Ist keine PRINTER IS Einheit, jedoch eine DISPLAY IS Einheit deklariert, so wird das Delay auf 0,0 gesetzt.
Ist keine PRINTER IS Einheit und keine DISPLAY IS Einheit deklariert, so wird das Delay auf INF, INF gesetzt.

SWDSP [ON/OFF] Anweisung
SWDSP OFF deaktiviert die DISPLAY IS Einheit.
SWDSP ON reaktiviert die DISPLAY IS Einheit.

SWPRT [ON/OFF] Anweisung
SWPRT OFF deaktiviert die PRINTER IS Einheit. Alle PRINT-Kommandos werden zum Display geschickt.
SWPRP ON reaktiviert die PRINTER IS Einheit.

DSP Funktion
DSP gibt die Adresse der DISPLAY IS Einheit in der Form pp.ssl zurück, wobei "pp"

die Primär-, "s" die Sekundäradresse und "l" die Loopnummer ist.

(Ich hoffe mal, daß das mit der Loopnummer auch funktioniert, weil ich nur ein IL-Modul habe und das deshalb nicht ausprobieren konnte)

KBD Funktion
KBD gibt die Adresse der KEYBOARD IS Einheit in der Form wie DSP zurück.

PRT Funktion
Dito für die PRINTER IS Einheit.

FILES\$ (<filespec>) Funktion
Gibt den Katalogeintrag des spezifizierten Files zurück (ähnlich CAT\$).

MESSAGE <errn>[,<opt>] Anweisung
Zeigt die durch <errn> spezifizierte Systemnachricht an (ähnlich MSG\$).
Mit <opt> lassen sich Optionen maskieren. Die Optionen-Maske ist ein Dezimalwert; die Bits der Binärdarstellung spezifizieren die jeweilige Option.

Decimal-

BIT	wert	Option
0	1	Message ohne Delay anzeigen
1	2	Message ohne "ERR/WRN" &ERRL anzeigen
2	4	ERRN und ERRL nicht abspeichern
3	8	Errorbeep
4	16	Delay ignorieren
5	32	ERROR erzeugen (andernfalls WARNING)

Optionen 0, 3 und 4 sind nur bei gelöschtem Bit 5 möglich.

KEYCTL [(k\$[,<opt>])] Funktion
Gibt eine Tastennummer zurück.
Wird KEYCTL ohne Parameter verwendet, wartet der Rechner, bis eine der folgenden Tasten gedrückt wird:

Taste: zurückgegebener Wert

Y	: 1
N	: 0
ATTN	: -0

Wird ein Nullstring übergeben, wartet der Rechner bis eine Taste gedrückt wird und gibt den Tastencode in dezimaler Form zurück.

Ist der String kein Nullstring, werden damit die erlaubten Tasten spezifiziert. Nur diese Tasten können die Funktion beenden. Zwei verschiedene Formate sind erlaubt: Entweder werden die Tasten durch je ein der Tastennummer entsprechendes Byte oder durch ein Doppelkreuz gefolgt von der Tastennummer angegeben.

Mit dem zweiten Parameter können Optionen maskiert werden, wobei das erste Bit

angibt, ob der Rechner auf einen Tastendruck warten soll, wenn der Tastaturpuffer leer ist und das zweite Bit, ob bei einem Druck auf die ATTN-Taste die Programmausführung nach Beendigung der Funktion unterbrochen werden soll.

COUNT (s\$,t\$) Funktion
Gibt die Häufigkeit, die t\$ in s\$ enthalten ist, zurück. Wird als t\$ ein Nullstring übergeben, wird t\$ als ein zwei Bytes großer Zeiger interpretiert, welcher auf den nächsten Zeiger zeigt.

CUT\$ (s\$,t\$,i[,j]) Funktion
Gibt aus einer Liste von Elementen in s\$, die durch einen Terminator t\$ getrennt sind,

HP-71 Utilities

KEYCTL

Tastatur Menüfunktion

Als erstes will ich KEYCTL (key control) vorstellen. Es ist eine Funktion, die es ermöglicht, ähnlich wie KEY\$ oder KEYWAIT\$ einzelne Tasten abzufragen. Es können optional bis zu zwei Parameter übergeben werden, wobei der erste ein String und der zweite ein numerisches Argument sein muß.

Wird kein Parameter angegeben, so reagiert der Rechner nur auf drei Tasten: "Y", "N" und "ATTN". Wird "Y" gedrückt, so wird 'eins' zurückgegeben, bei "N" 'null' und bei "ATTN" 'minus null'.

Mit dem String-Parameter werden die Tasten spezifiziert, die als Eingabe erlaubt sind. Es wird dann die Position der gedrückten Taste in dem String zurückgegeben. Zwei verschiedene Formate sind erlaubt: entweder werden die Tasten als einzelne Bytes angegeben, wobei der binäre Wert des Bytes der Tastennummer entspricht, oder es werden pro Taste ein Doppelkreuz gefolgt von der Tastennummer in dezimaler ASCII-Darstellung angegeben. Das sähe zum Beispiel im Falle der ungeschifteten

das i-te (bis j-te) Element zurück.
Wird als t\$ ein Nullstring übergeben, so wird dieser wie bei COUNT als Zeiger interpretiert.

NPOS (s\$,t\$,i) Funktion
Gibt die Position des i-ten Auftretens von t\$ in s\$ zurück.
Ist t\$ ein Nullstring, siehe unter COUNT.

Mathias Rabe
Muckenbrock 14
4803 Steinhagen

Cursor-Tasten, der END LINE-Taste und der ATTN-Taste so aus:

K=KEYCTL(CHR\$(47)&CHR\$(48)&CHR\$(50)&CHR\$(51)&CHR\$(38)&CHR\$(43))

bzw: K=KEYCTL("/023&+")
oder: K=KEYCTL("#47#48#50#51#38#43").

Wird ein Null-String übergeben, so wird die Tastennummer der gedrückten Taste zurückgegeben.

Mit dem zweiten Parameter können Optionen maskiert werden, wobei das erste Bit angibt, ob der Rechner auf einen Tastendruck warten soll, wenn der Tastaturpuffer leer ist und das zweite Bit, ob bei einem Druck auf die ATTN-Taste die Programmausführung nach Beendigung der Funktion unterbrochen werden soll.

Der Wert des Parameters sieht also wie folgt aus:

- Option 1 : 1 (nicht warten)
- Option 2 : 2 (nicht abbrechen)
- beide Optionen : 3

LEX	'KEYCTL'	
ID	#E1	
MSG	Ø	
POLL	Ø	
ENTRY	KEYCTL	
CHAR	#F	
KEY	'KEYCTL'	
TOKEN	#DE	
ENDTXT		
ARGERR EQU	#0BF19	"Invalid Arg"-Fehler erzeugen
ARGSTA EQU	#0E90C	eine reelle Zahl vom Mathe Stack lesen
ATNFLG EQU	#2F442	Adresse des "ATTENTION FLAGS"
CKSREQ EQU	#00721	check service request
CSLW5 EQU	#0ED3D	5 mal CSL W
CSRW5 EQU	#0ED2C	5 mal CSR W
FLT0H EQU	#1B223	Floating Point Zahl in eine Hexadezimalzahl umwandeln

```

FNRTN1 EQU    #0F216    Funktion return
FPOLL EQU     #1250A    fast poll
HDFLT EQU     #1B31B    Hexadezimalzahl in eine floating point Zahl umwandeln
PNDALM EQU    #2F761    Adresse "pending alarms"
POPBUF EQU    #010EE    eine Tastennummer aus dem Tastaturpuffer lesen
POPIS EQU     #0BD38    einen String vom Mathe Stack lesen
RPTKY EQU     #152BA    repeating keys überprüfen
SETTMO EQU    #13158    10 min. timeout setzen
SLEEP EQU     #006C2    auf einen Tastendruck warten
pWTKY EQU     #0001C    Poll Nummer "waitkey"
    
```


NIBHEX 840Z 0-2 Argumente. Zuerst ein String, dann eine Zahl

KEYCTL AD0EX D0 nach RZ retten (wird zum Schluß benötigt)
 RZ=A

?C=0 S wenn kein Parameter, dann YESNO
 GOYES YESNO

C=0 A falls nur ein Parameter, dann keine Optionen (Vorgabe)

C=C-1 S keine Optionen ?
 ?C=0 S
 GOYES NOOPT

GOSBVL ARGSTA ansonsten Optionen nach C (A)
 GOSBVL FLTDH
 D1=D1+ 16
 C=A A

NOOPT ST=C Optionen ins Status Register

GOSBVL POPIS String vom Mathe Stack lesen

?A#0 A Stringlänge größer 0 ?
 GOYES MENU dann MENÜ

GOSUB KEY# ansonsten lies eine Taste aus
 A=B A

END GOSBVL HDFLT in eine FloatingPoint Zahl konvertieren
 END2 C=A W

?ST=0 1 wenn Option 2 (ATTN-Taste wirkt wie jede andere Taste)
 GOYES ATN nicht gesetzt ist, dann überspringe das nächste

A=0 A sonst lösche das Attention-Flag
 D0=(5) ATNFLAG
 DAT0=A 1

ATN A=RZ D0 zurück schreiben
 AD0EX

GOVLNG FNRTN1 fertig !! (mit dem Wert in C (W)) (einziger Ausgang !!)

YESNO "Y"=1, "N"=0, "ATTN"=-0

GOSUB KEY#
A=0 W lies eine Taste aus

LC(2) 34 Taste "N" ?
?C=B B
GOYES ENDZ

LC(2) 6 Taste "Y" ?
?C=B B
GOYES KEY1

LC(2) 43 Taste "ATTN" .?
?C#B B
GOYES YESNO wenn nicht, dann noch mal
P= 15 sonst zeige auf das Vorzeichen
GONC END1 und fertig B.E.T.

KEY1 P= 14 zeige auf die erste Ziffer

END1 A=A+1 P inkrementiere erste Ziffer oder Vorzeichen
GONC END2 fertig B.E.T.

MENU Menü (gibt nur spezifizierte Tasten zurück)

C=0 W Länge des Strings (in Nibbles)
C=A A

ADIEX D1 auf Stringanfang
A=A+C A
D1=A

CSRB String in Bytes
D=C A

GOSUB cslw5 Länge und Anfang des Strings nach R1 retten
C=A A
R1=C

GOSUB KEYDEF String ggf. umwandeln ("#nn" nach CHR\$(nn))

MKEY MENU main loop
GOSUB KEY# eine Taste auslesen

C=R1
D0=C D0 auf Stringanfang
GOSUB csw5 und Stringlänge nach D (A)
D=C A

C=C-1 A

KEYCT D0=D0- 2 nächstes (erstes) Zeichen mit der ausgelesenen
A=DAT0 B Taste vergleichen.
?A=B B wenn gleich,
GOYES OUT dann OK und raus

```

-----
C=C-1 A sind noch Zeichen vorhanden ?
GONC KEYCT dann nächstes vergleichen
-----

?ST=0 0 wir sind jetzt am Ende des Strings angelangt
GOYES MKEY Option 1 nicht gesetzt ?
C=D A dann auf die nächste Taste warten
sonst 0 zurückgeben
-----

OUT D=D-C A Rückgabewert berechnen
C=D A
A=C A
-----

C=R1 D1 zurück schreiben
D1=C
-----

GOTO END fertig
-----

*****
cslw5 GOVLNG CSLW5
csrcw5 GOVLNG CSRW5
-----

*****
KEY# ein Taste auslesen und in B (A) zurückgeben
-----

CRZEX D1 und das Status Register retten
GOSUB cslw5
CDIEX
GOSUB cslw5 (es werden zwar nur drei Nibbles benötigt, aber das
C=ST spart Platz)
CRZEX
-----

?ST=0 0 Option 1 ?
GOYES KEYR50 nein, dann warte ggf. auch auf eine Taste
GOSBVL POPBUF ja, dann lies einfach den Tastaturbuffer aus
GONC KEYR72 und gib die Taste, die dort stand zurück,
GOSBVL RPTKY wenn da keine war, dann sieh nach "repeating keys"
GOC KEYR72 wir haben die Taste
B=0 A es sei denn, da war gar keine.
GONC KEYR72 B.E.T.
-----

KEYR50 diese Routine ist weitestgehend aus IDS3 abgeschrieben
-----

GOSBVL FPOLL LEX-Files die Chance geben, eine Taste zu "drücken"
CON(2) pWTKY (zB. KEYBOARD aus dem Forth-Assembler ROM)
?XM=0 wurde eine Taste "gedrückt" ?
GOYES KEYR69 dann steht sie in R0
-----

GOSBVL RPTKY repeating key ?
GOC KEYR72 dann haben wir die Tastennummer
-----

GOSBVL SETTMO setze 10 min. timeout
KEYR60 GOSBVL SLEEP und gehe schlafen (bis irgendetwas passiert)
GONC KEYR70 war es ein Tastendruck ?
GOSBVL CKSREQ sonst sieh nach wer uns geweckt hat

```

```

-----
D1=(5) PNDALM      timeout ?
C=DAT1 B
ST=C
?ST=0 3
GOYES KEYR60      wenn nicht, dann leg dich noch mal hin
C=0 A
LC(2) 99          sonst gib Taste "OFF" zurück
B=C A
GONC KEYR72      B.E.T.
-----
KEYR69 A=R0        Tastennummer von R0 nach B (A) ist vom Pollhandler
B=0 A            dort hingelegt worden
B=A B
GOC KEYR72      B.E.T.
-----
KEYR70 GOSBVL POPBUF Tastaturbuffer auslesen
GOC KEYR50      wenn da nichts war, dann noch mal warten
-----
KEYR72 CRZEX      Status und D1 zurück schreiben
ST=C
GOSUB csw5
CD1EX
GOSUB csw5
CRZEX
RTN            und zurück
-----
*****
KEYDEF          wandelt den String, wenn er in der Form "#a#b#c..."
                ist, in die entsprechenden Bytes um. Lässt sich der
                Anfang nicht Fehlerfrei umwandeln, so wird der
                gesamte String so genommen wie er ist. Tritt an
                späterer Stelle ein Fehler auf, wird eine
                Fehlermeldung ausgegeben.
-----
GOSUB GET#      fängt der String mit # und einer Zahl < 256 an ?
RTNC            wenn nicht, dann zurück (nichts wurde verändert)
GONC KEY0      wenn doch, dann haben wir die Zahl in B (A) B.E.T.
-----
KEYL GOSUB GETNUM lies nächste Zahl
GONC KEY0      kein Fehler ?
GOVLNG ARGERR  "Invalid Arg"
-----
KEY0 D1=D1- 2     schreib Byte an das Ende des umgewandelten Strings
C=B A
DAT1=C B
-----
?D#0 A         ist noch etwas umzuwandeln ?
GOYES KEYL
-----
C=0 W          Start und Länge des umgewandelten Strings nach R1
CD1EX
A=R1
C=A-C A
CSRB
GOSUB cslw5
C=A A
R1=C
RTN            und zurück

```

der nun folgende Code lies eine Zahl aus dem String und wandelt sie in eine acht Bit Binärzahl um

```

GET#                                wird nur das erste mal durchlaufen
                                     ist das erste Zeichen ein '#' ?
D0=A
D0=D0- 2
A=DAT0 B
LC(2) '#'
?A#C B
RTNYES                                nein
    
```

```

GETNUM GOSUB NZA                    lies ein Zeichen
RTNC                                         ist es keine Ziffer ?
B=A A
    
```

```

GET0 GOSUB NZA                        lies nächstes Zeichen
GONC GET1                               ist es eine Ziffer ?
?D#0 A                                  ist es nicht das Stringende ?
GOYES G0
RTN                                       wenn doch, dann sind wir fertig
    
```

```

G0 LC(2) '#'                           ist es das '#' ?
?A#C B
RTNYES                                   wenn nicht, dann zurück mit gesetztem Carry-Flag
RTN                                       sonst mit gelöschtem
    
```

```

GET1 C=B A                             es ist eine Ziffer
C=C+C A                                 die bereits gelesenen Ziffern müssen mit zehn
C=C+C A                                 multipliziert werden
B=B+C A
NIBHEX CS                               B=B+B A das ist ein Bug im Assembler
B=B+A A
?B#0 XS                                 ist der Wert größer als 255 ?
RTNYES                                   zurück mit gesetztem Carry-Flag
GONC GET0                               wenn nicht, dann nächste Ziffer B.E.T.
    
```

```

NZA                                    wandelt eine Ziffer in eine Binärzahl um
D0=D0- 2
D=D-1 A
?D=0 A                                  sind noch Zeichen da ?
RTNYES
A=0 A                                    lies ein Zeichen
A=DAT0 B
LC(2) '#'                               ist es das '#' ?
?A=C B
RTNYES
LC(2) '9'                               ist ei kleiner oder gleich "9" ?
?A>C B
RTNYES
LC(2) '0'                               ist es nicht kleiner als "0" ?
A=A-C B
RTN                                       zurück mit gesetztem Carry-Flag, wenn kleiner als "0"
                                     sonst mit gelöschtem
    
```

Mathias Rabe
Muckenbrock 14
4803 Steinhagen

CUSTUTIL

1026 Bytes, HP-71

... UND NOCH EIN LEXFILE: CUSTUTIL

Dieser Lexfile bringt einige hilfreiche Zusatzfunktionen, die man gerne im Rechner bereit hält, und die man bald nicht mehr missen möchte.

Da ist z.B. die wichtige (weil byte- und stromsparende) KEYWAIT\$-Funktion, sowie Umwandlungen von Tastennummern und Tastencodes, eine komfortable INPUT-Variante, eine Fehlertextfunktion und ein Display-Scroll Befehl.

CUSTUTIL stammt von HP direkt und ist Public Domain. Es wurde auch teilweise im "Software Developers Handbook" von HP beschrieben und ist auch in der Users Library erhältlich. Der File enthält 6 Schlüsselwörter. Die Übersicht nach LEXCHEC2 sieht so aus:

```
CUSTUTIL ID=52    SIZE=1026 Bytes
  Polls!
01 KEYWAIT$  XFN    82001 (-)
02 SCROLL    Stmt   82002
03 MSG$      XFN    82003 (#)
04 INLINE    Stmt   82004
05 KEYNAM$   XFN    82005 (#)
06 KEYNUM    XFN    82006 ($)
```

*

KEYWAIT\$:

Diese Funktion ist sehr nützlich, wenn es darum geht, den Computer im laufenden Programm anzuhalten und auf einen Tastendruck warten zu lassen. Dies ist zwar auch mit einer Schleife in BASIC realisierbar, doch hat dies den Nachteil, daß der Computer dabei im Laufmodus verbleibt und ebensoviel Strom verbraucht wie beim Abarbeiten eines Programms. Bei der Benutzung dieser Funktion jedoch schaltet der Computer in den stromsparenden Sleep-Modus um, solange er auf die Betätigung einer Taste wartet.

Die Stringfunktion lautet einfach KEYWAIT\$ (ohne Parameter) und liefert als Ergebnis die gedrückte Taste. K\$=KEYWAIT\$ ist z.B. ein Ersatz für eine BASIC-Zeile wie:

```
10 K$=KEY$ @ IF K$="" THEN 10
```

Genau wie bei KEY\$ wird als Ergebnis entweder ein Buchstabe oder eine Ziffer geliefert, oder aber bei g- und f- geschifteten Tasten und Steuertasten ein entsprechender Code.

Hier ein kleines Beispiel, wie auf einfache Weise eine Verzweigung durch Auswahl aus einem Menü in einem BASIC-Programm realisiert werden kann:

```
10 ON ERROR GOTO "ERR"
20 DISP "Vor/Zurück/Links/Rechts ?"
30 GOSUB KEYWAIT$ @ END
40 "V": A=A+1 @ RETURN
50 "Z": A=A-1 @ RETURN
60 "L": B=B-1 @ RETURN
70 "R": B=B+1 @ RETURN
80 "ERR": IF ERRN=30 THEN DISP "Nur: "; @ GOTO 20
90 ! Anm.: ERRN=30 ist "Statement not found"
100 BEEP @ DISP ERRM;ERRL @ END
```

Diese Technik der direkten Labelwahl funktioniert jedoch nur mit den einfachen Buchstabentasten A-Z. Der Grund dafür liegt in der Beschränkung durch das Betriebssystem, daß Labels nur mit Buchstaben beginnen können und keine Sonderzeichen darin enthalten sein dürfen. Da ich z.B. jedoch häufig Menüs verwende, die andere Tasten oder Tastenkombinationen er-

fordern (z.B. "fCAT" oder "a"), nehme ich hier die Funktion KEYNUM zu Hilfe, siehe weiter unten.

KEYNUM:

Die Funktion KEYNUM liefert eine Zahl zwischen 1 und 168, korrespondierend zu den Tasten-Identifikationsnummern auf Seite 123 des Benutzerhandbuches. Als Argument muß ein String dienen, der einen gültigen Tastennamen enthält. Also z.B.: "A", "#38" "fQ", "h", "g0".

KEYNAM\$:

Dies ist die genaue Umkehrfunktion zu KEYNUM, sie liefert den entsprechenden Tastennamen zur eingegebenen Tastennummer.

So ausgerüstet könnte man obiges Beispielprogramm umschreiben und durch Tasten wie z.B. "f DISP", "f DELETE", "+ " und "- " abfragen:

```
10 ON ERROR GOTO "ERR"
20 DISP "DISP/DELETE/+/- ?"
30 GOSUB "L" & STR$(KEYNUM(KEYWAIT$)) @ END
40 "L77": DISP A @ RETURN
50 "L91": A=0 @ RETURN
60 "L56": A=A+1 @ RETURN
70 "L42": A=A-1 @ RETURN
80 "ERR": IF ERRN=30 THEN DISP "Nur: "; @ GOTO 20
90 ! Anm.: ERRN=30 ist "Statement not found"
100 BEEP @ DISP ERRM;ERRL @ END
```

SCROLL:

Das Statement SCROLL dient dazu, den gerade im Display befindlichen Text an eine bestimmte Stelle zu scrollen. Außerdem wird mit der Programmausführung angehalten, bis eine Taste gedrückt wird. Das Argument hinter SCROLL kann 1 . . . 96 betragen und bezeichnet die Anzahl der Stellen, um die die Anzeige nach links zu rollen ist. Z.B.:

```
DISP "001: Hier beginnt der Text der Zeile" @ SCROLL 6
```

läßt zuerst den Text gemäß der eingestellten DELAY-Rate durch das Display rollen, hält dann aber an und zeigt in diesem Fall den 6. Buchstaben als ersten im Fenster an. Nun kann man sich mit den Cursor-left und -right Tasten den Text noch einmal nach belieben ansehen. Jede andere Taste aber läßt das Programm weiterlaufen.

MSG\$:

Diese Funktion dient dazu, aus dem Betriebssystem oder aus speziell dafür vorgesehenen Lexfiles Meldungstexte zu holen. Als Argument erwartet MSG\$ eine entsprechende Fehler- oder Meldungsnummer. Die Nummern 1 . . . 97 beziehen sich auf das Betriebssystem. Hier sind die meisten Texte enthalten, die man vom "normalen" Umgang mit dem HP-71B schon kennt, so z.B.:

```
MSG$(22)=>"Low Battery"
MSG$(30)=>"Stmt Not Found"
MSG$(54)=>"End of File"
MSG$(92)=>"Vfy: Align then ENDLN"
```

usw. Die vollständige Liste dieser Meldungen befindet sich im Referenzhandbuch auf Seite 380. Aber auch Lexfiles können Meldungen enthalten. Sie sind an der Angabe "Messages!" in der Übersicht zu erkennen. Um diese Meldungen aufzurufen, ist lediglich der Meldungsnummer die ID-Nummer des Lexfiles (dezimal) auf der 1000-er Stelle voranzustellen. Besitzt man z.B. einen HP-Texteditor, so gibt:

```
MSG$(240001)=>"Eof"
MSG$(240004)=>"Filename:"
```

Generell kann man sagen, das MSG\$ so ähnlich funktioniert wie ERRM\$, nur daß man hier willkürlich die gewünschte Meldung auswählen kann.

Eine Besonderheit stellen die Meldungsnummern 1001 ... 1097 dar. Sie werden immer dann statt der parallelen Nummern 1 ... 97 aufgerufen, wenn sich im System ein sogenannter Translator, also ein Übersetzer befindet, der anzeigt, daß alle Meldungen in eine andere Sprache als Englisch übersetzt werden sollen. Natürlich muß ein solcher Translator (ein spezieller Lexfile) auch die gesamten fremdsprachigen Meldungen enthalten.

HP regt stark an, daß jeder Programmierer möglichst regen Gebrauch von den vorhandenen Meldungen machen soll, um Programme leichter in andere Sprachen übersetzbar zu machen. Z.B.:

Statt: INPUT "Filename: "; F\$
DISP MSG\$(240004); @ INPUT " "; F\$

oder: DISP "Working . . ." DISP MSG\$(240010)

Übrigens sind die beiden Schlüsselwörter SCROLL und MSG\$ auch in anderen HP-Lexfiles enthalten, so z.B. im Forth/Assembler Modul und im Texteditor.

INLINE:

Nun zum Statement INLINE. Hier handelt es sich um eine Variante des INPUT-Befehls. Die genaue Syntax:

INLINE V\$,B1,C1,E\$,V1[,V2[,V3]]

dabei ist:

V\$ der Vorgabestring. Er wird in dem Feld angezeigt, in dem der Computer auch die Eingabe erwartet. Er kann vom Benutzer gelöscht, editiert oder übernommen werden. Wie beim INPUT-Befehl können auch sogenannte geschützte Bereiche eingebaut werden, die dann vom Cursor übersprungen werden und nicht editierbar sind.

B1 Position des ersten Buchstabens im Vorgabestrings, der im Fenster erscheinen soll. Ist normalerweise 1, sonst scrollt er um die angegebene Anzahl nach links, wie beim SCROLL-Befehl.

C1 Position, an der der Cursor stehen soll. Normalerweise 1. Wird die Zahl negativ eingegeben, so erscheint der Cursor als Insert-Cursor.

E\$ String, in dem alle Tastennamen aufgelistet sind, die den Befehl beenden. Das muß also keineswegs immer ENDLINE sein!

V1 eine numerische Variable, die angibt, die wievielte der Tasten aus E\$ zur Beendigung gedrückt wurde.

V2 gibt an, wo der Cursor bei Beendigung stand (neg.=Insert-Cursor)

V3 gibt an, wie weit der String gescrollt war bei Beendigung. Genauer: die Position des Buchstabens im String, der links im LCD stand.

Die ersten 4 Parameter sind Eingabe-, die letzten 3 Ausgabe-parameter. Der tatsächlich erfolgte Eingabestring wird zunächst nicht in eine Variable übernommen, sondern muß anschließend mit Hilfe der Funktion DISP\$ eingelesen werden. Hier ein Beispiel eines kleinen Menüs mit Hilfe von INLINE:

```
100 N$=CAT$(I) [1,8]
110 DISP "FILE? "; @INLINE N$,1,1, "#38#51#",V
120 ON V GOTO 150,130,140
130 I=I+1 @ GOTO 100
140 I=I-1 @ GOTO 100
150 F$=DISP$
```

Während also ein Filename vorgegeben wird, kann man ihn editieren, übernehmen oder mit den Pfeil nach oben oder unten Tasten einen anderen auswählen. Beendet wird der Befehl jeweils nur durch diese Pfeiltasten oder durch ENDLINE. Im letzteren Fall wird dann Zeile 150 angesprungen, in der der eingegebene Name nun endgültig in F\$ übernommen wird.

Hexdump-Listing CUSTUTIL

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

CUSTUTIL L ID#52 1026 Bytes

0123 4567 89AB CDEF ck

000:	3455	3545	5545	94C4	F0
001:	802E	0000	2120	4088	40
002:	3E70	0251	0600	0000	35
003:	F440	0000	0A90	00F2	21
004:	0B57	00FD	40EA	600D	6E
005:	2405	1600	F000	2910	34
006:	0DF0	0E85	00F0	2054	5F
007:	500F	B94E	4C49	4E45	91
008:	440D	B454	95E4	14D4	DC
009:	4250	BB45	495E	455D	57
00A:	460F	B454	9575	1494	B5
00B:	4542	107D	4357	4423	FF
00C:	0B35	3425	F4C4	C420	5A
00D:	1FF9	6950	5231	1B13	40
00E:	5112	1CD1	378B	6C11	A5
00F:	3510	B3D1	4A35	5453	AF
010:	5340	215D	D003	1C19	F8
011:	61E1	31B1	961E	4312	87
012:	F965	607B	C321	0D00	B0
013:	77A3	59F1	5B09	0C50	67
014:	5DED	0159	0071	0BDB	4E
015:	7393	31E4	7293	11B0	7F
016:	6118	D757	C7E6	350C	51
017:	1188	FC2D	E006	1721	EB
018:	3713	410B	C213	51C2	C4
019:	07D1	E51C	1137	1138	44
01A:	B6C8	1371	4B96	67E1	C5
01B:	81D9	14C8	5072	00D0	8A
01C:	07BF	2BF2	BF2E	61B7	A6
01D:	69F2	15C7	8210	38D3	1C
01E:	9450	9FFF	F553	0073	87
01F:	A27F	F58F	83DB	0D23	F6
020:	2C34	8B66	6137	D7C2	07
021:	10B1	34CE	CE10	9119	B9
022:	135D	1725	0313	2966	EA
023:	D372	40E5	3132	9663	37
024:	FCD8	A972	D230	38B1	82
025:	D113	7129	1358	FA9E	C5
026:	80D4	1811	4852	B31B	1E
027:	08D3	9390	1C11	4B13	A3
028:	7135	8BF0	0078	ADAB	95
029:	11B1	3210	2E28	AAFC	7E
02A:	109E	6E6E	6D53	2A08	3F
02B:	8FD7	9115	9B17	2112	29
02C:	1301	198F	771B	111B	60
02D:	1358	4685	77CC	1103	D3
02E:	8477	2C11	028F	13DB	43
02F:	0101	1377	CE13	1E37	82
030:	BE13	1027	3E13	1807	1F
031:	BD1A	F211	1D61	1081	3E
032:	ECE4	8206	1311	4F7C	AD
033:	B107	110E	4E41	0052	FC
034:	E70B	18D9	4B81	7B81	0F
035:	D286	640E	6155	311B	BD
036:	8F1C	6901	1231	F5D5	89
037:	1B37	4F21	4EB6	19EC	72
038:	40D4	1681	4818	314E	D9
039:	0B84	10B1	4C8F	8981	07
03A:	08F9	9C41	3181	9622	49
03B:	98F0	1500	7F11	D070	9D
03C:	5155	0B24	D61B	E74F	26
03D:	214E	F2F2	1811	4E06	32
03E:	D217	014F	96E8	1712	FB
03F:	15FA	73E0	3011	5D06	C4
040:	1AF0	68FA	3A81	75E0	5A
041:	78E3	7E50	75E0	07DA	9C
042:	7A70	78A3	7A40	71D0	4A
043:	07D5	06F5	F5D4	AC39	6C

```

044: 2880 A4FA A07E 40AC 7E
045: BACA 7873 7A10 71A0 29
046: 07D0 AEA7 1307 1638 D5
047: D84A 8014 A311 F966 67
048: FE16 1136 1341 FB88 9C
049: F214 58D8 71F0 E48F 1E
04A: B13B 1040 18FC 09E0 EE
04B: 867D 0948 8085 6AC0 C0
04C: 17F7 5C2C C4D0 D231 2A
04D: F58B A006 59D3 2A08 94
04E: 8DAB 8111 0831 B18D 95
04F: C3C1 032A 088D 14A1 6A
050: 1D97 7720 4D58 DOB7 40
051: F01B 874F 214E 0B87 27
052: 7200 B018 FD96 3031 C9
053: 1F96 2008 DB2E 208F 96
054: AB63 075E F7AD F76D 99
055: F8FA B630 7510 5AD7 06
056: E005 6077 008D 2713 33
057: 08FE A230 5001 3713 9D
058: 510A 8F9D F304 4187 94
059: 0F08 3150 5708 7300 5C
05A: 11A1 3585 48D6 6E20 71
05B: 4118 F875 C113 585A 90
05C: 8FB9 D808 F695 C117 07
05D: 1143 1C11 37CA 131D A5
05E: 479B E151 78DC 32F0 6B
05F: 8118 FC09 E017 F7B8 FF
060: 1D23 18A8 BA61 606C 19
061: D28F A90F 01CF 6CCF FB
062: 8ABD E7CC 18F4 0CA1 56
063: 100D 280F 0062 0E68 15
064: FA90 F007 80D0 1101 C8
065: 5111 CF8F E83B 17B9 6C
066: 1678 F811 7211 8EA1 76
067: 1057 0831 43D0 1000 24
068: 417F 8F62 5308 FA05 E0
069: 21FE 8FC2 B901 188F B8
06A: BA99 08D6 0890 8FFA 88
06B: CE03 4652 009B EDB1 B6
06C: 00BF 4F4F 48BE EA8F 7F
06D: 943B 1118 AD2C A8F2 46
06E: D2B1 4798 DE6A 20FE D7
06F: AFF4 FFFF 8F10 FA16 F5
070: 81F0 8F15 6811 438E 6A
071: 1700 CC54 OD0D 1AE8 11
072: 1B37 4F21 4AD2 31F5 09
073: B629 E140 D9AE A340 5B
074: 84F2 C2C2 1341 4E96 7A
075: E90C C57E D01A C74F C9
076: 1481 8314 60B8 410B C4
077: 1448 FE21 2064 FC8D 1B
078: DF8E 075F F17F 8F32 37
079: 2B14 0066 DA00 1361 12
07A: 0813 7109 8F2C 6004 BC
07B: 5111 9135 1181 348D 8C
07C: 8ACA 18F1 2700 6BDF 2C
07D: 8FD5 5911 CF13 71F9 5B
07E: 95F2 1451 3515 178D E0
07F: 8F5F 08D7 85C1 8D23 34
080: 860 34
    
```

Michael Fiedler
Friedrichstr. 17
6070 Langen

Korrekturen zu SPLINE 30 PRISMA 6/86

Erforderliche Programmkorrekturen und -verbesserungen

ETEMP0 Zeile 30

bisher

```
30 N$="GVC" @ INPUT "Knotenzahl=",
STR$(K);K @ IF K>16 THEN 30
```

neu

```
30 OPTION BASE 1 @ N$="GVC" @ INPUT "Knotenzahl=",
STR$(K);K @ IF K>16 THEN 30
```

Begründung:

OPTION BASE 0 würde ein unsinniges Ergebnis bringen

TEMPVLF0 Zeilen 710 und 1100

bisher

```
710 DIM C(K,K) @ MAT C=G*B
```

neu

```
710 DIM C(K,K),V2(K) @ MAT C=G*B
```

bisher

```
1100 DISP "Zeit = ";N*D/60;"min" @ IF K$#="#38" THEN 1100
```

neu

```
1100 DISP "Zeit = ";N*D/60;"min" @ K$=KEY$
@ IF K$#="#38" THEN 1100
```

Begründung:

In Zeile 710 ist die Dimensionierung von V2(K),
in Zeile 1100 die Gleichsetzung von K\$ mit KEY\$
bei der Reinschrift des Programmes verlorengegangen

SPLINE30 Zeile 920

bisher

```
"ERR": OFF ERROR @ IF ERRL=700 AND
ERRN=38 THEN "GRAF" ELSE RETURN
```

neu

```
"ERR": OFF ERROR @ IF ERRL=720 AND
ERRN=38 THEN "GRAF" ELSE RETURN
```

Begründung:

Bei einem RENUMBER-Befehl wurde vergessen, die durch
diesen Befehl nicht automatisch mitgeänderte Zeilennummer
von Hand auszubessern

SPLINE30 Zeile 920

bisher

```
"ERR": OFF ERROR @ IF ERRL=700 AND
ERRN=38 THEN "GRAF" ELSE RETURN
```

neu

```
"ERR": OFF ERROR @ IF ERRL=720 AND
ERRN=38 THEN "GRAF" ELSE RETURN
```

Begründung:

Bei einem RENUMBER-Befehl wurde vergessen, die durch
diesen Befehl nicht automatisch mitgeänderte Zeilennummer
von Hand auszubessern

Dr. Lutz
Bayernstr. 39
8501 Rückersdorf bei Nbg.

Funktion:

Installiert resident bis zu 26 Zweizeichenmakros für Drucker-Steuersequenzen.

Format:

[d:] [p]i[ad]ASPRN [/U: [/s] [/Mx | /Cnnn]]

Hinweise:

ASPRN ist ein residenter Makroprozessor und Editor der die Zeichen auf ihrem Weg zum Drucker abfängt. Wenn er ein Makro entdeckt – die Standardeinstellung ist eine Kombination aus der Tilde (~) und einem Buchstaben – setzt er es in die dazugehörige Steuersequenz um.

Mit ASPRN /S kann man Makros erzeugen und editieren. Mit ASPRN /Mx oder ASPRN /Cnnn kann man das Makrozeichen (Standard: ~) ändern: x kann ein beliebiges Zeichen sein, nnn die ASCII-Nummer des neuen Makrozeichens.

Mit ASPRN /U kann man ASPRN deinstallieren.

Beispiel:

Der HP LaserJet Series II benötigt die Steuersequenz
 ^[&100 ^ [10U ^ [s]p10v0s0b15T (^ [steht für ESC)
 um in Portrait-Modus umzuschalten. Viele Textprogramme können keine derart komplizierten Sequenzen verarbeiten. Mit ASPRN muß das Textprogramm nur das entsprechende Makro, z.B. ~A an den Drucker senden und ASPRN wandelt es in die Steuersequenz um.

Funktion:

Ermöglicht das Abschalten der von den geladenen residenten Programmen verwendeten Hotkeys. Diese Tastenkombinationen können dann von den normalen Anwendungsprogrammen benutzt werden.

Format:

[d:] [p]i[ad]ALLKEYS
 RESIDENT 1
 RESIDENT 2
 RESIDENT 3
 ...
 [d:] [p]i[ad]ALLKEYS
 oder
 [d:] [p]i[ad]ALLKEYS [/U]

Hinweise:

ALLKEYS muß zweimal aufgerufen werden: Einmal vor und einmal nach der Gruppe der residenten Programme, deren Hotkeys kontrolliert werden sollen.

Danach kann man mit <Ctrl-Alt-Y> (bei USA-Tastatur <Ctrl-Alt-Z>) die Hotkeys an- und ausschalten. Beim Ausschalten der Hotkeys wird ein Ton mit aufsteigender, beim Anschalten mit absteigender Frequenz ausgegeben.

Mit ALLKEYS /U läßt sich ALLKEYS deinstallieren.

Funktion:

Residenter Taschenrechner

Format:`[d:][pfad]CALC`**Hinweise:**

Nach dem Laden läßt sich CALC mit `<Alt-S>` aufrufen. Danach stehen die Rechenfunktionen zur Verfügung. Die Register sind 32-Bit breit. Im Fix-Modus stehen zwei Nachkommastellen zur Verfügung. Der Rechner bietet nur die Rechenarten, die auch die CPU bietet. Trigonometrische Funktionen, Wurzel, Quadrat, etc. sind nicht implementiert. Dafür sind aber NOT, AND, OR, XOR und SHIFT-Operationen vorhanden. Es kann hexadezimal, dezimal, oktäl und binär gerechnet werden.

Funktion:

Ermöglicht jetzt auch unter DOS 2.xx Programme aus einem anderen Verzeichnis als dem aktuellen aufzurufen.

Format:`[d:][pfad]RUN [c][s][d:][pfad] programm [argumente]`**Hinweise:**

Mit /C wechselt RUN in das angegebene Verzeichnis und startet das Programm. Mit /S bleibt RUN im aktuellen Verzeichnis und ruft das Programm auf. Voreinstellung ist /S.

DOS 3.xx erlaubt sowieso das Aufrufen von Programmen aus einem anderen Verzeichnis.

Partialbruchzerlegung

844 Zeilen, 1283 Bytes, 184 Regs., SIZE 041 + 2Z + 3NV + 8 NG, HP-41C, 4 MM

mit: Z = Zählergrad
 NV = Anzahl der voneinander verschiedenen Nenner - Nullstellen
 NG = Gesamte Anzahl der Nullstellen im Nenner (doppelte mitgezählt)

Jawohl, ich gestehe es, in der PRISMA 5/87 gab es schon einmal etwas ähnliches von mir zu diesem Thema, doch welch erbärmliches Blendwerk muß das gewesen sein, verglichen mit dieser, wirklich durchdachten und auch vollends ausgereiften Lösung des leidigen Partialbruchproblems in seiner vollständigen Allgemeinheit, dieser doch zermürbensten aller rechenfehlerträchtigen Prüfungsaufgaben – „Recht hat er, der alte Schnarcher, das muß ich haben!“ höre ich nun schon die Chöre der Programmhungrigen rufen und „Vertändle er nicht kostbare Zeit, fange er endlich an!“; so soll es sein:

Vorgegeben sei also für $a_m, b_n \neq 0$

$$f(x) = \frac{a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0}{b_n x^n + b_{n-1} x^{n-1} + \dots + b_1 x + b_0};$$

eine gebrochenrationale Funktion, deren Koeffizienten a. und b. den reellen oder auch, weil hier der allgemeine Fall abgehandelt werden soll, den komplexen Zahlen entstammen können. Ferner gelte $m < n$; d.h. der Grad des Zählers sei geringer als der des Nenners. Ist dies nicht so vorgegeben, so kann man ohne Schwierigkeit durch Polynomdivision mit Restglied eine derartige Form herstellen.

Um nun $f(x)$ integrieren oder z.B. auch invers LAPLACE-transformieren zu können, ist eine sogenannte Partialbruchzerlegung durchzuführen; was bedeutet, daß $f(x)$ als Summe ihrer Partialbrüche angeschrieben wird:

$$f(x) = \sum_{\gamma=1}^p \sum_{\mu=1}^{k_\gamma} \frac{c_{\gamma\mu}}{(x - x_{0\gamma})^\mu};$$

$x_{01}, x_{02}, \dots, x_{0p}$ sind hierbei die (möglicherweise komplexen) untereinander verschiedenen Nullstellen und k_1, k_2, \dots, k_p ihre jeweiligen Häufigkeiten.

Ein Beispiel möge dies veranschaulichen:

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

$$\frac{x^2 + 1/2}{(x - 1)^2 (x + 1)} = \frac{c_{12}}{(x - 1)^2} + \frac{c_{11}}{(x - 1)} + \frac{c_{21}}{(x + 1)}$$

Gleichung (A)

Die Nullstellen des Nenners sind $x_{01} = 1$ (zweifach) und $x_{02} = -1$ (einfach). Für die Koeffizienten c .. ergeben sich die folgenden Zahlenwerte:

$$c_{11} = 3/4; c_{12} = 5/8; c_{21} = 3/8$$

Nun gibt es natürlich viele Methoden, die Werte für c .. zu bestimmen, doch die kürzeste und eleganteste der mir bekannten ist die folgende:

Ist eine Nullstelle x_0 . des Nenners einfach (Häufigkeit 1), so genügt es, die frisch ins Leben gerufene Gleichung (A) auf beiden Seiten mit $(x - x_0)$ zu multiplizieren und den Grenzwert, gegen den die linke Seite für $x \rightarrow x_0$. strebt, zu bestimmen. Dies ist gleichzeitig der gesuchte Wert für $c_{.1}$.

Ist die Nullstelle x_0 . hingegen mehrfach (Häufigkeit k), so kann zunächst nur $c_{.1}$, der Koeffizient des Termes mit dem höchsten Grad, also

$$\frac{c_{.1}}{(x - x_0)^k}$$

bestimmt werden, indem nämlich Gleichung (A) auf beiden Seiten mit $(x - x_0)^k$ multipliziert und wieder der Grenzwert nach obigem Prinzip bestimmt wird.

Für die restlichen Glieder ist eine „Vorbehandlung“ der Gleichung (A) erforderlich: Der Term mit dem höchsten Rang (dessen Koeffizient $c_{.1}$ eben ermittelt wurde) wird auf die andere (linke) Seite der Gleichung gebracht und von der gebrochenrationalen Funktion unter Bildung des Hauptnenners subtrahiert. Anschließend vollführt man auf jeder linken Seite eine Polynomdivision durch $(x - x_0)$. Sie geht zwangsläufig auf und ergibt eine Funktion, mit deren Hilfe man den nächsten Koeffizienten derselben Nullstelle errechnen kann. Undsoweiter. Aber jetzt genug dieses jämmerlichen Gestopsels und flugs das vorher schon vorgestellte Beispiel durchgerechnet:

$$c_{11} = \lim_{x \rightarrow x_{01}} f(x) \cdot (x - 1)^2 = \lim_{x \rightarrow 1} \frac{x^2 + 1/2}{x + 1} = 3/4$$

$c_{12} = \dots$: Vorbehandlung notwendig:

$$\text{linke Seite: } f(x) - \frac{c_{11}}{(x - 1)^2} = \dots = \frac{x^2 + 1/2 - 3/4 \cdot x - 3/4}{(x - 1)^2 \cdot (x + 1)}$$

Polynomdivision:

$$\begin{array}{r} (x^2 - 3/4x - 1/4) : (x - 1) = x + 1/4 \\ \underline{x^2 - x} \\ 1/4x - 1/4 \\ \underline{1/4x - 1/4} \\ 0 \end{array}$$

Die neue linke Seite der Gleichung (A) lautet folglich:

$$f^*(x) = \frac{x + 1/4}{(x - 1)(x + 1)}$$

Hieraus ergibt sich nun der Koeffizient c_{12} :

$$c_{12} = \lim_{x \rightarrow x_{01}} [f^*(x) \cdot (x - 1)] = \lim_{x \rightarrow 1} \left[\frac{x + 1/4}{x + 1} \right] = 5/8$$

Übrig bleibt jetzt nur noch der letzte Koeffizient c_{21} , der aufgrund der Tatsache, daß x_{02} eine einfache Nullstelle ist, leicht und schnell bestimmt werden kann:

$$c_{21} = \lim_{x \rightarrow x_{02}} [f(x) \cdot (x + 1)] = \lim_{x \rightarrow -1} \left[\frac{x^2 + \frac{1}{2}}{(x - 1)^2} \right] = \frac{3}{8}$$

An diesem, noch etwas verhaltenen Rechenexempel wird rasch deutlich, daß der Arbeitsaufwand überproportional mit der Häufigkeit der Nennernullstellen steigt: „zu Fuß“ ausgerechnet wäre nicht nur enorm viel Zeit zu verwenden, sondern auch eine Eskalation der Rechenfehler unvermeidlich, besonders bei komplexen Werten, wie das nächste Beispiel zeigen wird. Ein zuverlässiges Programm ist deshalb sehr von Nutzen.

2. Beispiel:

$$f(x) = \frac{i + 2}{(x^2 + 1)}$$

Genau wie oben zerlegen wir als erstes den Nenner in seine Nullstellen ...

$$\frac{i + 2}{(x + i)(x - i)} = \frac{c_{11}}{(x + i)} + \frac{c_{21}}{(x - i)}$$

... und bestimmen hierauf die Grenzwerte:

$$c_{11} = \lim_{x \rightarrow x_{01}} [f(x) \cdot (x + i)] = \lim_{x \rightarrow -i} \left[\frac{i + 2}{x - i} \right] = \frac{i + 2}{-2i} = -\frac{1}{2} + i$$

$$c_{21} = \lim_{x \rightarrow x_{02}} [f(x) \cdot (x - i)] = \lim_{x \rightarrow i} \left[\frac{i + 2}{x + i} \right] = \frac{i + 2}{2i} = \frac{1}{2} - i$$

Kurz bevor wir nun unsere Aufgaben getrost dem Rechner anvertrauen können, noch einige Hinweise zum Programmablauf:

Das das Programm für die Verarbeitung von komplexen Zahlen ausgelegt ist, wird bei der Eingabe der Polynomkoeffizienten oder Nullstellen grundsätzlich nach Real- und Imaginärteil gefragt. Erscheint also "(I↑R)" in der Anzeige, so bedeutet dies, daß Imaginär- und Realteil durch ENTER getrennt eingetippt werden sollen und anschließend die R/S - Taste für die Fortsetzung des Programmablaufes zu betätigen ist. Um die Tastenarbeit bei ausschließlich reellen Koeffizienten (Imaginärteil gleich Null) nicht unnötig zu erschweren, wurde das Programm dergestalt ausgelegt, daß es in diesem Fall genügt, den Realteil anzugeben (ohne ENTER) und die R/S - Taste zu drücken; d.h. "(I↑R)" zu ignorieren.

Die Ausgabe der Koeffizienten c_{5u} erfolgt, falls möglich, in Bruchform (z.B. 7/11). Falls aber die Eingangsgrößen schon irrationale (z.B. $\sqrt{2}$) oder durch Rundung verfälschte rationale Zahlen (z.B. $2/3 \rightarrow 0.666666667$) waren, so erfolgt die Ausgabe im FIX-3-Format (z.B. 0.667). Wünscht man die FIX-9-Darstellung zu sehen, so genügt es, die „ \uparrow “-Taste zu betätigen. Wurden nämlich Real- und Imaginärteil der Lösung im FIX-3-Format vorgewiesen, so liegt der Realteil in X und der Imaginärteil im Y-Register. Ist hingegen nur ein Teil nicht auf Bruchform zu bringen, so liegt er in X.

Und nun zu den praktischen Anwendungen:

- 1) Wir wenden das Programm auf unsere vorher von Hand errechneten Zahlenbeispiele an:

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

```

XEQ "PBZ"
N ZAEHLER ?
2.00000 RUN
Z2= (I↑R)?
1.00000 RUN
Z1= (I↑R)?
0.00000 RUN
Z0= (I↑R)?
0.50000 RUN
ANZ. D. NST. ?
2.00000 RUN
1. NST. (I↑R)?
1.00000 RUN
GRAD ?
2.00000 RUN
2. NST. (I↑R)?
-1.00000 RUN
GRAD ?
1.00000 RUN
1. NST.
C11=3:4
C12=5:8
2. NST.
C21=3:8
    
```

$f(x) = \frac{i + 2}{(x + i)(x - i)}$

```

N ZAEHLER ?
0.00000 RUN
Z0= (I↑R)?
1.00000 ENTER↑
2.00000 RUN
ANZ. D. NST. ?
2.00000 RUN
1. NST. (I↑R)?
-1.00000 ENTER↑
0.00000 RUN
GRAD ?
1.00000 RUN
2. NST. (I↑R)?
1.00000 ENTER↑
0.00000 RUN
GRAD ?
1.00000 RUN
1. NST.
C11=-1:2+1 I
2. NST.
C21=1:2-1 I
    
```

Einige weitere Beispiele:

$f(x) = \frac{x^2 + ix + 1 - i}{(x - 1 + i)^3 \cdot (x - 1)}$

```

N ZAEHLER ?
2.00000 RUN
Z2= (I↑R)?
1.00000 RUN
    
```

```

Z1= (I↑R)?
  1.00000 ENTER↑
  0.00000 RUN
Z0= (I↑R)?
 -1.00000 ENTER↑
  1.00000 RUN
ANZ. D. NST. ?
  2.00000 RUN
1. NST. (I↑R)?
 -1.00000 ENTER↑
  1.00000 RUN
GRAD ?
  3.00000 RUN
2. NST. (I↑R)?
  1.00000 RUN
GRAD ?
  1.00000 RUN
1. NST.
C11=2+2 I
C12=3
C13=0-2 I
2. NST.
C21=0+2 I
    
```

Man bestimme das Integral

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

Der Versuch, die gebrochenrationale Funktion in ihre Partialbrüche zu zerlegen, scheitert, da der Grad des Zählers größer als der des Nenners ist:

```

N ZAEHLER ?
  3.00000 RUN
Z3= (I↑R)?
  1.00000 RUN
Z2= (I↑R)?
  0.00000 RUN
Z1= (I↑R)?
 -1.00000 RUN
Z0= (I↑R)?
  1.00000 RUN
ANZ. D. NST. ?
  2.00000 RUN
1. NST. (I↑R)?
 -2.00000 RUN
GRAD ?
  1.00000 RUN
2. NST. (I↑R)?
  2.00000 RUN
GRAD ?
  1.00000 RUN
    
```

N)=N SORRY

Aus diesem Grund wird die Funktion im folgenden durch eine Polynomdivision auf eine „angenehme“ Form gebracht:

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

Nun wird das Restglied in seine Partialbrüche zerlegt . . .

```

N ZAEHLER ?
  1.00000 RUN
Z1= (I↑R)?
  3.00000 RUN
Z0= (I↑R)?
  1.00000 RUN
ANZ. D. NST. ?
  2.00000 RUN
1. NST. (I↑R)?
 -2.00000 RUN
GRAD ?
  1.00000 RUN
2. NST. (I↑R)?
  2.00000 RUN
GRAD ?
  1.00000 RUN
1. NST.
C11=5:4
2. NST.
C21=7:4
    
```

. . . worauf die Integration ein Kinderspiel ist:

Es gilt demnach:

$$\int \frac{x^3 - x + 1}{x^2 - 4} dx = \int \left[x + \frac{5/4}{(x + 2)} + \frac{7/4}{(x - 2)} \right] dx = \frac{1}{2} x^2 + 5/4 \ln |x + 2| + 7/4 \ln |x - 2| + c$$

Und, hier am Ende dieses Artikels angelangt, möchte ich nicht versäumen, all den fleißigen Arbeitern im Hintergrund kräftig die Hand zu schütteln: Besonders hervorgeraten haben sich Thomas „Dummvogel“ Wenger, der wieder mal nur Blödsinn geliefert hat (weiter so, Thomas!), Werner „Mossimau“ Moßner und Jani „Cadillac“ Skedelj, die Korrektur lasen, sowie Angelika „Süße Maus“ Schünke, die den Autor in schweren Stunden immer wieder aufgeheitert hat. Freunde, was wär' ich ohne Euch! Ehrlich!

```

01+LBL "PBZ"
DEG CF 29 CF 27 CLX
STO 27 SIGN 35 STO 17
SIGN STO 08 + STO 13
E5 STO 26 1/X ENTER↑
+ STO 25 E3 STO 15
" N ZAEHLER ?" PROMPT
STO 00 ST+ X RCL 17 +
RCL 15 / RCL 17 +
STO 05 RCL 00 STO 07

36+LBL 01
"Z" FIX 0 ARCL 07
"↑= (I↑R)?" CLST
PROMPT STO IND 05
ISG 05 -- X<Y
STO IND 05 DSE 07 --
ISG 05 GTO 01 RCL 17
RCL 00 ST+ X + RCL 13
+ STO 06
"ANZ. D. NST. ?" PROMPT
STO 01 RCL 08 STO 18
RCL 06 RCL 01 + DSE X
RCL 15 / RCL 06 +
STO 03
    
```

SERIE 40

```

73+LBL 00
FIX 0 CLA ARCL 18
"-F. NST. (I+R)?" CLST
PROMPT RCL 06 RCL 01
+ RCL 18 ST+ X +
RCL 13 - X<>Y
STO IND Y RDN ISG X
"" X<>Y STO IND Y
"GRAD ?" PROMPT ST+ 27
STO IND 03 ISG 18 ""
ISG 03 GTO 00 RCL 00
RCL 27 X<>Y? GTO 23
"=N SORRY" PROMPT
GTO 24

```

```

110+LBL 23
DSE 27 RCL 06 RCL 01
3 * + STO 09 RCL 27
RCL 08 + ST+ X +
STO 22 LASTX + STO 23
RCL 01 STO 03 RCL 23
STO 28 RCL 08 +
STO 16

```

```

134+LBL 02
RCL 08 STO 31 STO 33
STO 34 RCL 06 RCL 01
+ RCL 03 - RCL IND X
STO 18 RCL 09 RCL 27
ST+ X + RCL 08 +
RCL 15 / RCL 09 +
STO 24 0

```

```

158+LBL 13
STO IND Y ISG Y GTO 13
X<>Y RCL 08 - INT
RCL X RCL 00 RCL 08 +
ST+ X - RCL 15 / +
STO 07 RCL 17 RCL 00
ST+ X + RCL 08 +
STO 05

```

```

183+LBL 25
RCL IND 05 STO IND 07
DSE 05 DSE 07 GTO 25
RCL 25 ST+ 24 SF 07

```

```

192+LBL 03
FS?C 07 GTO 12 RCL 06
RCL 01 + RCL X RCL 01
ST+ X + RCL 08 -
RCL 15 / + RCL 25 +
STO 05 RCL 06 STO 07
RCL 22 RCL 27 ST+ X +
RCL 08 - RCL 15 /
RCL 22 + RCL 25 +
STO 14 RCL 25 -
RCL 13 RCL 15 / + 0

```

```

232+LBL 29
STO IND Y ISG Y GTO 29
X<>Y RCL 13 - RCL 08
STO IND Y

```

```

241+LBL 11
RCL IND 07 STO 10
RCL IND 05 STO 11
RCL 05 RCL 08 +
RCL IND X STO 12
RCL 14 STO 19 RCL 06
RCL 01 + RCL 03 -
RCL 07 X=Y? GTO 26

```

```

261+LBL 22
RCL 14 STO 19 RCL 13
+ RCL IND X STO IND 19
STO 02 RCL 19 RCL 08
+ RCL X RCL 13 +
RCL IND X STO IND Z
STO 04 RCL 13 ST+ 19

```

```

280+LBL 27
RCL 04 RCL 12 *
RCL 02 RCL 11 * -
RCL 19 RCL 13 + RDN
RCL IND T STO 20 +
STO IND 19 RCL 19
RCL 08 + RCL X RCL 13
+ X<>Y RCL IND Y
STO 21 RCL 02 RCL 12
* - RCL 04 RCL 11 *
- STO IND Y RCL 20
STO 02 RCL 21 STO 04
ISG 19 GTO 27 RCL 04
RCL 12 * RCL 02
RCL 11 * - STO IND 19
RCL 19 RCL 08 +
RCL 02 RCL 12 * CHS
RCL 04 RCL 11 * -
STO IND Y DSE 10
GTO 22

```

```

342+LBL 26
ISG 07 "" ISG 05
GTO 11 RCL 22 RCL 27
ST+ X + RCL 15 /
RCL 22 + RCL 25 +
STO 07 RCL 09 RCL 08
+ RCL 15 1/X -
RCL 25 + STO 05

```

```

367+LBL 28
RCL IND 05 RCL 33 *
RCL 34 * RCL IND 07
RCL 29 * RCL 31 *
RCL 34 * - RCL 07
RCL 08 + RDN
RCL IND T RCL 30 *
RCL 31 * RCL 33 * +
STO IND 05 RCL 05
RCL 08 + RCL IND X
RCL 33 * RCL 34 *
RCL IND 07 RCL 30 *
RCL 31 * RCL 33 * -
RCL 07 RCL 08 + RDN

```

```

RCL IND T RCL 29 *
RCL 31 * RCL 34 * -
STO IND Y ISG 05 ""
ISG 07 GTO 28 RCL 33
ST+ 31 RCL 34 ST+ 31
RCL 06 RCL 01 +
RCL 01 RCL 03 - ST+ X
+ RCL IND X STO 11
X<>Y RCL 08 +
RCL IND X STO 12
RCL 09 RCL 27 ST+ X +
RCL 15 / RCL 09 +
RCL 25 + STO 10 CLX
STO 05 STO 07

```

```

460+LBL 30
RCL 10 RCL 08 +
RCL IND X X<> 07
X<> IND Y RCL IND 10
X<> 05 X<> IND 10
RCL 10 RCL 13 +
RCL 05 RCL 11 *
ST+ IND Y RCL 07
RCL 12 * ST- IND Z
RCL 10 3 + RCL 05
RCL 12 * ST+ IND Y
RCL 07 RCL 11 *
ST+ IND Z ISG 10
GTO 30

```

```

494+LBL 12
RCL 06 RCL 01 +
RCL 01 RCL 03 - ST+ X
+ RCL IND X STO 11
X<>Y RCL 08 +
RCL IND X STO 12
RCL 24 STO 14 RCL 27
STO 10 CLX STO IND 28
STO IND 16

```

```

517+LBL 14
RCL 10 X=0? GTO 21
RCL 12 RCL 11 RCL 10
XEQ 18 STO 05 X<>Y
STO 07 RCL IND 14
ST+ Y ST+ Z RDN

```

```

ST+ IND 16 X<>Y
ST+ IND 28 RCL 14
RCL 08 + RCL IND X
ST+ 05 ST+ 07 RCL 05
ST+ IND 16 RCL 07
ST- IND 28 ISG 14
DSE 10 GTO 14

```

```

548+LBL 21
RCL IND 14 ST+ IND 28
RCL 14 RCL 08 +
RCL IND X ST+ IND 16
RCL 06 RCL 01 + STO Y
RCL 01 ST+ X + RCL 08
- RCL 15 / + RCL 25
+ STO 10 RCL 06

```

```
STO 14 RCL 08 STO 05
CLX STO 07
```

```
577+LBL 15
RCL 11 RCL IND 10 -
RCL 10 RCL 08 +
RCL IND X CHS RCL 12
+ X<>Y RDN X<>Y X#0?
GTO 20 X<>Y X#0?
GTO 16 X<>Y
```

```
597+LBL 20
RCL IND 14 XEQ 18
STO 02 X<>Y STO 04
RCL 02 RCL 05 *
RCL 04 RCL 07 * -
RCL 02 RCL 07 *
RCL 04 RCL 05 * +
STO 07 X<>Y STO 05
```

```
620+LBL 16
ISG 14 -- ISG 10
GTO 15 RCL IND 28
RCL 05 * RCL IND 16
RCL 07 * + X<> IND 28
RCL 07 * CHS
RCL IND 16 RCL 05 * +
STO 21 RCL 05 X+2
RCL 07 X+2 + RCL 31
* STO IND 16 STO 32
RCL IND 28 XEQ 40
STO IND 28 STO 29 X<>Y
STO IND 16 STO 33
RCL 13 ST+ 28 ST+ 16
RCL 21 STO IND 28
RCL 32 STO IND 16
RCL IND 28 XEQ 40
STO IND 28 STO 30 X<>Y
STO IND 16 STO 34
RCL 13 ST+ 28 ST+ 16
DSE 18 GTO 03 DSE 03
GTO 02 GTO 08
```

```
679+LBL 18
STO 21 CF 06 X#0?
SF 06 CLX STO 20 SIGN
ENTER↑ FS? 06 RTN
STO 19 R↑ STO 04 R↑
STO 02
```

```
695+LBL 19
RCL 02 RCL 19 *
RCL 04 RCL 20 * -
X<> 19 RCL 04 *
RCL 02 RCL 20 * +
STO 20 DSE 21 GTO 19
RCL 20 RCL 19 RTN
```

```
716+LBL 08
TONE 9 CF 29 RCL 01
STO 03 RCL 23 STO 07
```

```
723+LBL 04
RCL 06 RCL 01 +
RCL 03 - RCL IND X
STO 18 STO 09 RCL 01
RCL 03 - RCL 08 +
CLA FIX 0 ARCL X
"-I. NST." AVIEM
```

```
742+LBL 05
CF 07 CF 08 "-C" FIX 0
RCL 01 RCL 03 -
RCL 08 + ARCL X
RCL 09 RCL 18 -
RCL 08 + ARCL X "-I="
XEQ 07 FS? 07 SF 08
STO 05 RCL 13 ST+ 07
RCL 07 RCL 08 +
RCL IND X RCL IND 07 *
X#0? GTO 09 X#0? "-I+"
X<0? "-I" XEQ 07 "-I"
```

```
780+LBL 09
FS? 08 RCL 05 FIX 9
PROMPT RCL 13 ST+ 07
DSE 18 GTO 05 DSE 03
GTO 04 FIX 9 RTN
GTO 08
```

```
794+LBL 07
RCL 07 RCL 08 +
RCL IND X RCL IND 07
X<>Y XEQ 40 RCL Y
RCL Y R-P X<>Y RDN
RCL 26 X<=Y? GTO 10
R↑ R↑ FIX 0 CF 29
ARCL Y RCL 08 X=Y?
GTO 17 "-I-" ARCL Y
```

```
820+LBL 17
RDN RTN
```

```
823+LBL 10
SF 07 R↑ R↑ / FIX 3
ARCL X RTN
```

```
831+LBL 40
RCL Y RCL Y
```

```
834+LBL 06
MOD LASTX X<>Y X#0?
GTO 06 + ST/ Z / RTN
END
```

Michael Schilli
Daucherstraße 2
8900 Augsburg

Fragezeichen?

Ich besitze einen HP 41 CX mit der Serien-Nummer 2701 S 21 117 und einen Kartenleser mit der Serien-Nr. 2712 S 80 142. Nach dem Einlesen von Programmen mit dem Kartenleser ebenso wie beim Ausführen von "GETP" oder auch von "PURFL" befindet sich auf der "Catalog"- und auf der ".I"-Taste (Taste -41 und 81) die Tastenzuweisung "X_ROM 07,37 = ABS" oder aber etwas seltener "X-ROM 00,00 = W" aber auch schon vereinzelt "dc^T>52-\$ T+NylAa ----" Wenn ich es bewußt provozieren will, geschieht natürlich nichts, aber, wenn ich es nicht beabsichtige, sind diese Funktionen plötzlich da. Was mache ich falsch?

Mit der "dc^T>52-\$T+NylAa----" -Funktion kann man offenbar auf lokale Marken zugreifen bzw. ausführen. Wahrscheinlich sind diese Zusammenhänge schon lange anderweitig klar. Wie

kommen die o.g. Tastenzuweisungen auf die und nur auf die Tasten -41 und 81? Was mache ich falsch? Wie kann ich das vermeiden? Liegt evtl. ein Fehler im Rechner vor? Schwirren irgendwelche Bytes im Rechner umher? Nach "Memory lost" konnte ich beim Ausführen der o.g. Funktionen die o.g. Tastenzuweisungen nicht erreichen. Gibt es dafür eine Erklärung?

Noch eine andere Frage zum Schluß:

Ich wäre an einer Kontaktaufnahme mit Mitgliedern in meiner näheren Umgebung sehr interessiert, wer Interesse hat, bitte bei mir melden.

Hans-Ulrich Hempel
Feldkamp 6
4400 Münster

Steuer

433 Zeilen, 1153 Bytes, 165 Regs., SIZE 023, HP-41CV, X-F, PRINTER

Berechnung des „Zu Versteuernden Einkommens“ für die Jahre 1986 bis 1988 (und voraussichtlich auch für 1989)

Hallo Freunde,
im PRISMA 1/88 hat GEORG HOFFMEIER sein Steuerprogramm für 1988 vorgestellt (und vorher schon in 87.3.28 das für 1986/87). Es erfordert – für die Berechnung der Einkommensteuer – die vorherige Bestimmung des „zu versteuernden Einkommens“ = ZVE, und das ist manchmal eine recht umständliche Prozedur.

Das beiliegende Programm berechnet nun für die Mehrheit der Steuerfälle das „ZVE“ und man kann, wenn man das HOFFMEIER-Programm hinzulädt, auch gleich die Steuer ermitteln.

Leider kommt es bedingt durch lange Krankheit, für die 87'er Steuererklärungen etwas spät. Die Steuerbescheide für dieses Jahr kommen aber erst später und die kann man dann damit nachprüfen.

Das Programm folgt dem „SCHEMA zur Selbstberechnung“ auf Seite 15/16 der „Anleitung zur Einkommensteuererklärung“. Es berechnet nach Eingabe der dort geforderten Grunddaten die aufgrund von Einkunftsarten, Familienstand, Lebensalter, Kinderzahl, Rentenversicherungsstatus usw. anzurechnenden Frei-, Pauschal- und Entlastungsbeträge bis hin zum tatsächlich zu versteuernden Einkommen (ZVE).

Die enge Anlehnung an das SCHEMA bedeutet aber auch, daß das Programm nur relativ „glatte“ Fälle rechnen kann. Für die unzähligen Sonderfall-Kombinationen wäre echte Computerkapazität (und dazu noch eine Steuerinspektorausbildung) erforderlich.

Die Programmlänge wurde so abgestimmt, daß man noch eines der von GEORG HOFFMEIER in Heft 87.3.28 (für 1986/87) bzw. 1/88 Seite 40 (für 1988/89) vorgestellten Steuerprogramme hinzuladen kann, um anschließend die Einkommensteuer zu berechnen.

(Vorsicht: Beide Programme füllen den Rechner bis zum Rand = 1 Reg. ist noch frei! Deshalb ASN-Tastenzuordnungen löschen, sonst kommt „NO ROOM“).

A. Arbeitsbereich des Programms:

Alle Einkunftsarten.

Nicht mehr untergebracht werden konnte die Berechnung der Vorsorgepauschale für Ehegatten, von denen einer RV-pflichtigen, der andere RV-freien (Beamte usw.) Arbeitslohn bezieht. (SCHEMA Zeile 54-71, s. Erl. zu den Beispielen)

B. Zeitlicher Geltungsbereich:

1986 bis 1988 (vermutl. auch 1989).
Anpassung: In Zeile 368 sind für 1986/87 = 4536 und für 1988/89 = 4752 einzusetzen = Der Haushaltsfreibetrag für Alleinstehende mit Kindern (Stkl. 2).

C. Grundsätzlich:

Die Eingabeaufforderungen erscheinen zunächst in der Anzeige und werden dann mit der Eingabe zusammen ausgedruckt. Wenn in einer Zeile kein Wert einzugeben ist, **muß eine NULL** eingegeben werden! Plausibilitätsprüfungen waren wegen der Speicherknappheit nicht möglich.

Alle Abzugsbeträge (Werbungskosten, Sonderausgaben usw.) sind positiv – ohne Minuszeichen – einzugeben.

Immer nur die **tatsächlichen** Aufwendungen eingeben, mögliche höhere Pauschbeträge berücksichtigt das Programm.

D. Erläuterungen zum Ablauf und zu den Beispielen.

Verwendete Abkürzungen zu den EST-Formularen für 1987:

ERKL 71-74 = Est-Erklärung Zeile 71-74

SCH 7 = Schema zur Selbstberechnung Zeile 7

N 3 = Anlage N zur Est-Erklärung Zeile 3

ANL 12 RO = Anleitung zur Est-Erkl. Seite 12 rechts oben

Start: XEQ „ZVE“, nach den Eingaben weiter mit R/S.

E. Beispiel 1:

Berufstätiges Ehepaar mit Kindern

1. SPLITTING = 3, NEIN = 0:

Die „3“ ist einzugeben

2. RV-FREI? J=2/N=0: Beider Arbeitsverhältnisse sind rentenversicherungspflichtig, also „0“ (s. auch Beisp. 2 für RV-Freie – Beamte u.a.)

3. LOHN? (des Ehemanns) aus N 3 und Anl. KSO 55 (Abgeordnete)

4. DARIN VERS.BEZ? Im Lohn sind keine Versorgungsbezüge lt. N 22 (Pensionen) enthalten, also „0“

5. WERB.KO.? Summe der anrechenbaren Werbungskosten aus N 37-63

6. RENTE? und DAVON STPFL.? Eingabe jeweils „0“ (s.Bsp. 2)

7. SELBST.ARB.? Hier 3680 aus Vorträgen bei einer VHS.

Wichtig: Überwiegen die Einkünfte aus selbst. Arbeit (Ärzte usw.) die der anderen Einkunftsarten, kann ein Freibetrag von 5% der Einnahmen, max. DM 1200 (nur unter Nr. 21 = SO.ABZ.?) abgesetzt wer-

den. Dies gilt unter gleichen Voraussetzungen auch für die Ehefrau.

8. EINK.KAP.V.? Zinsen, Dividenden usw. lt. Anlage KSO

9. VV+SO.EINK?±DM: Hier können untergebracht werden (in der Summe auch als Minusbetrag = CHS) Einkünfte oder Verluste aus Vermietung lt. Anlagen V und FW

– Vermietung und Verpachtung lt.

Anlagen V und FW

– Gewerbebetrieb lt. Anlage GSE

– Sonstige Einkünfte lt. Anlage KSO 45-53

– Land- und Forstwirtschaft lt. Anlage L. Die Einkünfte sind zutreffendenfalls um den Freibetrag bis zu DM 2000 (Ehegatten 4000) zu kürzen.

10. HAT EHEFR.EINK? Bei JA= 4 werden für die Ehefrau nochmals alle Einkunftsarten (Nr. 3-9) abgefragt.

11. GES(amt)BR(utto)EINK(ommen) wird ausgedruckt (vor Abzug von Freibeträgen usw.)

12. EHEM>64? Ebenso für die Ehefrau. JA=Wenn der betr. Ehegatte im Steuerjahr älter als 64 Jahre war/wurde. Steuert die Altersfrei- und -entlastungsbeträge.

13. GES.EINKUENFTE=Ausdruck gemäß SCH 22

14. SOND.AUSG.?: Sonderausgaben lt. ERKL 75-83, SCH 23-25

15. VORS.AUFW.?: Sonderausgaben lt. ERKL 63-64/67-71, SCH 28

16. AG.ANT.RV?: Arbeitgeberanteil zur ges. Rentenversicherung lt. ERKL 65-66, SCH 30; Eingabe bei RV-freien Stpfl. s. Bsp. 2

17. BAUSPK.?: Beiträge zu Bauspar-kassen lt. ERKL 73, SCH 34

18. ANGER.VORS.A.= Ausdruck der anrechenbaren Vorsorgeaufwendungen aus den Berechnungen in SCH 28-53

19. AGW.BEL.?: Außergewöhnliche Belastungen lt. ERKL 86-117

20. KINDERZAHL.?: Zur Berechnung der K-Freibeträge. Im Bsp. wurden 2,5 Kinder eingegeben, weil einem Ehegatten für ein Kind aus einer früheren Ehe nur der halbe Freibetrag zusteht.

21. SO.ABZ.?: Als sonstige Abzüge können eingesetzt werden

– der Freibetrag für Eink. aus selbst. Arbeit, s. oben Nr. 7

– Der Haushaltsfreibetrag für „Stkl. 2 ohne Kinder“, s. Bsp. 3

– Verluste aus Vorjahren lt. ERKL 84-85.

– Weitere der möglichen Abzüge, die direkt vom ZVE gekürzt werden dürfen

22. STPFL.EINK.= Ausdruck des zu versteuernden Einkommens (ZVE)

(Dann: Drucker auf NORM schal-

ten und das HOFFMEIER-Steuerprogramm aufrufen!)

F. Beispiel 2: Altersrentner mit Werkspension, RV-frei, Ehefrau hat kein Einkommen (Eventuelle Einkünfte der Ehefrau müßten ebenfalls RV-Frei sein! Siehe Einschränkung eingangs in Abs. A)

- Es werden nur diejenigen Abfragenummern erläutert, die im Beispiel 1 nicht behandelt wurden bzw. im Bsp. 2 abweichend behandelt werden müssen.

2. RV-FREI? JA=2, Eingabe = 2. Hierunter fällt der in ANL 12 RO bzw. N 31-36 aufgeführte Personenkreis, insbesondere aktive/pensionierte Beamte u.ä.; weiterbeschäftigte Altersrentner; Rentner mit Werkspension; Vorstandsmitglieder von AG's u.a.

3. LOHN? Eingabe lt. N 3

4. DARIN VERS.BEZ? Eingabe wie vorstehend (N 22), da der „LOHN“ in diesem Beispiel eine Werkspension ist.

6. RENTE?: Bruttobetrag der Rente(n) eingeben (KSO 40) DAVON STPFL.%?: Ertragsanteil der Rente in % eingeben (KSO 41). Bei mehreren Renten mit unterschiedlichen Ertragsanteilsätzen: Vorher Summe der Ertragsanteile aller Renten in DM errechnen und ihren %-Satz vom Gesamtrentenbrutto ermitteln. Dieser %-Satz kann mit beliebigen Dezimalstellen eingegeben werden. Ausgedruckt wird (Wegen FIX 0) ein gerundeter Satz, gerechnet wird mit den Dezimalen.

10. HAT EHEFR.EINK?: Eingabe „0“

12. EHEM>64?/FRAU>64?: Hier sind beide älter als 64 Jahre.

15. VORS.AUFW.?: Lt. ERKL 68-71

16. AG-ANT.RV?:

- Bei Pensionsempfängern/Altersrentnern ist „0“ einzugeben.

- Bei noch im normalen Berufsleben stehenden Angehörigen des Personenkreises lt. ANL 12 RO (N 32-36) und Abgeordneten sind 9% des Betrages aus Nr. 3 (Lohn) einzugeben.

Merkmale:

In Nr. 2 = RV-FREI? ist JA

In Nr. 3 = LOHN? ist vorhanden

In Nr. 4 = DARIN VERS. BEZ? ist „0“

G. Beispiel 3: Alleinstehende mit Kindern, RV-pflichtig

1. SPLITTING? NEIN=0

2. RV-FREI? NEIN=0

20. KINDERZAHL? In diesem Beispiel = 2. Bei SPLITTING = NEIN und Eingabe einer Kinderzahl berücksichtigt das Programm den Haushaltsfreibetrag für Alleinstehende mit Kindern.

Sonderfall: Bei „Stkl. 2 ohne Kinder“ wird die Kinderzahl mit „0“ eingegeben, der Hh-Freibetrag würde nicht gewährt. In diesem Falle ist er unter SÖ.ABZ.? (s. Nr. 21) einzusetzen, für 1986/87 mit 4536 und für 1988/89 mit 4752.

Ich habe versucht, mit nur drei Beispielen die Handhabung des Programms zu erklären, hoffentlich war's ausreichend.

Die **Druckaufbereitung** im Programm „ZVE“ (Als Neu-Mitglied - 3360 - weiß ich nun nicht, was PRISMA dazu schon gebracht hat).

Das Unterprogramm steht unter LBL D ab Zeile 397 bis END, benötigt 71 Bytes + X/F-Modul und ist - unter einer globalen Marke - als unabhängiges Unterprogramm recht nützlich.

Zweck:

Text links- und Zahl rechtsbündig auszu- drucken.

Ablauf:

- Text ins A-Reg. geben, Zahl ins X-Reg. - Übrige Stackwerte gehen verloren, also sichern. Flags oder Datenregister werden nicht benötigt.

- XEQ „DR“. Das Programm berechnet mittels ALENG/AROT die erforderlichen Leerstellen, variabel für jede Zeile, und fügt sie zwischen Text und Zahl ein.

- Nach dem Druck steht die Zahl wieder im X-Reg., das A-Reg. ist gelöscht.

- Max. Textlänge: Stellenbedarf der größtmöglichen Zahl in dieser Druckzeile ermitteln + 1. Die Differenz bis 24 steht für Text zur Verfügung.

- Ist die Zahl doch einmal länger als geplant und war die Textlänge voll ausgenutzt, kommt ein „BEEP“, die Zahl wird richtig ausgedruckt und als Text erscheint „A-REG>24“. Das Programm läuft korrekt weiter.

01*LBL "ZVE"	41 INT	81 %
02 CLX	42 4000	82 INT
03 CLRG	43 X>Y?	83 STO 19
04 X<>F	44 X<>Y	84 200
05 FIX 0	45 ST- 16	85 X>Y?
06 CF 29	46 ST- 17	86 X<>Y
07 "BERECHG.STPFL.E"	47 600	87 ST- 19
08 "EINKOMMEN"	48 RCL 17	88 "SELBST.ARB.? DM"
09 PRA	49 X>Y?	89 PROMPT
10 ADV	50 X<>Y	90 XEQ D
11 "SPLITTING=3, NE"	51 ST- 17	91 STO 20
12 "EINK=0"	52 RCL 17	92 ST+ 22
13 PROMPT	53 STO 18	93 "EINK.KAP.V.? DM"
14 XEQ D	54 400	94 PROMPT
15 X>0?	55 ST- 18	95 XEQ D
16 SF 03	56 "WERB.KO.? DM"	96 ST+ 22
17 "RV-FREI?J=2/N=0"	57 PROMPT	97 ST+ 03
18 PROMPT	58 XEQ D	98 RCL 03
19 XEQ D	59 564	99 400
20 X>0?	60 X<=Y?	100 FS? 03
21 SF 02	61 X<>Y	101 ST+ X
22 ADV	62 ST- 18	102 RCL 04
23 "EINK.EHEMANN/LE"	63 0	103 -
24 "EINK=0"	64 RCL 18	104 X>Y?
25 PRA	65 X<>Y	105 X<>Y
26*LBL A	66 X<>Y	106 ST+ 04
27 "LOHN? DM"	67 STO 18	107 RCL 03
28 PROMPT	68 "RENTE? DM"	108 RCL 04
29 XEQ D	69 PROMPT	109 -
30 STO 15	70 XEQ D	110 FC? 04
31 STO 17	71 STO 19	111 STO 02
32 STO 22	72 ST+ 22	112 FS? 04
33 "DARIN VERS.BEZ?"	73 "DAVON STPFL.	113 RCL 02
34 "EINK=0"	74 37	114 FS? 04
35 PROMPT	75 XTOA	115 -
36 XEQ D	76 "EINK=0?"	116 ST+ 20
37 ST- 15	77 PROMPT	117 "VV+SÖ.EINK.?+D"
38 STO 16	78 XEQ D	118 "EINK=0"
39 40	79 RCL 19	119 PROMPT
40 %	80 X<>Y	120 XEQ D

SERIE 40

121 ST+ 20	183 XEQ E	245 ST- 00	307 ST+ 02	369 FS? 03
122 ST+ 22	184 RCL 18	246 ST+ 01	308 FS? 03	370 0
123 FC? 03	185 RCL 09	247 RCL 00	309 ST+ X	371 ST- 20
124 GTO C	186 +	248 X<=0?	310 X<=Y?	372 "SO.ABZ.? DM"
125 FS? 04	187 RCL 19	249 GTO 02	311 X<>Y	373 PROMPT
126 GTO C	188 RCL 10	250 2	312 RDN	374 XEQ D
127 ADV	189 +	251 /	313 STO 00	375 ST- 20
128 "HAT EHEFR.EINK?"	190 +	252 INT	314 RDN	376 CLX
129 "+J=4/N=0"	191 RCL 20	253 2	315 FC? 02	377 X<>F
130 PROMPT	192 RCL 11	254 ST/ 02	316 1170	378 RCL 20
131 XEQ D	193 +	255 RDN	317 FS? 02	379 ADV
132 X>0?	194 +	256 RCL 02	318 1 E3	380 "STPFL.EINK.= DM"
133 GTO B	195 STO 05	257 X>Y?	319 RCL 02	381 XEQ D
134+LBL C	196 STO 20	258 X<>Y	320 FS? 03	382 ADV
135 FS? 04	197 ADV	259 ST- 00	321 X<>Y	383 RTN
136 XEQ E	198 "GES.EINKUENFTE="	260 ST+ 01	322 RDN	384+LBL B
137 RCL 13	199 XEQ D	261+LBL 02	323 X<=Y?	385 6.015008
138 RCL 22	200 ADV	262 RCL 17	324 X<>Y	386 REGSWAP
139 +	201 "SOND.AUSG.? DM"	263 FC? 06	325 RDN	387 SF 04
140 ADV	202 PROMPT	264 GTO 09	326 RCL 00	388 ADV
141 "GES.BR.EINK.=DM"	203 XEQ D	265 FS? 05	327 +	389 "EINKOMMEN EHEFR"
142 XEQ D	204 270	266 GTO 10	328 54	390 "+AU"
143 ADV	205 FS? 03	267 GTO 11	329 /	391 PRA
144 FS? 03	206 ST+ X	268+LBL 09	330 INT	392 GTO A
145 "EHEN>64?J=1/N=0"	207 X>Y?	269 FC? 01	331 54	393+LBL E
146 FC? 03	208 X<>Y	270 GTO 11	332 *	394 6.015008
147 "STPF>64?J=1/N=0"	209 RDN	271+LBL 10	333 300	395 REGSWAP
148 PROMPT	210 CHS	272 RCL 15	334 FS? 03	396 RTN
149 XEQ D	211 RCL 20	273 .4	335 ST+ X	397+LBL D
150 SF IND X	212 +	274 *	336 X>Y?	398 ALENG
151 X=0?	213 STO 20	275 INT	337 X<>Y	399 ARCL Y
152 GTO 01	214 "VORS.AUFW.? DM"	276 3000	338 RDN	400 ALENG
153+LBL 00	215 PROMPT	277 X<=Y?	339 RCL 01	401 23
154 RCL 15	216 XEQ D	278 X<>Y	340 X<=Y?	402 X>Y?
155 RCL 20	217 STO 00	279 RDN	341 X<>Y	403 GTO 07
156 X<0?	218 "AG-ANT.RV? DM"	280 -	342 ST- 20	404 RDN
157 CLX	219 PROMPT	281+LBL 11	343 ADV	405 ENTER↑
158 +	220 XEQ D	282 ST+ 14	344 "ANGER.VORS.A.="	406 X<> Z
159 40	221 3000	283 FC? 04	345 XEQ D	407 AROT 427 ARCL T
160 %	222 FS? 03	284 GTO 03	346 ADV	408 - 428 PRA
161 RND	223 ST+ X	285 XEQ E	347 "AGW.BEL.? DM"	409 X<>Y 429 RCL T
162 3000	224 X<>Y	286 CF 04	348 PROMPT	410 CHS 430 BEEP
163 X<Y?	225 -	287 SF 06	349 XEQ D	411 24 431 CLA
164 X<>Y	226 X<0?	288 GTO 02	350 ST- 20	412 + 432 GTO 08
165 RDN	227 0	289+LBL 03	351 0	413 STO L 433 END
166 ST- 20	228 RCL 00	290 FS? 06	352 0	414+LBL 06
167 CLX	229 X>Y?	291 XEQ E	353 FS? 01	415 "+ "
168 FS? 05	230 X<>Y	292 0	354 720	416 DSE L
169 GTO 04	231 ST- 00	293 FS? 02	355 FS? 05	417 GTO 06
170+LBL 01	232 STO 01	294 1 E3	356 720	418 RDN
171 FC? 03	233 "BAUSPK.? DM"	295 STO 02	357 +	419 AROT
172 GTO 04	234 PROMPT	296 RCL 14	358 ST- 20	420 PRA
173 "FRAU>64?J=5/N=0"	235 XEQ D	297 9	359 FIX 1	421 CLA
174 PROMPT	236 RCL 00	298 %	360 "KINDERZAHL?"	422 RDN
175 XEQ D	237 +	299 INT	361 PROMPT	423+LBL 08
176 SF IND X	238 STO 00	300 ENTER↑	362 XEQ D	424 RTN
177 X=0?	239 2340	301 ENTER↑	363 FIX 0	425+LBL 07
178 GTO 04	240 FS? 03	302 ENTER↑	364 2484	426 "A-REG>24 "
179 XEQ E	241 ST+ X	303 FC? 02	365 *	
180 GTO 00	242 STO 02	304 2340	366 ST- 20	
181+LBL 04	243 X>Y?	305 FS? 02	367 X=0?	
182 FS? 05	244 X<>Y	306 1 E3	368 4752	

August Schorr (336a)
Wildschwanbrook 42 D
2000 Hamburg 73

E-Technik

Widerstände und wie sie berechnet werden.

für PRG „EREI“ 70 Zeilen, 119 Bytes, 17 Regs., SIZE 000, HP-41C, X-F

für PRG „SPT“ zusätzlich IL Drucker 99 Zeilen, 191 Bytes, 28 Regs.

Vorbemerkung:

Widerstände, Kondensatoren o. Spulen für elektronische Schaltungen gibt es nicht in jedem beliebigen Wert. Die Internationale Elektrotechnische Kommission (IEC) hat für diese Bauelemente Normreihen entworfen nach denen sie produziert werden. Die Anzahl der Werte einer Reihe hängt von der Genauigkeit ab, mit dem die Bauelemente produziert werden sollen (Bild 1). Für Bauelemente mit 20% gilt die Reihe E6, für 10% die Reihe E12 und für 5% Genauigkeit gilt die Reihe E24. Den Wert eines Bauelementes erhält man durch Multiplikation eines Normwertes der E-Reihe mit einer Zehnerpotenz.

E6 20%	E12 10%	E24 5%
1,0	1,0	1,0
		1,1
	1,2	1,2
		1,3
1,5	1,5	1,5
		1,6
	1,8	1,8
		2,0
2,2	2,2	2,2
		2,4
	2,7	2,7
		3,0
3,3	3,3	3,3
		3,6
	3,9	3,9
		4,3
4,7	4,7	4,7
		5,1
	5,6	5,6
		6,2
6,8	6,8	6,8
		7,5
	8,2	8,2
		9,1

Bild 1

Aufgabe des Programmes „EREI“:

Bei Programmen, die zur Berechnung elektronischer Schaltungen verwendet werden, müssen die ermittelten Werte für Widerstände o. Kondensatoren anschlie-

ßend durch Werte der entsprechenden E-Reihe „von Hand“ ersetzt werden. Das ist sehr lästig, besonders dann, wenn der Programmablauf deswegen unterbrochen werden muß oder wenn die Schaltung mit den „neuen Normwerten“ noch einmal durchgerechnet werden muß, um Abweichungen von gewünschten Sollwerten der Schaltung beurteilen zu können.

Das Programm „EREI“ schließt nun diese Lücke. Es verwandelt eine im X-Reg. stehende Zahl in einen Normwert gleicher Zehnerpotenz. Die Werte der E-Reihen stehen in einem 25 Register großen Datenfile „E24“ im erweiterten Speicher. Eine Steuerung des Programmes „EREI“ erfolgt mit den Flags 00...03. Flag 00 u. 01 wählen die E-Reihe aus.

Flag 01 , Flag 00	Ausgewählte E-Reihe	
0	0	E24
0	1	E12
1	X	E6

Die Flags 02 und 03 beeinflussen die Auswahl b.z.w. kommentieren das Ergebnis.

Bedeutung der Flagzustände vor dem Programmstart

Flag 03 , Flag 02	Bedeutung	
0	0	nächstgelegenen Wert auswählen
0	1	nächstniedrigeren Wert auswählen
1	0	nächsthöheren Wert auswählen

Bedeutung der Flagzustände nach dem Programmdurchlauf

Flag 03 , Flag 02	Bedeutung	
0	1	es wurde ein niedriger Wert gewählt
1	0	es wurde ein höherer Wert gewählt

Beispiele:

Eingabe Flag 00...03 = 0	X=426,87
Ausgabe Flag 00...02 = 0	
Flag 03 = 1	X=430
Eingabe Flag 00 = 1	
Flag 01...03 = 0	X=7,44
Ausgabe Flag 00 , 03 = 0	
Flag 01 , 02 = 1	X=6,8
Eingabe Flag 01 , 03 = 1	
Flag 00 , 02 = 0	X=16,37 E-6
Ausgabe Flag 01 , 03 = 1	
Flag 00 , 02 = 0	X=22,00 E-6

Beschreibung des Programmes „EREI“

- 1... 5 Das Datenfile wird zum Arbeitsfile.
- 6...11 Von der eingegebenen Zahl wird der Zehnerexponent entfernt. Der Zahlenwert der Eingabe steht jetzt in Reg. M (C). Zeile 11 \triangleq 127,00
- 12..14 Der Zehnerexponent wird berechnet und isoliert vom Zahlenwert in Reg. N (\) abgelegt.
- 15...25 Für den Suchvorgang wird die günstigste Startadresse des Datenfiles mit der Formel $Adr.=4*INT(6*LOG(Zahlenwert -0,2))$ bestimmt. Benachbarte Werte einer E-Reihe unterscheiden sich etwa durch den Faktor $\sqrt[n]{10}$. n=6 bei E6, n=12 bei E12 u.s.w. .

Umgestellt erkennt man ungefähr die Position des gesuchten Wertes in der E-Reihe.

- 26...27 Damit es keine Unordnung gibt.
- 28...43 Hier werden die Normwerte bestimmt die ober und unter dem Zahlenwert liegen
- 31...39 Es wird dafür gesorgt, daß nur Zahlen der gewünschten Reihe genommen werden.
- 44...69 Die Flags 02 u. 03 werden entsprechend gesetzt und die Ergebniszahl aus Normwert und gespeicherter Zehnerpotenz zusammengesetzt.

Das Datenfile „E24“

Wie schon erwähnt benötigt das Programm „EREI“ die Zahlen der E24-Reihe in einem Datenfile im X-M.

Das Hilfsprogramm „C-E24“ erzeugt das benötigte File „E24“. Einfach das Hilfsprogramm laden, starten und wieder löschen. Fertig!

Anwendungsbeispiel für das Programm „EREI“

In einer Spannungsteilerschaltung (Bild 2) sollen die Widerstände R1 u. R2 berechnet werden. Die Größe der Widerstände bestimmt die maximale Ausgangsspannung U3 und die minimale Ausgangsspannung U2.

Das Programm „SPT“ fragt nach den Werten für U1, U2 u. U3 sowie nach dem Widerstand des Potentiometers. Es berechnet daraus die Werte für R1 u. R2 und ermittelt die nächstgelegenen Normwerte aus der E-12 Reihe. Zum Schluß werden noch die tatsächlich einstellbaren Spannungen U2 u. U3 berechnet und mit ihrer prozentualen Abweichung vom gewünschten Spannungswert angezeigt.

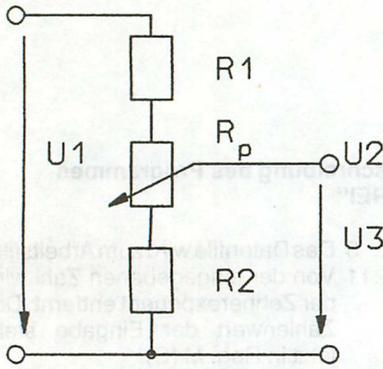


Bild 2

Die Ausgabe der Werte erfolgt über den Drucker. In Bild 3 sehen Sie noch zwei Beispiele für die Ausgabe.

```

U1= 24.00E0
U2= 11.00E0
U3= 19.00E0

RP= 22.00E3

R1= 13.75E3 > 15.00E3
R2= 30.25E3 > 33.00E3
U2= 11.31E0 : 2.86%
U3= 18.86E0 : -0.75%
    
```

Bild 3a

```

01*LBL "C-E24"      32 2,7
02 "E24"            33 SAVEX
03 CLX              34 3
04 SF 25            35 SAVEX
05 SEEKPT           36 3,3
06 FS?C 25          37 SAVEX
07 RTN              38 3,6
08 25               39 SAVEX
09 CRFLD            40 3,9
10 CLX              41 SAVEX
11 SEEKPTA          42 4,3
12 1                 43 SAVEX
13 SAVEX            44 4,7
14 1,1              45 SAVEX
15 SAVEX            46 5,1
16 1,2              47 SAVEX
17 SAVEX            48 5,6
18 1,3              49 SAVEX
19 SAVEX            50 6,2
20 1,5              51 SAVEX
21 SAVEX            52 6,8
22 1,6              53 SAVEX
23 SAVEX            54 7,5
24 1,8              55 SAVEX
25 SAVEX            56 8,2
26 2                57 SAVEX
27 SAVEX            58 9,1
28 2,2              59 SAVEX
29 SAVEX            60 10
30 2,4              61 SAVEX
31 SAVEX            62 END
    
```

```

U1= 36,00E0
U2= 14,00E0
U3= 18,50E0

RP= 470,00E0

R1= 1,828E3 > 1,800E3
R2= 1,462E3 > 1,500E3
U2= 14,32E0 : 2.31%
U3= 18,81E0 : 1.68%
    
```

Bild 3b

```

01*LBL "SPT"
02 ENG 3
03 "U1= "
04 XEQ 00
05 STO 01
06 "U2= "
07 XEQ 00
08 STO 02
09 "U3= "
10 XEQ 00
11 STO 03
12 ADV

13*LBL 03
14 "RP= "
15 XEQ 00
16 STO 00
17 ADV
18 RCL 01
19 RCL 03
20 -
21 RCL 00
22 *
23 RCL 03
24 RCL 02
25 -
26 /
27 STO 04
28 LASTX
29 1/X
30 RCL 00
31 *
32 RCL 02
33 *
34 STO 05
35 RCL 04
36 "R1= "
37 XEQ 01
38 STO 04
39 RCL 05
40 "R2= "
41 XEQ 01
42 STO 05
43 RCL 00
44 RCL 05
45 +
46 ST+ 04
47 RCL 01
48 *
49 RCL 04
50 /

51 X<> 05
52 RCL 01
53 *
54 RCL 04
55 /
56 RCL 02
57 "U2= "
58 XEQ 02
59 RCL 05
60 RCL 03
61 "U3= "
62 XEQ 02
63 ADV
64 ADV
65 ADV
66 ADV
67 GTO 03

68*LBL 00
69 PROMPT
70 ARCL X
71 PRA
72 RTN

73*LBL 01
74 ARCL X
75 "+ > "
76 ACA
77 1
78 X<>F
79 X<>Y
80 XEQ "EREI"
81 ARCL X
82 ACA
83 PRBUF
84 RTN

85*LBL 02
86 ARCL Y
87 "+ : "
88 -
89 LASTX
90 /
91 100
92 *
93 FIX 2
94 ARCL X
95 "+%"
96 ENG 3
97 PRA
98 RTN
99 END

01*LBL "EREI"
02 ENG 3
03 "E24"
04 5
05 SEEKPTA
06 X<>Y
07 STO [
08 X<>Y
09 AROT
10 X<>Y

11 "F++"
12 RCL [
13 /
14 STO \
15 X<> L
16 X<>Y
17 1/X
18 -
19 LOG
20 6
21 *
22 INT
23 4
24 *
25 SEEKPT
26 FS? 01
27 CF 00
28 GETX
29*LBL 00
30 STO J
31 CLX
32 ENTER↑
33 RCLPT
34 FS? 00
35 1
36 FS? 01
37 3
38 +
39 SEEKPT
40 RCL [
41 GETX
42 X<=Y?
43 GTO 00
44 FS? 03
45 GTO 01
46 X<> J
47 FS? 02
48 GTO 01
49 X=Y?
50 GTO 01
51 -
52 LASTX
53 X<> [
54 RCL J
55 -
56 ABS
57 X<=Y?
58 GTO 02
59 SF 02
60 RCL [
61 GTO 01
62*LBL 02
63 SF 03
64 RCL J
65*LBL 01
66 RCL \
67 *
68 RND
69 CLA
70 END
    
```

Gehrungsprogramm

197 Zeilen, 280 Bytes, 40 Regs., SIZE 039, HP41 CV

Ich stelle hier ein Programm zur Verfügung, daß das Herz eines jeden Schreibers höher schlagen läßt. Das Programm berechnet die benötigten Maße und Winkel zur Herstellung trichterförmiger Gehäuse nach Eingabe der Maße G1, G2, F, K und Winkel Alfa.

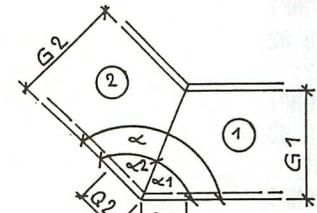
Die Berechnung geht von der Verwendung gleich dicken Materials aus. Bei einer normalen Gehrung ($G1 = G2$) erübrigt sich die Eingabe von G2. Wenn $G1 \neq G2$ wird die innere oder äußere Gehrungskante bündig, die Differenz wird mit Maß W angezeigt. Sollten beide Kanten bündig sein muß ungleich dickes Material verwendet werden.

Ob der flache oder der steile Teil mit 1, beziehungsweise 2 bezeichnet wird ist nicht von Bedeutung.

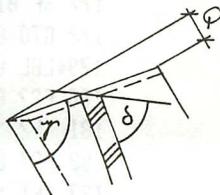
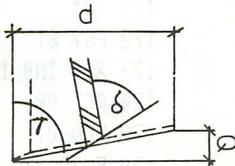
Nach Beendigung der Ausgabe kann neu begonnen, oder die Ausgabe mit R/S wiederholt werden.

- G1? Ausladung eines beliebigen Teiles im Grundriss
- G2? Ausladung des zu G1 gehörenden Teiles
- F? Höhe im Frontalschnitt
- A1? Gesamtwinkel Alfa im Grundriss, 0-180 Grad
- K? Materialdicke

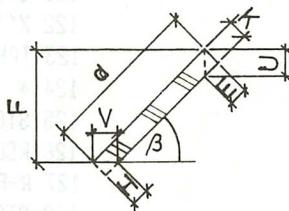
Mögliche Gehrungstypen



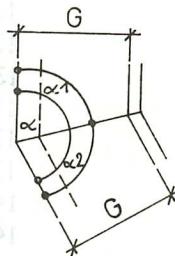
Normale Gehrung
 $G1 = G2$
 $\alpha_1 = \alpha_2$
 $\alpha = C-180 \text{ Grad}$



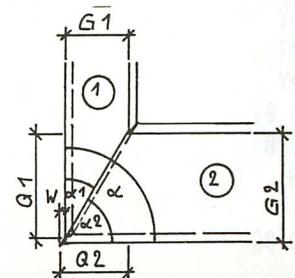
Ausklappung



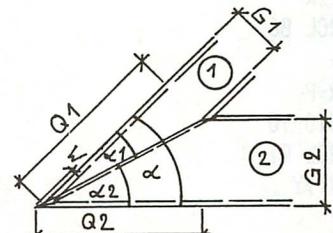
rechtwinkliger Schnitt



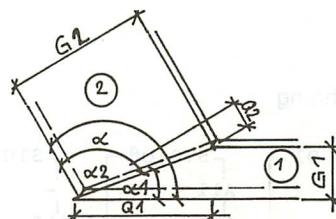
Grundriss



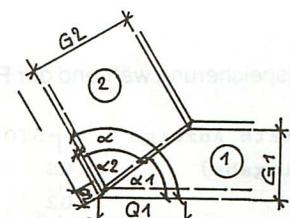
Falsche Gehrung
 $G1 \neq G2$
 $\alpha_1 \neq \alpha_2$
 $\alpha = 90 \text{ Grad}$
 rechtwinklig



Falsche Gehrung
 $G1 \neq G2$
 $\alpha_1 \neq \alpha_2$
 $\alpha = 0-89,9 \text{ Grad}$
 spitzwinklig



Falsche Gehrung
 $G1 \neq G2$
 $\alpha_1 \neq \alpha_2$
 $\alpha > 90 < 180 \text{ Grad}$
 $\gamma > 90 \text{ Grad}$



Falsche Gehrung
 $G1 \neq G2$
 $\alpha_1 \neq \alpha_2$
 $\alpha > 90 < 180 \text{ Grad}$

EINGABE		SPEICHER
F?		02
K?		03
A1?	α	04
G 1?		00/01
G 2?		00/01
AUSGABE		
Q 1		15
d I		16
Be 1	β I	17
Δ 1	Freiwinkel	18
Ga 1	γ I	19
Δ 1	Freiwinkel	20
H 1		21
V 1		22
U 1		23
E 1		24
De 1	δ I	25
Δ 1	Freiwinkel	26
I 1		27
Q 2		28
d 2		29
Be 2	β 2	30
Δ 2	Freiwinkel	31
Ga 2	γ 2	32
Δ 2	Freiwinkel	33
H 2		34
V 2		35
U 2		36
E 2		37
W 2		38
A1 I	α 1	08/I2
A1 2	α 2	08/I2

01+LBL "+"	51 X>Y?	101 "V"	151 SF 00
02 E	52 GTO 11	102 XEQ IND 11	152 XEQ IND 11
03 STO 11	53 SF 02	103 RCL 09	153 TAN
04 X<> d	54 RCL 00	104 /	154 RCL 03
05 SF 27	55 RCL 01	105 "U"	155 *
06 FIX 2	56 X>Y?	106 XEQ IND 11	156 "I"
07 "F?"	57 SF 03	107 RCL 03	157 XEQ IND 11
08 PROMPT	58 RCL 08	108 RCL 09	158+LBL 05
09 STO 02	59 CHS	109 /	159 RCL 07
10 "K?"	60 100	110 "E"	160 RCL 09
11 PROMPT	61 +	111 XEQ IND 11	161 ATAN
12 STO 03	62 FS? 03	112 RCL 04	162 COS
13 "A!?"	63 STO 08	113 RCL 08	163 RCL 03
14 PROMPT	64+LBL 11	114 -	164 *
15 STO 04	65 RCL 08	115 STO 12	165 FS? 01
16 "G1?"	66 RCL 10	116 FS? 01	166 -
17 PROMPT	67 P-R	117 GTO 05	167 STO 07
18 "G2?"	68 "Q"	118 RCL 02	168 RCL 12
19 PROMPT	69 XEQ IND 11	119 RCL 10	169 TAN
20+LBL 02	70 RCL 02	120 R-P	170 /
21 STO 00	71 RCL 01	121 STO 06	171 "W"
22 X<>Y	72 R-P	122 X<>Y	172 FS? 01
23 X=0?	73 "d"	123 TAN	173 XEQ IND 11
24 X<>Y	74 XEQ IND 11	124 *	174 RCL 00
25 STO 01	75 X<>Y	125 STO 05	175 RCL 01
26 RCL 04	76 "Be"	126 RCL 06	176 FC?C 01
27 SIN	77 SF 00	127 R-P	177 SF 01
28 /	78 XEQ IND 11	128 STO 13	178 GTO 02
29 STO 06	79 RDN	129 RCL 08	179+LBL 01
30 15	80 X<>Y	130 TAN	180 FS? 01
31 FC? 01	81 /	131 *	181 "+2 "
32 STO 14	82 ATAN	132 RCL 05	182 FC? 01
33 RCL 04	83 FC?C 03	133 /	183 "+1 "
34 RCL 00	84 GTO 11	134 ATAN	184 ABS
35 LASTX	85 CHS	135 RCL 12	185 ARCL X
36 /	86 100	136 TAN	186 X=0?
37 P-R	87 +	137 RCL 13	187 PROMPT
38 RCL 06	88+LBL 11	138 *	188 STO IND 14
39 +	89 "Ca"	139 RCL 05	189 ISG 14
40 R-P	90 SF 00	140 /	190 ""
41 STO 10	91 XEQ IND 11	141 ATAN	191 FC?C 00
42 RCL 01	92 R↑	142 +	192 RTN
43 X<>Y	93 TAN	143 2	193 90
44 /	94 STO 09	144 /	194 -
45 ASIN	95 RCL 03	145 ABS	195 "Δ"
46 STO 08	96 *	146 90	196 XEQ IND 11
47 RCL 04	97 "H"	147 X<>Y	197 END
48 -	98 XEQ IND 11	148 FS? 02	
49 ABS	99 RCL 03	149 -	
50 90	100 R-P	150 "De"	

Umspeicherung während der Programmausführung

Letzte Anzeige (Eingabe)	STO 00	STO 01	STO 08	STO 12
De	G2	GI	A11	-
Q	G2	GI	A11	A12
w	GI	G2	A11	A12
Q	GI	G2	A12	A11
Q	G2	GI	A11	A11

Roger Heizig
Fürstenlandstr. 5a
CH-9500 Wil

Neue Display Charakter

HP-41CX, Rambox oder Ähnliches, Assembler

Wer Ken Emery's „M Code for Beginners“ schon gelesen hat, weis es schon: Das neue Display (abgerundete Ecken) kann 48 weitere Character darstellen. Nicht alle Character sind neu, aber unter den Neuen sind die, die man sich schon immer gewünscht hat: Alle Kleinbuchstaben! Erstaunlich finde ich, daß HP das Betriebssystem nicht gleich an das neue Display angepaßt hat. Fast noch erstaunlicher ist, daß W&W die Revision B des CCD-Moduls nicht angepaßt hat.

Ich habe mir für's Erste eine kleine Prompting-Funktion ausgedacht, die im wesentlichen dazu dient, sich alle neuen Kleinbuchstaben anschauen zu können.

Betrachtet man die LCD-Character-Tabelle, so gelten folgende Abbildungsvorschriften:

- 0 → 12
- 1 → 13
- 2 → 10
- 3 → 11
- 10 → 11 (a,b,c,d,e bleiben erhalten)

Die Funktion hat ein „alpha input only“-Format (z.B. LBL, GTO). Man kann bis sieben Character eingeben, die dann in Kleinbuchstaben oder andere Zeichen übersetzt werden.

Hier nun die Funktion:

- 0C1 a:
- 23E >
- 101 A
- 000 NOP
- 04E C=0 ALL
- 270 RAMSLCT
- 278 READ 9(Q)
- 10E A=C ALL
- 3D9 ?NC XQ
- 01C 07F6 (ENLCD)
- 3C1 ?NC XQ
- 0B0 2CF0 (CLLCD)
- 346 ?A≠0 S&X
- 073 JNC+14
- 130 LDI
- 060 HEX: 060
- 206 C×C+A S&X
- 056 C=0 XS
- 236 C=C+1 XS
- 3D8 C<>ST
- 144 CLRF 6
- 284 CLRF 7
- 3D8 C<>ST
- 3E8 SLSABC (WRITE 15(e))
- 38E RSHFA ALL
- 38E RSHFA ALL
- 393 JNC-14
- 060 POWOFF
- 000 NOP
- 3E0 RTN

LCD CHARACTER TABLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
2	:	"	#	\$	%	&	'	<	>	*	÷	{	}	~	/	
3	0	1	2	3	4	5	6	7	8	9	█	,	∠	=	∧	∇
10	f	g	b	c	d	e	t	r	z	x	z	z	z	z	z	z
11	π	e	β	τ	λ	φ	-	∩	∪	∩	∪	∩	∪	μ	ε	λ
12	τ	ω	b	c	d	e	f	g	h	i	j	k	l	m	n	o
13	p	q	r	s	t	u	v	w	x	y	z	{	}	~	/	

Die neuen Character lassen sich auch für Funktions- und Modulnamen benutzen, da die Funktion PROMFC(05C7) auch Special Character darstellen kann. Message Lines können nicht auf die Funktion MESSL(07EF) zurückgreifen, wenn sie Special Character beinhalten. MESSL ist dafür nicht vorgesehen.

Marcus Otto (3721)
Argentinische Allee 177
1000 Berlin 37

FLCS

104 Zeilen, 182 Bytes, 26 Regs., SIZE 000, HP-41C, CCD, 1 X-F

Ich habe mein Programm in Anlehnung an das Programm CHK aus der Beschreibung des CCD-Module-Handbuches geschrieben. Dazu habe ich das Programm FLHD von Michael Kamp, aus PRISMA Nr. 6/87, etwas abgeändert und gleich in mein Programm zur Berechnung der File-Header-Adresse implementiert, da dieses Programm sehr schnell die nötige Adresse berechnen kann.

nicht vollständig in einem X-Memory-Bereich liegt, sondern verteilt ist auf zwei oder sogar auf drei Bereiche.

Das Programm berechnet die richtige Prüfsumme eines Files im erweiterten RAM-Bereich und korrigiert anschließend diese Summe, damit das File wieder in den Hauptspeicher zurückgeholt werden kann.

Es wird nur der Stack verändert. Im ALPHA-Register ist nach der Ausführung der Filename enthalten.

Anwendung:

1. XEQ FLCS
2. Auf die Anfrage FL NAME: __ Name des zu korrigierenden Files eingeben
3. Beim Signal BEEP ist das File korrigiert und man kann es in den Arbeitsspeicher übertragen.

Es ist darauf zu achten, daß nur Filenamen mit dem Kennbuchstaben „P“ mittels diesem Programm korrigiert werden. Ansonsten ist es möglich, daß Fehler entstehen und Informationen überschrieben werden.

01*LBL "FLCS"	23 GTO 01	45 -	67 BEEP	89 X<>Y
02 SF 27	24 CLX	46 CHS	68 RTN	90 RTN
03 "FL NAME! "	25 STO J	47 STO Z	69*LBL 10	91*LBL 20
04 PNTA	26 RDN	48 ISG Y	70 RCL Z	92 A+B
05 CF 27	27 RCLPTA	49*LBL 00	71 DSE X	93 PEEKB
06 FLSIZE	28 STO \	50 RDN	72 STO Y	94 X<>Y
07 64	29 CLX	51 A-	73 0	95 A-
08 STO \	30 2	52 PEEKB	74 XEQ 13	96 PEEKB
09 5	31 XEQ 13	53 ST+ J	75 GTO 00	97 X<>Y
10 XEQ 20	32*LBL 00	54 DSE Z	76*LBL 13	98 CLX
11 X<> \	33 STO Z	55 GTO 00	77 -	99 16
12 191	34 X<>Y	56 RCL \	78 X>Y?	100 ST* Z
13 0	35 -	57 X#0?	79 RTN	101 /
14*LBL 01	36 7	58 GTO 10	80 X<>Y	102 INT
15 XEQ 13	37 *	59 RCL Z	81 -	103 +
16 PEEKR	38 RCL \	60 A-	82 LASTX	104 END
17 STO J	39 -	61 RCL J	83 2	
18 CLX	40 X>0?	62 256	84 XEQ 20	
19 FLSIZE	41 CLX	63 MOD	85 +	
20 2	42 CHS	64 POKEB	86 LASTX	Peter Lüthi
21 +	43 STO \	65 CLX	87 238	Blickensdorferstr. 20
22 DSE \	44 LASTX	66 STO J	88 -	CH 6312 Steinhausen

XROM Version 2

124 Zeilen, 207 Bytes, 030 Regs., SIZE 002, HP-41CV, (IL, PRINTER)

Programm zur Ermittlung aller verwendeten XROM-Funktionen innerhalb eines Programms. (Nach einer Anregung aus PRISMA 1/88 p37)

Hier ist nun eine korrigierte Version meines Programms aus dem Prisma 3/88 S. 36. Diese Version hat an kritischen Stellen 3 Byte GOTO's für Sprünge über 112 Bytes hinweg. Dadurch wird das Programm wesentlich schneller ausgeführt.

Funktionsweise:

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

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

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

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

01*LBL "XROM"	29 GTO 04	- ja, > Multibyte-Behandlung
02 CLX	30 112	- nein
03 STO 00	31 -	Zeilen 2 bis 8 ? (32 bis 143)
04 SF 27	32 X<0?	
05 "PRGM? "	33 GTO 02 (3-Byte)	- ja, > 1-Bytebefehl
06 PNTA	34 16	- nein
07 PHD	35 -	Zeile 9 ? (144 bis 159)
08 PPLNG	36 X<0?	
09 CHS	37 GTO 05	- ja, > 2-Bytebefehl
10 A+B	38 7	- nein
11 STO 01	39 X<Y?	XROM-Befehl ? (160 bis 167)
12 PHD	40 GTO 03	- nein
	41 RDN	- ja
13*LBL 01	42 4	aus erstem Byte die XROM Nr berechnen
14 ISG 00	43 *	
15*LBL 00	44 X<>Y	Adresse des 2. Bytes
16 PEEKB	45 A-	2. Funktionsbyte lesen
17 16	46 PEEKB	
18 -	47 64	welche der 4 Möglichkeiten ist es ?
19 X<0?	48 /	(0..63 64..127 128..191 192..255)
20 GTO 02 (3-Byte)	49 INT	zur XROM Nr addieren
21 13	50 ST+ Z	
22 -	51 CLA	
23 X<0?	52 FIX 0	
24 GTO 11 (3-Byte)	53 ARCL 00	Zeilennummer ins ALPHA-Register
25 CF 00	54 FIX 2	
26 3	55 " XROM "	Append "XROM"
27 -	56 RDN	
28 X<0?	57 LASTX	aus dem zweiten Byte die Funktionsnummer

Datenbarcodes zu BANNER

Heft 3/88, Seite 39

Wie vielen vielleicht schon aufgefallen ist, fehlten im letzten Heft die Datenbarcodes zum Programm BANNER, deren Eingabe per Hand wohl unzumutbar sein dürfte.

Die Barcodes hatte es ganz einfach noch nicht gegeben, das ist die Erklärung, da das Programm zum Ausdruck noch nicht fertig war.

```

58 FRC          berechnen
59 .64          (Funktionsnummer = 0 bis 63)
60 *
61 RCL Z        XROM Nr abrufen
62 +           zur Funktionsnummer addieren
63 ARCL X       ins ALPHA-Register laden
64 AVIEW        anzeigen/drucken
65 GTO 02

66*LBL 03      hier weiter, wenn kein XROM-Befehl
67 -
68 25
69 -           Rest Zeile A und Zeile B ? (168 bis 191)
70 X<0?
71 GTO 05      - ja, > 2-Bytebefehl
72 14
73 -           GLOBAL-Befehl ? (192 bis 205)
74 X<0?
75 GTO 06      - ja, > GLOBAL
76 2
77 -           letzte 2-Bytebefehle ? (206 und 207)
78 X<0?
79 GTO 05      - ja
80 32
81 -           Zeilen D und E ? (208 bis 239)
82 X<0?
83 GTO 07      - ja, > 3-Bytebefehle
84 GTO 08      - nein, > TEXT

85*LBL 05      2-Bytebefehle
86 RDN
87 1           Schleifensteuerung
88 GTO 08

89*LBL 06      GLOBAL-Befehle (3. Byte gibt Länge an)
90 RDN
91 A-          Adresse auf 2. Byte
92 GTO 09

93*LBL 07      3-Bytebefehle
94 RDN
95 2           Schleifensteuerung
96 GTO 08

97*LBL 04      Multibyte (2. Byte gibt Länge an)
98 RDN
99*LBL 09
100 A-         Adresse 1 Byte tiefer
101 PEEKB      Byte holen
102 240
103 -         Länge berechnen (wird als Steuerzahl
              verwendet)

104*LBL 08
105 X<>Y       Steuerzahl und Adresse vertauschen
106*LBL 10
107 A-         Adresse erniedrigen
108 DSE Y      noch mehr ?
109 GTO 10     - ja
110 X<>Y       - nein, Steuerzahl und Adresse t.
111 GTO 02     weiter

112*LBL 11     Zahleingaben (Folge von 1-B.bef.)
113 FS? 00    letztes Byte auch schon Zahl ?
114 DSE 00    - ja, > Zeilennr wieder erniedrigen
115 SF 00     Flag 00 setzen = Zahleingabe

116*LBL 02     nächste Byteadresse berechnen
117 X<>Y
118 A-
119 RCL 01     Adresse des Programmendes
120 X<>Y
121 X>Y?      noch nicht fertig ?
122 GTO 01 (3-Byte) - ja, > weiter in der Schleife
123 CLD       fertig, Anzeige löschen
124 END

207 Bytes 2 Datenspeicher

(C) 5/88 CCD 2970
    
```

Text: 1 15 9 15 1 Nr. 0



Text: 46 2A 12 2A 46 Nr. 1



Text: 9 1D 2B 9 9 Nr. 2



Text: 39 45 45 39 45 Nr. 3



Text: 7F 16 26 26 1B Nr. 4



Text: 80 2 2 2 4 Nr. 5



Text: 11 21 80 21 11 Nr. 6



Text: 61 59 47 59 61 Nr. 7



Text: 39 45 45 3D 5 Nr. 8



Text: 9 1D 3F 1D 9 Nr. 9



Text: 63 15 9 11 61 Nr. 10



Text: 41 3D 21 21 1D Nr. 11



Text: 61 51 59 65 43 Nr. 12



Text: 11 9 79 9 5 Nr. 13



Text: 9 56 80 56 9 Nr. 14



Armin Jakob
Obere Felsenstraße 10
9000 St. Gallen

Text: 3F 4A 4A 4A 3F Nr. 15



Text: 5F 22 2 22 5F Nr. 16



Text: 31 4B 4E 4A 31 Nr. 17



Text: 79 15 16 15 79 Nr. 18



Text: 39 45 46 3D 41 Nr. 19



Text: 79 16 15 16 79 Nr. 20



Text: 39 46 45 3E 41 Nr. 21



Text: 3D 44 43 44 3D Nr. 22



Text: 39 46 45 46 39 Nr. 23



Text: 3F 42 41 42 3F Nr. 24



Text: 3D 42 41 42 3D Nr. 25



Text: 7F A 80 4A 4A Nr. 26



Text: 39 45 39 55 59 Nr. 27



Text: 15 35 1D 17 15 Nr. 28



Text: 49 7F 4A 42 23 Nr. 29



Text: 56 2B 56 2B 56 Nr. 30



Text: 1 1 1 1 1 Nr. 31



Text: 1 1 60 1 1 Nr. 32



Text: 1 4 1 4 1 Nr. 33



Text: 15 80 15 80 15 Nr. 34



Text: 25 2B 80 2B 13 Nr. 35



Text: 24 14 9 65 63 Nr. 36



Text: 37 4A 57 21 51 Nr. 37



Text: 1 1 4 1 1 Nr. 38



Text: 1 1D 23 42 1 Nr. 39



Text: 1 42 23 1D 1 Nr. 40



Text: 15 9 3F 9 15 Nr. 41



Text: 9 9 3F 9 9 Nr. 42



Text: 1 41 31 1 1 Nr. 43



Text: 9 9 9 9 9 Nr. 44



Text: 1 61 61 1 1 Nr. 45



Text: 21 11 9 5 3 Nr. 46



Text: 3F 52 4A 46 3F Nr. 47



Text: 1 43 80 41 1 Nr. 48



Text: 63 52 4A 4A 47 Nr. 49



Text: 22 42 4A 4E 34 Nr. 50



Text: 19 15 13 80 11 Nr. 51



Text: 28 46 46 46 3A Nr. 52



Text: 3D 4B 4A 4A 31 Nr. 53



Text: 2 72 A 6 4 Nr. 54



Text: 37 4A 4A 4A 37 Nr. 55



Text: 7 4A 4A 2A 1F Nr. 56



Text: 1 1 15 1 1 Nr. 57



Text: 1 41 35 1 1 Nr. 58



Text: 9 15 23 42 1 Nr. 59



Text: 15 15 15 15 15 Nr. 60



Text: 1 42 23 15 9 Nr. 61



Text: 3 2 52 A 7 Nr. 62



Text: 3F 42 5E 56 1F Nr. 63



Text: 7F 12 12 12 7F Nr. 64



Text: 80 4A 4A 4A 37 Nr. 65



Text: 3F 42 42 42 23 Nr. 66



Text: 42 80 42 42 3F Nr. 67



Text: 80 4A 4A 4A 42 Nr. 68



Text: 80 A A A 2 Nr. 69



Text: 3F 42 42 52 73 Nr. 70



Text: 80 9 9 9 80 Nr. 71



Text: 1 42 80 42 1 Nr. 72



Text: 21 41 41 41 40 Nr. 73



Text: 80 9 15 23 42 Nr. 74



Text: 80 41 41 41 41 Nr. 75



Text: 80 3 D 3 80 Nr. 76



Text: 80 5 9 11 80 Nr. 77



Text: 3F 42 42 42 3F Nr. 78



Text: 80 A A A 7 Nr. 79



Text: 3F 42 52 22 5F Nr. 80



Text: 80 A 1A 2A 47 Nr. 81



Text: 27 4A 4A 4A 33 Nr. 82



Text: 2 2 80 2 2 Nr. 83



Text: 40 41 41 41 40 Nr. 84



Text: 8 19 61 19 8 Nr. 85



Text: 80 21 19 21 80 Nr. 86



Text: 64 15 9 15 64 Nr. 87

Text: 4 5 79 5 4 Nr. 88

Text: 62 52 4A 46 44 Nr. 89

Text: 1 80 42 42 1 Nr. 90

Text: 3 5 9 11 21 Nr. 91

Text: 1 42 42 80 1 Nr. 92

Text: 5 3 80 3 5 Nr. 93

Text: 41 41 41 41 41 Nr. 94

Text: 1 2 8 2 1 Nr. 95

Text: 21 55 55 55 79 Nr. 96

Text: 80 49 45 45 39 Nr. 97

Text: 39 45 45 45 21 Nr. 98

Text: 39 45 45 49 80 Nr. 99

Text: 39 55 55 55 9 Nr. 100

Text: 9 7F A 3 1 Nr. 101

Text: 9 15 55 55 3D Nr. 102

Text: 80 9 5 5 79 Nr. 103

Text: 1 45 7E 45 1 Nr. 104

Text: 21 41 41 3E 1 Nr. 105

Text: 80 11 29 45 1 Nr. 106

Text: 1 42 80 41 1 Nr. 107

Text: 79 5 19 5 79 Nr. 108

Text: 7D 9 5 5 79 Nr. 109

Text: 39 45 45 45 39 Nr. 110

Text: 7D 15 25 25 19 Nr. 111

Text: 19 25 15 7D 41 Nr. 112

Text: 7D 9 5 5 9 Nr. 113

Text: 49 55 55 55 21 Nr. 114

Text: 5 3F 45 21 1 Nr. 115

Text: 3D 41 41 21 7D Nr. 116

Text: 1D 21 41 21 1D Nr. 117

Text: 3D 41 31 41 3D Nr. 118

Text: 45 29 11 29 45 Nr. 119

Text: 5 49 31 9 5 Nr. 120

Text: 45 65 55 4D 45 Nr. 121

Text: 9 79 9 79 5 Nr. 122

Text: 1 1 80 1 1 Nr. 123

Text: 9 9 2B 1D 9 Nr. 124

Text: 64 56 4A 42 64 Nr. 125

Text: 80 9 9 9 1 Nr. 126

0.00000000 Nr. 127

Programmsammlung Power Pack

HP-28 C: POWER-PACK

Im folgenden habe ich Programme aus verschiedenen Bereichen für den HP-28 zusammengestellt. Alle Programme setzen voraus, daß der Anwender eine dem Rechner adäquate Intelligenz besitzt, und verzichten deshalb auf die einfachen Legalitätsprüfungen bei Eingaben.

Zu den Funktionen des HP-28C gehören auch die des PLOT-Menüs zur graphischen Darstellung von Gleichungen. Diese Abbildungen sind normalerweise allerdings zweifach beschränkt: Die Funktionen erscheinen in einem Ausschnitt, dessen Breite die Höhe etwa um das Vierfache übertrifft, und weiterhin können höchstens zwei Funktionen zugleich dargestellt werden.

Die Höhe des Displays läßt sich natürlich nicht verändern, jedoch kann der Drucker Bilder aus der Anzeige nahtlos zusammenfügen. Das Plotten von mehr als zwei Funktionen übereinander ist sogar im Rechnerdisplay möglich, da der Befehl DRAW im Programm **nicht** jedesmal das Display löscht, sondern nur überschreibt. Diese Anwendung ist allerdings nicht so spektakulär, da mehrere Funktionsplots schnell zu einem Chaos im Display werden. Soll eine Funktion jedoch mit zwei Asymptoten ausgestattet werden, so ist das jetzt möglich.

Die Programme sollten so einfach wie möglich zu bedienen sein. Daher greifen alle auf PPAR zurück, und speichern diese nach Programmende zurück.

MPL ist das Einzige der Plotprogramme, welches ohne Drucker benutzbar ist. Es stellt auf dem Display mehrere Funktionen zugleich dar. Dazu werden erst die Funktionen in den Stack gespeichert, wobei alles erlaubt ist, was auch DRAW akzeptiert. Dann muß die Anzahl der durch die Funktionen belegten Stackebenen (also **nicht** die Anzahl der Funktionen) eingegeben werden. Diese Informationen bleiben im Stack erhalten.

NPLOT druckt und vergrößert die Abbildungen von MPL. Zunächst empfiehlt es sich, die Funktionen mit MPL im Rechnerdisplay zu plotten, und zu prüfen, ob die Funktionen im richtigen Ausschnitt abgebildet werden. Für NPLOT müssen die gleichen Eingaben wie bei MPL gemacht werden. Zusätzlich muß dann in Ebene 1 noch ein Vergrößerungsfaktor (n , ganzzahlig und positiv) eingegeben werden. Dieser Faktor bewirkt, daß alles was von MPL auf einem LCD-Display abgebildet wird, von NPLOT auf n LCD's verteilt und ausgedruckt wird.

PLOT vergrößert ebenso wie NPLOT, verwendet aber nur die unter EQ gespeicherte Gleichung, ist damit also für DRAW das, was NPLOT für MPL ist.

PLOT braucht wie NPLOT einen Vergrößerungsfaktor in Ebene 1, sonst aber keine Eingaben.

Bei NPLOT und PLOT kann es vorkommen, daß die x-Achse auf dem Drucker in doppelter Breite gedruckt wird, wenn sie auf die Unterkante des LCD-Displays fällt und im nächsten Display noch einmal erscheint. Die etwas aufwendige Schleife bei diesen Programmen stellt sicher, daß beim oberen und unteren Ende des Funktionsplots auf dem Drucker die Original-Werte für PMAX und PMIN eingesetzt werden, damit hinterher nicht etwa die X-Achse fehlt.

CM berechnet Binomialkoeffizienten, die z.B. für das Pascalsche Dreieck oder Kombinationen benötigt werden. Als Definition der Binomialkoeffizienten wird oft die rechte Definition angegeben. Sie schränkt n jedoch auf positive Ganzzahlen ein. Die eigentliche Definition (in der Mitte) läßt für n jedoch reelle Werte zu (s. Beispiele). Für k dürfen dagegen nur positive Ganzzahlen eingesetzt werden; mit $k = 0$ ist der Wert des Binomialkoeffizienten nach Definition 1.

Die Programme PRFL und KR enthalten einen IFERR-Test. Falls in dieser Struktur tatsächlich ein Fehler auftritt, ist der Stackinhalt davon abhängig, ob LAST aktiviert war. Sollte dies nicht der Fall sein, können die Programme versagen.

Anmerkung der Redaktion

Mit dem Befehl Comb aus dem Statistikenmenü des HP28S lassen sich für positive Integer die Binomialkoeffizienten $\binom{U}{K}$ ermitteln.

PRFL entspricht der HP-41 Funktion PRFLAGS. PRFL benötigt keine Eingaben, stellt aber auf dem Drucker den Zustand aller Flags durch 0 oder 1 (= gesetzt) dar. Das größte Problem ist, dieses Programm in den Speicher zu bekommen. Um den infinitesimal kleinen Speicher des HP-28C nicht allzu sehr zu belasten, habe ich die Steuercodes für CR, „Fettdruck ein“ und „Fettdruck aus“ gleich zusammen mit Text im ersten String untergebracht. Zunächst muß das Programm **TEXT** eingegeben werden. Es soll aber nicht unter der Variablen **TEXT** abgespeichert werden, sondern sofort mit EVAL ausgeführt werden. Das Ergebnis ist der Textstring, der auch im Programm auftaucht. Nun wird EDIT ausgeführt und vor dem String das Programmbegrenzungszeichen (<<) gesetzt, und dahinter die im Listing von PRFL aufgeführten Befehle.

KR berechnet die Krümmung einer Kurve. R (also $1/K$) ist der Durchmesser des Krümmungskreises, der eine Funktion y an der Stelle $(x|y(x))$ berührt. Es gibt

den Fall $K = 0$ z.B. bei Geraden, die ja nirgends gekrümmt sind, außerdem an Wende- und Sattelpunkten von Funktionen. Umgekehrt kommt $R = 0$ nicht vor, so daß die Berechnung von K durchaus sinnvoll ist um „Infinite Result“ zu vermeiden. Die Eingaben bei KR entsprechen denen der Standardfunktion „d/dx“: Funktion in Ebene 2, unabhängige Variable in Ebene 1. Das Programm löscht – sofern vorhanden – die unabhängige Variable in Ebene 1, und speichert ihren Wert im Stack zwischen, da andernfalls ihr Wert sofort in die 1. Ableitung eingesetzt und ausgerechnet wird; so könnte aus ihr keine 2. Ableitung gebildet werden. KR gibt die Funktion der Krümmung $K(x)$ aus. War die unabhängige Variable vor Programmstart bereits im USER-Menue, so wird sie zurückgespeichert, und kann mit EVAL eingesetzt werden. Man kann auch mit COLCT, EDIT die Funktion in ein Programm verwandeln und abspeichern.

Die folgenden drei Programme dienen der Datumsberechnung. Der Tag wird dazu in Ebene 3 eingegeben (für FR nicht erforderlich), der Monat in Ebene 2 und das Jahr (z.B. 1988, nicht 88) in Ebene 1. Wer als Dezimaltrennzeichen das Komma verwendet (also „RDX,“ eingestellt hat), kann das Datum in der gewohnten Form – also durch Punkte getrennt – eingeben. Der HP-28 verteilt dann die Eingabe richtig auf die Stackebenen.

JDN berechnet das Julianische Datum, eine fortlaufende Tageszählung. Mit ihr lassen sich einfach Summen und Differenzen von Tagen bilden. Die Formeln stehen im PPC-ROM-USER-Manual unter CJ.

DOW berechnet den Wochentag und schreibt den Namen in Ebene 1.

FR nach Eingabe von Monat und Jahr gibt FR alle Freitage aus, die auf einen 13. fallen. FR läuft in einer Endlosschleife. Diese wurde mit DO ... UNTIL 0 END erzeugt, da zwischen UNTIL und END immer eine Testanweisung steht, die eine 0 oder 1 erzeugt. Bei 1 wird nach END, bei 0 nach DO fortgefahren.

Die Programme GL2 und GL3 lösen Gleichungen 2. und 3. Grades (s. Beispiele). Die verwendeten Formeln wurden zusammen mit meinem HP-15 Programm in PRISMA 85.2.11 abgedruckt. GL4 konnte ich wegen Platzmangel im HP-28 bisher nicht entwickeln.

GL2 löst quadratische Gleichungen. Dazu muß p in Ebene 2 und q in Ebene 1 eingegeben werden. Es können auch quadratische Gleichungen mit komplexen Koeffizienten gelöst werden (s. Beispiel).

Kubische Gleichung

$$x^3 + ax^2 + bx + c = 0$$

Beispiele:

$$x^3 - 2x^2 + 5x + 26 = 0$$

$$a = -2; b = 5; c = 26;$$

$$L: -2; 2 + 3i; 2 - 3i;$$

$$x^3 - 21x - 20 = 0$$

$$a = 0; b = -21; c = -20;$$

$$L: 5; -1; -4;$$

Quadratische Gleichung

$$x^2 + px + q = 0$$

Beispiel:

$$x^2 - (1-2i)x - 6 + 4i = 0$$

$$p = -1 + 2i; q = -6 + 4i;$$

$$L: -2; 3 - 2i;$$

$$K = \frac{1}{R} = \frac{y''(x)}{(1+y'^2(x))^{3/2}}$$

Krümmung von y(x) an der Stelle x.

$$S = \int_a^b \sqrt{1 + y'^2(x)} dx$$

Bogenlänge S der Funktion y(x) im Intervall [a|b].

$$\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k!} = \frac{n!}{(n-k)!k!}$$

Binomialkoeffizient (Definition).

```
KR
< 1 CF DUP IFERR RCL
THEN DROP SWAP ELSE 1 SF
OVER PURGE SWAP ROT END
OVER & DUP2 SWAP & SWAP
SQ 1 + 1,5 ^ / IF 1 FS?C
THEN ROT ROT STO ELSE
SWAP DROP END >
```

```
2: 'COSH(X)'
1: 'X'
```

```
KR
COSH(X)/(SQ(SINH(X)
)+1)^1,5'
```

```
3 'X' STO EVAL
9,86603716546E-3
```

```
S
< OVER 1 GET DUP PURGE 4
ROLL SWAP & SQ 1 + J ROT
ROT J >
```

```
3: 'COSH(X)'
2: ( X 2,5 )
1: ,00001
```

```
S
2: 70,5763498534
1: 7,05650180238E-4
```

```
CM
< SWAP IF OVER 0 ≠ THEN
> D & FACT 1 0 LAST 1 -
FOR I N 1 - * NEXT SWAP
/ > ELSE DROP2 1 END >
```

```
49 6 CM
13983816
```

```
-3,5 4 CM
23,4609375
```

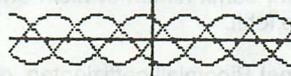
```
'X' 3 CM
'X*(X-1)*(X-2)/6'
```

```
MPL
< CLLCD 2 OVER 1 + FOR I
I PICK STEQ DRAW NEXT >
```

```
4: 'SIN(X-4*PI/3)'
3: < X SIN >
2: 'SIN(X+4*PI/3)'
1: 3
```

```
PPAR
{ (-6,8,-1,5) (6,8,1,6)
X 1 (0,0) }
```

MPL PRLCD

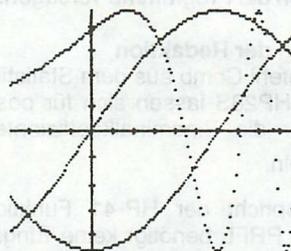


```
NPLOT
< PPAR DUP CR LIST > 4
DROPN - IM 0 SWAP R > C >
f n p d < 1 n FOR I P
LIST > 4 DROPN I 1 - n /
d * + PMAX I n / 1 - d *
+ PMIN CLLCD 1 f FOR J J
PICK STEQ DRAW NEXT
PRLCD NEXT f P > 'PPAR'
STO CLMF >
```

```
PPAR
{ (-1,-1) (2,5.1) X 1
(0,0) }
```

```
4: < X EXP SIN >
3: 'SIN(X)=-COS(X)'
```

NPLOT

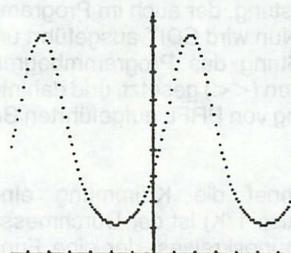


```
PLOT
< PPAR CR LIST > 4 DROPN
DUP2 - IM 0 SWAP R > C 4
ROLL > d n < 1 n FOR I
DUP2 1 1 - n / d * +
PMAX I n / 1 - d * +
PMIN CLLCD DRAW PRLCD
NEXT > PMAX PMIN CLMF >
```

```
EQ
< X SIN EXP >
```

```
PPAR
{ (-6.0) (6.3) X 1 (0.0)
}
```

3 PLOT



```
GL2
< SWAP -2 / DUP SQ ROT -
J + LAST - >
```

```
GL3
< 2 / ROT ROT > a b < a
3 / DUP SQ b 2 / - * + b
3 * a SQ - 9 / DUP2 3 ^
SWAP SQ + IF DUP 0 >
THEN J ROT + LAST - SIGN
LAST ABS 3 INV * SWAP
SIGN LAST ABS 3 INV ^ *
- SWAP DROP ELSE DROP
DUP ABS J DUP ROT * ROT
SWAP / ACOS 3 / COS *
DUP + END a 3 / - LAST 2
* + DUP2 * b + > GL2 >
```

```
TEXT
< 10 CHR 27 CHR + 253
CHR + "HP-28 FLAGS" +
27 CHR + 252 CHR + 10
CHR DUP + +
"NO. 0 1 2 3 4 5 6 7 8
9" >
```

PRL

HP-28 FLAGS

```
No. 0 1 2 3 4 5 6 7 8 9
PR1 DROP 0 6 FOR a a
>STR "x " + 0 9 FOR b
" " + a 10 * b + IFERR
FS? THEN DROP " END
>STR + NEXT PR1 DROP
NEXT CR >
```

PRL

HP-28 FLAGS

```
No. 0 1 2 3 4 5 6 7 8 9
0x 0 0 0 0 0 0 0 0 0 0
1x 0 0 0 0 0 0 0 0 0 0
2x 0 0 0 0 0 0 0 0 0 0
3x 0 1 0 0 0 0 0 1 1 1
4x 1 1 1 1 1 1 1 0 0 1
5x 0 1 0 0 0 0 0 0 0 1
6x 1 0 1 1 0
```

JDN

```
< SWAP 2,85 - 12 / + DUP
+ y < 367 * IP + y IP
1,75 * - IP y 100 / IP
,75 * - IP 1721115 + >
```

DOW

```
< JDN 7 MOD 1 + ( Mo Di
Mi Do Fr Sa So ) SWAP
GET >
```

FR

```
< 0 > m j a < DO 2 m j
JDN 7 MOD IF 0 = THEN
"Fr. 13. n n IF 9 ≤ THEN
"0" + END " >STR + " " +
j >STR + a 1 + DUP 4
MOD 'a' STO DISP END m 1
+ IF DUP 12 > THEN DROP
1 m, STO 1 j + 'j' STO
ELSE 'm' STO END UNTIL 0
END >
```

```
24.2.1963 DOW
'So'
```

```
8.8.1988 JDN
2447382
7.7.1977 JDN
2443332
```

4050

(Ergebnis wie bei der HP-41 Funktion DDAYS)

```
1.1988 FR
FR. 13.05.1988
FR. 13.01.1989
FR. 13.10.1989
FR. 13.04.1990
FR. 13.02.1990
FR. 13.09.1991
```

STRINGLX

856 Bytes, HP-71

... UND NOCH EIN LEXFILE

Bei diesem Lexfile handelt es sich um eine sehr leistungsfähige Systemerweiterung des HP-71B was das Bearbeiten von Strings betrifft.

Da wird z.B. das Suchen nach bestimmten Zeichen im String verbessert, Zeichen am Anfang oder Ende eines Strings können abgeschnitten werden, die Umwandlung in Kleinbuchstaben wird ermöglicht, Strings können rotiert, gedreht und vervielfältigt werden und einzelne Bits in einem String von Bytes können bearbeitet werden.

STRINGLX stammt von HP direkt und ist Public Domain. Es wurde unter anderem im „Software Developers Handbook“ von HP beschrieben und ist auch in der Users Library erhältlich. Der File enthält 11 Schlüsselwörter. Die Übersicht nach LEXCHECK2 sieht so aus:

STRINGLX ID=52	SIZE=856 Bytes	Polis!
08	LTRIM\$	XFN 82008 (\$[,,\$])
09	RTRIM\$	XFN 82009 (\$[,,\$])
0A	TRIM\$	XFN 82010 (\$[,,\$])
0B	RPT\$	XFN 82011 (\$,#)
0C	REV\$	XFN 82012 (\$)
0D	LWRC\$	XFN 82013 (\$)
0E	LWC\$	XFN 82014 (\$)
0F	ROT\$	XFN 82015 (\$,#)
10	SPAN	XFN 82016 (\$,\$[,,\$])
11	MEMBER	XFN 82017 (\$,\$[,,\$])
12	SBIT\$	XFN 82018 (\$,#[,,\$][,,\$])
13	SBIT	XFN 82019 (\$,#,#)

Die Funktion LTRIM\$ wird dazu benutzt, um von einem gegebenen String etwaige überflüssige Leerzeichen am Anfang

GL3 löst kubische Gleichungen, wenn a in Ebene 3, b in 2 und c in Ebene 1 eingegeben wurden. Mit den angegebenen Beispielen wird zugleich auch GL2 ausgetestet. Die Lösung in Ebene 3 ist immer reell.

S berechnet die Bogenlänge einer Funktion. Die Bogenlänge ist der Weg, den man zurücklegt, wenn man auf dem Graph der Funktion von einem Punkt zu einem anderen entlangwandert, also nicht den kürzesten Weg wählt. Die Eingaben müssen wie bei der Integrationsfunktion (f) gemacht werden, mit der Einschränkung, daß die Funktion in Ebene 3 kein Programm sein darf, also ein algebraischer Ausdruck sein muß.

Ralf Pfeifer (116)
Rubensstraße 5
5000 Köln 50

abzuschneiden. Die Funktion RTRIM\$ bewirkt dasselbe am Ende eines Strings. TRIM\$ schließlich „beschneidet“ beide Enden des Strings.

Dies kann z.B. nützlich sein, um den Filenamen eines Katalogeintrages zu isolieren. Angenommen, Sie haben einen File namens TEST als 2. File im Speicher, so erhält man mit

N\$=CAT\$(2) [1,8]

zwar in N\$ den Filenamen TEST, allerdings mit 4 Leerzeichen auf die Länge 8 aufgefüllt.

N\$=RTRIM\$(N\$)

bringt dann das gewünschte Ergebnis mit einer Stringlänge von 4, mit dem man dann weiterarbeiten kann.

Wird in den genannten Funktionen ein zweiter Stringparameter angegeben, so wird nicht um Leerzeichen gekürzt, sondern um den angegebenen Buchstaben. Einige Beispiele:

LTRIM\$(“ abc def “) =“abc def “
RTRIM\$(“AB UND ZU“) =“AB UND ZU“
TRIM\$(“** 1 ** 2 **“,““) =“ 1 ** 2“

Die Funktion RPT\$ ergibt einen String, der sich selbst wiederholt um den angegebenen Faktor:

PRT\$(“WAU“,4)=“WAUWAUWAUWAU“
RPT\$(“-“,80) =(ergibt eine ganze Zeile von Minuszeichen)

Natürlich muß darauf geachtet werden, daß eine etwaige Stringvariable vorher ausreichend lang dimensioniert ist!

Die Funktion REV\$ ist bereits eine alte Bekannte. Sie kommt auch in einigen anderen Lexfiles vor (DISLEX, DESAL) und wurde auch schon einmal alleine vorgestellt (REVLEX). Der Ausgangstring wird einfach umgedreht:

REV\$(“TIER“)=“REIT“

oder, wer kennt das noch nicht:

REV\$(“EIN NEGER MIT GAZELLE ZAGT IM REGEN NIE“)?

LWC\$ und LWRC\$ sind beide gleich (das eine ist kürzer, das andere ist analog zu UPRC\$ gebildet) und wandeln den ganzen String in Kleinbuchstaben um, soweit möglich. .

LWC\$(“FRAGE: 6 Plus 2 ?“)
=“frage: 6 plus 2 ?“

ROT\$ wird dazu benutzt, um einen String um eine bestimmte Anzahl von Stellen zu drehen. Ein positives Argument dreht nach rechts, ein negatives nach links. Die am Ende des Strings hinausgeschobenen Zeichen werden am anderen Ende wieder angefügt. Z.B.:

ROT(“DAS LETZTE MAL “,4)
=“MAL DAS LETZTE“
ROT\$(“DAS LETZTE MAL “,-4)
=“LETZTE MAL DAS“

Die Funktionen SPAN und MEMBER sind Erweiterungsfunktionen zu POS. Mit ihnen kann die Position des ersten gefundenen Buchstabens in einem String ermittelt werden, der zu einer definierten Menge gehört (MEMBER) oder nicht gehört (SPAN). Die definierte Menge muß im Setstring (2. Parameter) angegeben werden. Z.B.:

MEMBER(“Heute ist der 2. Juni“,
“0123456789“)=15

da an 15. Stelle erstmals eine Ziffer vorkommt.

SPAN(“10011011“,“01“)=0

da kein Zeichen außer 0 und 1 vorkommt.

Wird der optionale 3. Parameter verwendet, so bezeichnet er die Startposition, ab der gesucht werden soll. Fehlt er, so wird ab Stringanfang gesucht.

Nun zu den Funktionen SBIT\$ und SBIT. Diese sind für den fortgeschrittenen Programmierer gedacht, wenn er Strings dazu benutzt, nicht nur einfache Buchstaben und Satzzeichen darin zu speichern, sondern Binärdaten. Schließlich kann ja jedes Zeichen eines Strings einen Wert zwischen 0 und 255 annehmen und damit genau ein Byte speichern. Diese Bytes lassen sich nun durch diese Funktionen Bit für Bit ansprechen, entweder um sie zu ändern (SBIT\$), oder sie auszulesen (SBIT).

Mit SBIT\$(str\$,n,b,v) wird im String str\$ das n-te Byte angesprochen und dessen b-tes Bit auf den Wert v gesetzt. v darf nur den Wert 0 oder 1 annehmen, b nur 0 bis 7. Wird kein v angegeben, so wird das entsprechende Bit einfach invertiert. Wird auch kein b angegeben, so werden alle 8 Bits invertiert. Beispiel:

NUM(SBIT\$(CHR\$(255),1,7,0))=127

da der String nur aus einem Byte besteht, das den Wert 255 hat. Bei diesem Wert sind alle Bits gesetzt. Wird nun das höchstwertige Bit (Nr. 7) gelöscht, so bedeutet dies in der Binärarithmetik eine Subtraktion um 128.

Achtung: SBIT\$ ist eine Stringfunktion und liefert als Ergebnis den geänderten String. Der ursprüngliche String wird dabei nicht verändert. Es sei denn man programmiert z.B.:

```
A$=SBIT$(A$, 2)
```

Da sich im ASCII-Zeichensatz des HP-71B die Groß- und Kleinbuchstaben nur durch das Bit Nr. 5 unterscheiden, könnte man z.B. als Ersatz für die LWC\$ Funktion programmieren:

```
10 FOR I=1 TO LEN (A$)
20 A$=SBIT$(A$,I,5,1)
30 NEXT I
```

Die Funktion SBIT liefert nur ein Bit als Ergebnis, also entweder 0 oder 1. SBIT (str\$,n,b) liefert den Wert des b-ten Bits im n-ten Byte des Strings str\$. Damit kann man also sofort die Funktion von SBIT\$ testen:

```
SBIT(SBIT$(CHR$(255),1,7,0),1,7)=0
```

weil wir es ja gerade auf 0 gesetzt hatten! Ein weiteres gutes Beispiel, wofür SBIT sinnvoll eingesetzt werden kann ist das Testen des Graphik-Displays auf gesetzte Pixels:

```
B=SBIT(GDISP$(1,0))
```

ergibt den Wert des Pixels oben links im Display.

Daß STRINGLX sehr nützlich sein kann, ist damit wohl hinreichend gezeigt. Die Möglichkeiten der Anwendung der einzelnen Funktionen sind schier unbegrenzt und voll der Phantasie des Programmierers überlassen. Viele gute Ideen dazu wünscht

Michael Fiedler
Friedrichstraße 17
6070 Langen

Hexdump-Listing STRINGLX

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

```
STRINGLX L ID#52 856 Bytes
```

```
0123 4567 89AB CDEF ck
000: 3545 2594 E474 C485 8D
001: 802E 0000 2120 4088 40
002: F860 0258 0310 0000 50
003: FA70 0000 0A01 0000 83
004: 0C31 00F7 5014 100F 5A
005: 9803 4100 FC40 3730 25
006: 0F63 0002 00FA 1050 F0
007: 200F F00C F100 F140 18
008: 9F30 0FE7 0A42 00F7 B5
009: 20D4 200F 6606 6400 C8
00A: F370 A850 0FBC 4452 B6
00B: 594D 4428 07C4 7534 C2
00C: 42E0 9C47 5253 442D 20
00D: 0BD4 54D4 2454 2511 ED
00E: 7255 4654 2C07 25F4 37
00F: 4542 F072 5054 542B 9C
```

```
010: 0B25 4525 94D4 4290 82
011: 9352 4944 5422 1735 9B
012: 2494 4531 7350 514E 73
013: 4019 4525 94D4 42A0 E7
014: 1FF9 6950 5E21 1B13 F3
015: 5112 1CB1 378B 6A11 7B
016: 3510 B3B1 4A32 5453 7D
017: 5021 5DB0 0441 2850 D3
018: 8416 8104 4128 406A 3B
019: 0044 1285 0851 A4E3 20
01A: 1029 4AC1 71A4 8A83 E4
01B: 1137 C213 71C1 14F1 F7
01C: 71D7 AF07 284A F813 19
01D: 3100 C010 1102 1318 DC
01E: 1D7D 7486 0021 C18A 60
01F: 9F01 4F96 770C D5DE EF
020: 1711 3710 A118 1358 E6
021: 6171 8A92 114F 967A 07
022: 0171 CD5D EC51 1111 2D
023: A8FE E0B1 119E 9135 98
024: 7014 AF0D 4BF0 BF02 1B
025: 0A0C 1CF1 5178 DC32 46
026: F041 18FE 83B1 6EEF 64
027: 4111 3710 8135 AF07 FD
028: AC38 1CBF 0BF0 BF03 79
029: 114D 531A 5D73 102A 3A
02A: EA1C 1171 A5C4 8114 DB
02B: F9E1 1F9E 7CEA 6214 C4
02C: D53E 1181 3560 9F84 36
02D: 4238 416B 0084 4238 0A
02E: 5180 DF89 2327 6434 94
02F: 117D 4349 194D 0151 5A
030: 131B 08D3 9390 DOE4 F7
031: 2010 07A4 3AF0 7D23 BE
032: 81C1 0113 710A C2C2 51
033: 1357 613A F213 7C21 0F
034: 0B81 CD6D 7110 11B1 EE
035: 35CC 4F0C C4F1 1C1C 69
036: F55F 11B1 3566 4087 8B
037: 13F1 C15A 311A 1341 5C
038: 19D5 CF4D DCD4 2E1C 7F
039: 114F 14A1 6196 6A08 2A
03A: 617D 5C0C D5AE 871A 96
03B: C7F9 2113 1371 31EA 36
03C: 81C7 5726 C8E8 4227 80
03D: 2624 9079 6246 0632 A9
03E: F101 AF07 2628 AC90 DF
03F: 1CF6 76E1 0213 7135 0C
040: C210 810B D711 98F9 28
041: 43B1 DBE2 4927 4421 B9
042: 12D8 1101 311B 495F 59
043: 2142 1301 6F13 28BA A0
044: 908D D449 0111 CC10 02
045: 1421 11B1 338F EE0B 14
046: 157E 1131 3713 5EAD 52
047: 86EC D842 276E 174B 40
048: 1411 7BB1 8514 0183 10
049: 1606 D6E8 41F8 C444 2B
```

Fortsetzung auf Seite 48

Korrektur zu DIVISION ohne Grenzen

Heft 2/88

Folgende Fehler haben sich in den Artikel eingeschlichen:

Auf Seite 24, 1. Spalte unten muß es 0,1,2,3,...,22 heißen.
Auf Seite 24, 2. Spalte muß es oben heißen: „Damit wird die Verarbeitungszeit bei gegebener Zifferzahl verkürzt“

Das 2. Beispiel auf Seite 25, 1. Spalte bitte durch das folgende ersetzen:

Beispiel 2. 33785 : 345

```
3450033785 Feld A
3450933785 -
- 27 -
3450906785 -- Der Rest ist korrekt --
- 36 -
3450903185 -
- 45 -
3450902735 Der „reduzierte“ Dividend ist jetzt 2735
```

Ganz unten ist bei der vorletzten 120080120 eine 0 zuviel drin.

In der rechten Spalte der Seite muß es unter Beispiel 4. 1110011022 heißen, dort ist eine 1 zuviel gewesen.

Im 1. Strukturprogramm zum Hauptprogramm habe ich aus Versehen zweimal das Wort Ziffer statt der gemeinten Zuffer verwendet.

Zu guter Letzt noch ein Formelfehler im Unterprogramm U1: hier steht in U4 v:=+1; es muß heißen v:=v-1.

Werner Peppel
An der Halde 3
7325 Boll

↑ (Roger Heizig) 1607

Computerclub Deutschland e.V.

Zeile 1 (1- 6) CCD-Barcodes



Zeile 2 (6- 13) CCD-Barcodes



Zeile 3 (13- 19) CCD-Barcodes



Zeile 4 (19- 31) CCD-Barcodes



Zeile 5 (31- 43) CCD-Barcodes



Zeile 6 (43- 53) CCD-Barcodes



Zeile 7 (53- 62) CCD-Barcodes



Zeile 8 (62- 73) CCD-Barcodes



Zeile 9 (73- 80) CCD-Barcodes



Zeile 10 (80- 89) CCD-Barcodes



Zeile 11 (89- 97) CCD-Barcodes



Zeile 12 (97- 106) CCD-Barcodes



Zeile 13 (106- 116) CCD-Barcodes



Zeile 14 (116- 127) CCD-Barcodes



Zeile 15 (127- 140) CCD-Barcodes



Zeile 16 (140- 150) CCD-Barcodes



Zeile 17 (150- 158) CCD-Barcodes



Zeile 18 (158- 170) CCD-Barcodes



Zeile 19 (170- 178) CCD-Barcodes



Zeile 20 (178- 183) CCD-Barcodes



Zeile 21 (183- 191) CCD-Barcodes



Zeile 22 (191- 197) CCD-Barcodes



EREI (Ulrich Laag) 1608

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes



Zeile 2 (3- 11) CCD-Barcodes



Zeile 3 (11- 19) CCD-Barcodes



Zeile 4 (19- 28) CCD-Barcodes



Zeile 5 (28- 37) CCD-Barcodes



Zeile 6 (37- 45) CCD-Barcodes



Zeile 7 (45- 53) CCD-Barcodes



Zeile 8 (53- 61) CCD-Barcodes



Zeile 9 (61- 70) CCD-Barcodes



Zeile 10 (70- 70) CCD-Barcodes



C-E24 (Ulrich Laag) 1609

Computerclub Deutschland e.V.

Zeile 1 (1- 2) CCD-Barcodes



Zeile 2 (2- 10) CCD-Barcodes



Zeile 3 (10- 16) CCD-Barcodes



Zeile 4 (16- 22) CCD-Barcodes



Zeile 5 (22- 28) CCD-Barcodes



Zeile 6 (28- 33) CCD-Barcodes



Zeile 7 (33- 39) CCD-Barcodes



Zeile 8 (39- 44) CCD-Barcodes



Zeile 9 (44- 49) CCD-Barcodes



Zeile 10 (49- 54) CCD-Barcodes



Zeile 11 (54- 60) CCD-Barcodes



Zeile 12 (60- 62) CCD-Barcodes



SPT (Ulrich Laag) 1610

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes



Zeile 2 (3- 7) CCD-Barcodes



Zeile 3 (7- 16) CCD-Barcodes



Zeile 4 (16- 21) CCD-Barcodes



Zeile 5 (21- 34) CCD-Barcodes



Zeile 6 (34- 40) CCD-Barcodes



Zeile 7 (40- 47) CCD-Barcodes



BARCODES

Zeile 8 (47- 57) CCD-Barcodes



Zeile 9 (57- 62) CCD-Barcodes



Zeile 10 (62- 71) CCD-Barcodes



Zeile 11 (71- 76) CCD-Barcodes



Zeile 12 (76- 82) CCD-Barcodes



Zeile 13 (82- 87) CCD-Barcodes



Zeile 14 (87- 95) CCD-Barcodes



Zeile 15 (95- 99) CCD-Barcodes



Zeile 20 (71- 73) CCD-Barcodes



Zeile 21 (73- 77) CCD-Barcodes



Zeile 22 (77- 84) CCD-Barcodes



Zeile 23 (84- 88) CCD-Barcodes



Zeile 24 (88- 91) CCD-Barcodes



Zeile 25 (91- 93) CCD-Barcodes



Zeile 26 (93- 97) CCD-Barcodes



Zeile 27 (97- 106) CCD-Barcodes



Zeile 28 (106- 115) CCD-Barcodes



Zeile 29 (115- 117) CCD-Barcodes



Zeile 30 (117- 121) CCD-Barcodes



Zeile 31 (121- 126) CCD-Barcodes



Zeile 32 (126- 128) CCD-Barcodes



Zeile 33 (128- 129) CCD-Barcodes



Zeile 34 (129- 136) CCD-Barcodes



Zeile 35 (136- 141) CCD-Barcodes



Zeile 36 (141- 142) CCD-Barcodes



Zeile 37 (142- 145) CCD-Barcodes



Zeile 38 (145- 147) CCD-Barcodes



Zeile 39 (147- 149) CCD-Barcodes



Zeile 40 (149- 157) CCD-Barcodes



Zeile 41 (157- 166) CCD-Barcodes



Zeile 42 (166- 173) CCD-Barcodes



Zeile 43 (173- 173) CCD-Barcodes



Zeile 44 (173- 179) CCD-Barcodes



Zeile 45 (179- 187) CCD-Barcodes



Zeile 46 (187- 197) CCD-Barcodes



Zeile 47 (197- 198) CCD-Barcodes



Zeile 48 (198- 201) CCD-Barcodes



ZVE (August Schorr) 1611

Computerclub Deutschland e.V.

Zeile 1 (1- 5) CCD-Barcodes



Zeile 2 (5- 7) CCD-Barcodes



Zeile 3 (7- 8) CCD-Barcodes



Zeile 4 (8- 11) CCD-Barcodes



Zeile 5 (11- 12) CCD-Barcodes



Zeile 6 (12- 17) CCD-Barcodes



Zeile 7 (17- 19) CCD-Barcodes



Zeile 8 (19- 23) CCD-Barcodes



Zeile 9 (23- 24) CCD-Barcodes



Zeile 10 (24- 27) CCD-Barcodes



Zeile 11 (27- 33) CCD-Barcodes



Zeile 12 (33- 33) CCD-Barcodes



Zeile 13 (33- 38) CCD-Barcodes



Zeile 14 (38- 46) CCD-Barcodes



Zeile 15 (46- 53) CCD-Barcodes



Zeile 16 (53- 56) CCD-Barcodes



Zeile 17 (56- 59) CCD-Barcodes



Zeile 18 (59- 68) CCD-Barcodes



Zeile 19 (68- 71) CCD-Barcodes



Zeile 49 (201- 203) CCD-Barcodes



Zeile 50 (203- 211) CCD-Barcodes



Zeile 51 (211- 214) CCD-Barcodes



Zeile 52 (214- 218) CCD-Barcodes



Zeile 53 (218- 220) CCD-Barcodes



Zeile 54 (220- 226) CCD-Barcodes



Zeile 55 (226- 233) CCD-Barcodes



Zeile 56 (233- 238) CCD-Barcodes



Zeile 57 (238- 245) CCD-Barcodes



Zeile 58 (245- 255) CCD-Barcodes



Zeile 59 (255- 264) CCD-Barcodes



Zeile 60 (264- 271) CCD-Barcodes



Zeile 61 (271- 280) CCD-Barcodes



Zeile 62 (280- 287) CCD-Barcodes



Zeile 63 (287- 294) CCD-Barcodes



Zeile 64 (294- 304) CCD-Barcodes



Zeile 65 (304- 309) CCD-Barcodes



Zeile 66 (309- 317) CCD-Barcodes



Zeile 67 (317- 326) CCD-Barcodes



Zeile 68 (326- 334) CCD-Barcodes



Zeile 69 (334- 344) CCD-Barcodes



Zeile 70 (344- 344) CCD-Barcodes



Zeile 71 (344- 347) CCD-Barcodes



Zeile 72 (347- 352) CCD-Barcodes



Zeile 73 (352- 358) CCD-Barcodes



Zeile 74 (358- 360) CCD-Barcodes



Zeile 75 (360- 366) CCD-Barcodes



Zeile 76 (366- 372) CCD-Barcodes



Zeile 77 (372- 374) CCD-Barcodes



Zeile 78 (374- 380) CCD-Barcodes



Zeile 79 (380- 381) CCD-Barcodes



Zeile 80 (381- 385) CCD-Barcodes



Zeile 81 (385- 389) CCD-Barcodes



Zeile 82 (389- 390) CCD-Barcodes



Zeile 83 (390- 394) CCD-Barcodes



Zeile 84 (394- 400) CCD-Barcodes



Zeile 85 (400- 409) CCD-Barcodes



Zeile 86 (409- 417) CCD-Barcodes



Zeile 87 (417- 426) CCD-Barcodes



Zeile 88 (426- 429) CCD-Barcodes



Zeile 89 (429- 433) CCD-Barcodes



PBZ (Michael Schilli) 1612

Computerclub Deutschland e.V.

Zeile 1 (1- 5) CCD-Barcodes



Zeile 2 (5- 14) CCD-Barcodes



Zeile 3 (14- 23) CCD-Barcodes



Zeile 4 (23- 25) CCD-Barcodes



Zeile 5 (25- 35) CCD-Barcodes



Zeile 6 (35- 40) CCD-Barcodes



Zeile 7 (40- 47) CCD-Barcodes



Zeile 8 (47- 54) CCD-Barcodes



Zeile 9 (54- 59) CCD-Barcodes



Zeile 10 (59- 65) CCD-Barcodes



Zeile 11 (65- 76) CCD-Barcodes



Zeile 12 (76- 77) CCD-Barcodes



Zeile 13 (77- 85) CCD-Barcodes



Zeile 14 (85- 95) CCD-Barcodes



Zeile 15 (95- 99) CCD-Barcodes



BARCODES

Zeile 16 (99- 107) CCD-Barcodes



Zeile 17 (107- 109) CCD-Barcodes



Zeile 18 (109- 118) CCD-Barcodes



Zeile 19 (118- 128) CCD-Barcodes



Zeile 20 (128- 137) CCD-Barcodes



Zeile 21 (137- 146) CCD-Barcodes



Zeile 22 (146- 156) CCD-Barcodes



Zeile 23 (156- 166) CCD-Barcodes



Zeile 24 (166- 176) CCD-Barcodes



Zeile 25 (176- 185) CCD-Barcodes



Zeile 26 (185- 191) CCD-Barcodes



Zeile 27 (191- 200) CCD-Barcodes



Zeile 28 (200- 212) CCD-Barcodes



Zeile 29 (212- 221) CCD-Barcodes



Zeile 30 (221- 232) CCD-Barcodes



Zeile 31 (232- 240) CCD-Barcodes



Zeile 32 (240- 249) CCD-Barcodes



Zeile 33 (249- 260) CCD-Barcodes



Zeile 34 (260- 268) CCD-Barcodes



Zeile 35 (268- 277) CCD-Barcodes



Zeile 36 (277- 288) CCD-Barcodes



Zeile 37 (288- 296) CCD-Barcodes



Zeile 38 (296- 306) CCD-Barcodes



Zeile 39 (306- 316) CCD-Barcodes



Zeile 40 (316- 326) CCD-Barcodes



Zeile 41 (326- 337) CCD-Barcodes



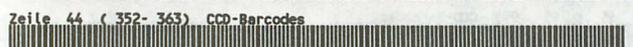
Zeile 42 (337- 344) CCD-Barcodes



Zeile 43 (344- 352) CCD-Barcodes



Zeile 44 (352- 363) CCD-Barcodes



Zeile 45 (363- 371) CCD-Barcodes



Zeile 46 (371- 380) CCD-Barcodes



Zeile 47 (380- 390) CCD-Barcodes



Zeile 48 (390- 399) CCD-Barcodes



Zeile 49 (399- 407) CCD-Barcodes



Zeile 50 (407- 417) CCD-Barcodes



Zeile 51 (417- 426) CCD-Barcodes



Zeile 52 (426- 433) CCD-Barcodes



Zeile 53 (433- 444) CCD-Barcodes



Zeile 54 (444- 454) CCD-Barcodes



Zeile 55 (454- 464) CCD-Barcodes



Zeile 56 (464- 472) CCD-Barcodes



Zeile 57 (472- 483) CCD-Barcodes



Zeile 58 (483- 493) CCD-Barcodes



Zeile 59 (493- 503) CCD-Barcodes



Zeile 60 (503- 512) CCD-Barcodes



Zeile 61 (512- 521) CCD-Barcodes



Zeile 62 (521- 530) CCD-Barcodes



Zeile 63 (530- 539) CCD-Barcodes



Zeile 64 (539- 546) CCD-Barcodes



Zeile 65 (546- 554) CCD-Barcodes



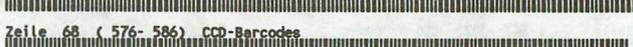
Zeile 66 (554- 564) CCD-Barcodes



Zeile 67 (564- 576) CCD-Barcodes



Zeile 68 (576- 586) CCD-Barcodes



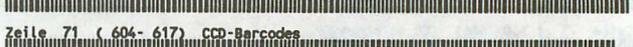
Zeile 69 (586- 595) CCD-Barcodes



Zeile 70 (595- 604) CCD-Barcodes



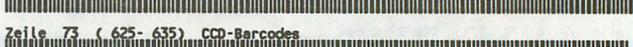
Zeile 71 (604- 617) CCD-Barcodes



Zeile 72 (617- 625) CCD-Barcodes



Zeile 73 (625- 635) CCD-Barcodes



Zeile 74 (635- 646) CCD-Barcodes



Zeile 75 (646- 652) CCD-Barcodes



Zeile 76 (652- 660) CCD-Barcodes



Zeile 77 (660- 666) CCD-Barcodes



Zeile 78 (666- 673) CCD-Barcodes



Zeile 79 (673- 680) CCD-Barcodes



Zeile 80 (680- 688) CCD-Barcodes



Zeile 81 (688- 698) CCD-Barcodes



Zeile 82 (698- 708) CCD-Barcodes



Zeile 83 (708- 715) CCD-Barcodes



Zeile 84 (715- 725) CCD-Barcodes



Zeile 85 (725- 736) CCD-Barcodes



Zeile 86 (736- 740) CCD-Barcodes



Zeile 87 (740- 749) CCD-Barcodes



Zeile 88 (749- 759) CCD-Barcodes



Zeile 89 (759- 765) CCD-Barcodes



Zeile 90 (765- 775) CCD-Barcodes



Zeile 91 (775- 780) CCD-Barcodes



Zeile 92 (780- 788) CCD-Barcodes



Zeile 93 (788- 797) CCD-Barcodes



Zeile 94 (797- 804) CCD-Barcodes



Zeile 95 (804- 813) CCD-Barcodes



Zeile 96 (813- 820) CCD-Barcodes



Zeile 97 (820- 829) CCD-Barcodes



Zeile 98 (829- 839) CCD-Barcodes



Zeile 99 (839- 844) CCD-Barcodes



FLCS (Peter Lüthi) 1613

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes



Zeile 2 (3- 6) CCD-Barcodes



Zeile 3 (6- 12) CCD-Barcodes



Zeile 4 (12- 19) CCD-Barcodes



Zeile 5 (19- 28) CCD-Barcodes



Zeile 6 (28- 37) CCD-Barcodes



Zeile 7 (37- 47) CCD-Barcodes



Zeile 8 (47- 55) CCD-Barcodes



Zeile 9 (55- 62) CCD-Barcodes



Zeile 10 (62- 70) CCD-Barcodes



Zeile 11 (70- 78) CCD-Barcodes



Zeile 12 (78- 87) CCD-Barcodes



Zeile 13 (87- 96) CCD-Barcodes



Zeile 14 (96- 104) CCD-Barcodes



XROM (Armin Jakob) 1614

Computerclub Deutschland e.V.

Zeile 1 (1- 5) CCD-Barcodes



Zeile 2 (5- 9) CCD-Barcodes



Zeile 3 (9- 17) CCD-Barcodes



Zeile 4 (17- 25) CCD-Barcodes



Zeile 5 (25- 33) CCD-Barcodes



Zeile 6 (33- 42) CCD-Barcodes



Zeile 7 (42- 51) CCD-Barcodes



Zeile 8 (51- 55) CCD-Barcodes



Zeile 9 (55- 63) CCD-Barcodes



Zeile 10 (63- 72) CCD-Barcodes



Zeile 11 (72- 82) CCD-Barcodes



Zeile 12 (82- 91) CCD-Barcodes



Zeile 13 (91- 101) CCD-Barcodes



Zeile 14 (101- 109) CCD-Barcodes



Zeile 15 (109- 117) CCD-Barcodes



Zeile 16 (117- 124) CCD-Barcodes



Temperaturverlauf in einem Wärme- quellennetzwerk

PRISMA 87.03.20

Die Anmerkung der Redaktion war nicht ganz vollständig, die angesprochenen Unterprogramme N und E sind im Programm Zeile 9998 und 9999 implementiert, werden also bei Verwendung anderer Drucker mit anderen Befehlssequenzen und Streichen der Aufrufe in Zeile 290 überflüssig.

Ankündigung:

Die im PRISMA 87.03.19 „Temperaturverlauf in einem Wärmequellennetzwerk“ zu „Grundlagen“ angekündigte Arbeit: K. Lutz und E. Thum, „Erweitertes Wärmequellennetzwerk zur Berechnung instationärer Zustände“ ist in „Zeitschrift für elektrische Energietechnik“, etz-Archiv Band 9, 1987, Heft 6, Seite 199-201 erschienen.

Das angekündigte Programm DCAT ist derzeit noch in der Erprobungsphase.

(Martin Meyer (1000), Redaktion)

K. Lutz
Bayernstraße 39
D-8501 Rückersdorf

Fortsetzung von Seite 40

04A:	F101	AF07	2A11	028A	24
04B:	8141	198B	EA0E	2109	B8
04C:	55F8	7160	76A1	1117	20
04D:	F911	1278	9111	2119	6A
04E:	EA7C	8187	1901	1270	B7
04F:	8111	2D86	84D8	8842	5B
050:	4850	80DF	AF2A	7E10	EE
051:	9892	6689	3137	6114	C3
052:	167D	115A	5814	8AC2	F3
053:	5948	DOA4	C94C	54B4	9A
054:	4119	AC61	0977	E042	0E
055:	37EE	05B2	8148	AC32	E2
056:	A444	D181	CAC6	80DF	C1
057:	1198	0C51	0977	B056	EB
058:	06F7	D7AB	057F	119D	E8
059:	6AF0	75B0	1008	AA3E	5A
05A:	C64E	DEA4	9D13	7135	A6
05B:	CA13	314F	D3E7	25A0	11
05C:	E4E0	A675	6FBE	E5E4	54
05D:	8705	2131	1101	37C2	BC
05E:	137A	F00E	F796	A702	0C
05F:	EB04	6F5C	94A5	1B46	3B
060:	571D	50EF	7DD9	69A0	5C
061:	FF0E	F7D3	0EFF	14D1	9F
062:	3111	0632	C884	3384	61
063:	062D	E8DD	8DB0	8DB1	1B
064:	3B11	7F8D	322B	18D8	73
065:	3DB0	1BBB	8F21	4213	35
066:	0011	321B	BB8F	2140	53
067:	01AF	1D88	2281	D81D	4F
068:	137C	9C91	3513	4832	3B
069:	7151	11C1	14A1	4F14	E3
06A:	914C	161C	D5BE	03F	C7

Aktive Werbung für den CCD

Liebe Clubmitglieder,

es ist immer wieder wichtig, für unseren Anwenderclub im Bekanntenkreis Werbung zu machen. Ob Familie, Schule, Universität, Arbeitsplatz oder Geschäftskollegen - der CCD ist für ALLE interessant, die sich mit HP-Taschencomputern oder auch mit Personal Computern unter MS-DOS befassen.

Außerdem befindet sich bei uns im Club auch noch eine ATARI-Gruppe, die sich bislang aber noch im "Test-Stadium" befindet und deshalb auch noch nicht im gegenüberliegenden Anmeldeformular aufgezählt ist.

Happy Marketing wünscht

Alf-Norman Tietze

Clubadressen:

1. Vorsitzender

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

2. Vorsitzender

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

Schatzmeister Mitgliederverwaltung

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

1. Beisitzer

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

2. Beisitzer Geowissenschaften

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

PRISMA-Nachsendedienst

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

Programm-Bibliothek HP-41

Beirat

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

Programm-Bibliothek HP-71

Henry Schimmer (786)
Homburger Landstraße 63
6000 Frankfurt 50

Serie 80 Service

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

Beirat

MS-DOS Service

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

Hardware 41

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

Grabau GR7 Interface

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

CP/M-80 Service

Peter-C. Spaeth
Michaeliburgstraße 4
8000 München 80

E-Technik

Werner Meschede (2670)
Sorpestraße 4
5788 Siedlingshausen

Mathematik

Andreas Wolpers (349)
Steinstraße 15
7500 Karlsruhe

Vermessungswesen

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

Regionalgruppe Berlin

Jörg Warmuth (79)
Wartburgstraße 17
1000 Berlin 62

Regionalgruppe Hamburg

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

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

Beirat

Regionalgruppe Karlsruhe

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

Regionalgruppe Köln

Frank Ortman (1089)
Okerstraße 24
5090 Leverkusen 1

Regionalgruppe München

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

Regionalgruppe Rhein-Main

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

Beirat

Peter Kemmerling (2466)
Danzigerstraße 17
4030 Ratingen

Beirat

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

Beirat

Günther Schwarz (2658)
Bodelschwingstraße 34
3408 Duderstadt 1

Atari Service

Werner Müller
Classen-Kappellmannstr. 30 a
5000 Köln 41

Der CCD e. V. bietet seinen Mitgliedern:

- Kostenloser Bezug der Clubzeitschrift PRISMA
- Programmierhilfen für Anfänger und Fortgeschrittene
- Anwenderprogramme aller Art
- Informationen über neue Produkte
- Erfahrungsberichte
- Informationen über Hardwareumbauten und -ergänzungen
- Einblick in die synthetische Programmierung
- Einblick in die Maschinenspracheprogrammierung
- Zugriff auf umfangreiche Programmbibliotheken
- Vergünstigungen bei der Anschaffung von Geräten und Zubehör
- Kostenlose Kleininserate in der Clubbörse
- Zugriff auf die CCD-Mailbox

MITGLIEDSCHAFT

Aufnahmegebühr:	Firmen	DM 160,—
	Schüler, Studenten	DM 20,—
	alle anderen	DM 40,—
Jahresbeitrag:	Taschenrechner	DM 60,—
	CP/M	DM 150,—
	MS-DOS	DM 150,—
	CP/M + MS-DOS	DM 210,—
Zuschlag:	Ausland Europa	DM 20,—
	übriges Ausland	DM 50,—

(In den Beiträgen der CP/M- bzw. MS-DOS-Gruppe ist der Bezug von monatlichen INFO-Disketten enthalten).

AUNAHMEANTRAG

Ich will dem CCD e.V. beitreten und lege die Aufnahmegebühr und einen Jahresbeitrag als Verrechnungsscheck über _____ bei.

Name: _____
 Vorname: _____
 Firma: _____
 Straße: _____
 PLZ, Ort: _____

Computerclub Deutschland e. V.

Schwalbacher Straße 50

6000 Frankfurt 1

(Unterschrift)

MS-DOS HP-Taschencomputer CP/M

PRISMA

Computerclub Deutschland e. V. · Schwalbacher Str. 50 · 6000 Frankfurt 1

Die ständig wachsende Leistungsfähigkeit der Microcomputer erfordert immer tiefer gehende Kenntnisse über Aufbau, Arbeitsweise und Einsatzmöglichkeiten dieser Geräte. Um dies zu erreichen, haben wir 1981 den CCD e. V. als unabhängigen Anwenderclub gegründet und haben zur Zeit über 2000 Mitglieder. Über unsere Clubzeitschrift PRISMA unterhalten wir einen intensiven Erfahrungsaustausch.

Unser Ziel ist, den Benutzern der genannten Systeme beim Einsatz der Geräte Hilfestellung zu leisten und Ihnen zu ermöglichen, die Fähigkeiten dieser Computer voll auszunutzen. Weiterhin wollen wir unseren Mitgliedern Kenntnisse vermitteln, die weit über die Hersteller-Informationen hinausgehen.

Falls ein Mitglied zur Lösung eines Problems Hilfe braucht, kann es über die Clubleitung und/oder die Clubzeitschrift Kontakt zu anderen Mitgliedern schaffen. Somit profitiert jedes Mitglied vom Wissen der anderen und kann mit eigenen Ideen und Vorschlägen die Clubarbeit unterstützen.

Die bestehenden Personal-Computer-Gruppen erweitern den Informationsaustausch durch monatlich erscheinende Informationen auf Disketten.

Über die Datex-P-Mailbox des CCD werden weitere Informationen ausgetauscht.



Computerclub Deutschland e. V. · Schwalbacher Str. 50 · 6000 Frankfurt 1

Postvertriebsstück
Gebühr bezahlt

D 2856 F

CCD – Computerclub Deutschla
Schwalbacherstraße 50
D-6000 Frankfurt am Main 1

CCD

ISSN 0176-8735

PRISMA

Juli/August 1988 Nr. 4

In USA neue HP Taschenrechner !

HP-22S und HP-32S

Der HP-22S ist ein wissenschaftlicher Taschenrechner mit algebrarischem Eingabesystem - also ohne (!) UPN und hat folgende Eigenschaften aufzuweisen:

- NICHT programmierbar
- Algebrarisches Eingabesystem
- über 200 wissenschaftliche Funktionen
- Statistik für zwei Variablen mit Korrelation, Regression und Gewichtung
- Wahrscheinlichkeitsrechnung mit Fakultät und Kombinatorik
- Basismwandlung: 16, 10, 8 und 2
- Einheitenrechnung für Masse, Temperatur, Länge und Volumen
- Finanzmathematik: Cash-Flow

- Polar/Rechtwinklig und HMS, HR
- Polynomische Gleichungen
- Nullstellenbestimmung

Listenpreis: 60.- US\$

Etwas anders der neue HP-32S. Dieser ist ein technisch-wissenschaftlicher Taschenrechner mit UPN Eingabesystem und etwas Speicherplatz. Seine Eigenschaften lassen sich wie folgend auflisten:

- Programmierbar (390 Bytes)
- 27 Speicherregister
- UPN Eingabesystem
- Menüs und Softkeys

- über 180 wissenschaftliche Funktionen
- alles, was der HP-22S hat, jedoch ohne finanzmathematische Funktionen
- Numerische Integration
- Komplexe Zahlen
- Zufallszahlen

Listenpreis: 70.- US\$

Beide Taschenrechner haben ein vertikales Design, ähnlich wie der schon bekannte HP-17B, d.h. also nur eine "Hälfte" des HP-28. Markteinführung und Preise für Deutschland sind uns zur Zeit noch unbekannt. Möglicherweise kann da eine Anfrage bei HP bereits Klarheit verschaffen.

(ant)

*Hardware · Software
Servicestation
Beratung · Zubehör*

OSBORNE

Management by Computer.

WORDLORD · Textverarbeitung · CAD-Anwendungen · Komplettsysteme

PCE

PFORTNER GMBH

Computer-Technik · Elektronik

Branchenlösung für Klein- u. Mittelbetriebe

Postfach 1220 · 4133 Neukirchen-Vluyn

Telefon 0 28 45/3 22 94



Sonderpreise für CCD-Mitglieder