

# PRISMA

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

1988 Nr. 1

D 2856 F



Das ist ER! Der neue Handheld-Computer Marys II von CMT. Mit der geballten Leistung eines 8088-Prozessors bietet er vielfältige Möglichkeiten zum „Out-Door-Einsatz“. Ein erster Bericht zeigt die Leistungsmerkmale dieses neuen Computers.

Aber auch Hewlett-Packard ruht sich nicht aus. Für März '88 sind drei neue Taschenrechner angekündigt. Näheres ist dazu auf Seite 22 in diesem Heft nachzulesen.

## Clubnachrichten

Mitgliederversammlung  
Clubbörse  
Reise nach USA  
NEU: Marys II

## Grundlagen

Technisch-wissenschaftliche  
Textverarbeitung auf PC's

## MS-DOS

Referenzblätter

## Serie 80

HP-Serie 80 Story

## Serie 70

Lineare Netzwerke  
Update zum Handbuch  
Telefonverzeichnis  
Gas  
Lexfile FINDLEX

## Serie 40

Neue HP-Taschenrechner  
Die Barcodes des HP-41  
Dreiecksberechnungen  
Jacobi-Verfahren  
Messen mit HP-IL  
Tips, Tricks & Utilities  
Gleitpunkt-Arithmetik  
Steuer '88  
Tschebyscheff Bandpaß

Zubehör für HP41CV wg. Systemwechsel zu verkaufen:  
 Kartenleser DM 200,-  
 Barcode-Leser DM 200,-  
 XT-Modul DM 100,-  
 CCD-Modul DM 200,-  
 Mathe-Modul DM 50,-  
 Statistik-Modul DM 50,-  
 ca. 100 CCD-Magnetkarten DM 100,-  
 Prisma alle Hefte mit C. Wickes  
 und Best of Prisma DM 100,-  
 Alle Zubehörteile mit Original-  
 Anleitung, Gesamtabnahme Preis DM 900,-  
 Alle Preise sind incl. Porto und VB-Preise  
 CCD 2209 P. Weiss, ☎ 030/4653874 (ab 17.00 Uhr)

**HP 75 + 8 K**, Visicalc, Text, JL und Kartenleser, nur  
 komplett für DM 850,-  
**HP 110** mit Doppel-Diskette, Think-Jet-Drucker, Word  
 Star und d-Base II, nur komplett für DM 5.000,-.  
 Kurt Heller, Köln, Büro: ☎ 0221/492182,  
 privat: ☎ 0221/4971316

Computer **OSBORNE** 1 DQD, 64 kB, 2 LW, int./ext.  
 Monitor, Bj. 83 seriell/par./Modem-Schnittst., Softw.:  
 Wordstar, Dbase, Super calc, Fibu, Basic, Turbo-Pasc.  
 usw., VB 750,- DM, ☎ 0201/233233, H. Szostak

Zu verkaufen: **HP 71B** mit IL-Modul, PAC-Screen Inter-  
 face und Diskettenlaufwerk HP 9114; zusammen  
 DM 5000,-; alle Teile sind neuwertig.  
 Schwanhäuser, Karl-Loewe-Str. 8, 7550 Rastatt,  
 ☎ 07222/69448 (abends)

**FORTH/ASSEMBLER ROM** für den HP 71B, HP 82441A  
 für 200,- DM, Plotter-Handbuch zum Modul für den HP  
 41C/CV/CK 20,-, Debugger Software auf 5 1/4" Diskette  
 für HP 71B Lexfiles lauffähig auf HP Vectra 50,- DM,  
 Disassembler HP 71B, Programmsammlung Vermes-  
 sungswesen HP 41 und HP 71 sowie Lexfilesammlung  
 auf Anfrage gegen einen Freiumschlag. Ulrich Kulle,  
 Schnuckentritt 14, 3000 Hannover 51, ☎ 0511/6042728

Verkaufe **HP-41 CX** mit Magnetkartenleser, Barcodeleser,  
 Akku, Ladegerät, 2 X-Memory-Module, Standard-Modul,  
 Mathematik-Modul, 180 Magnetkarten, Literatur (Jarrett,  
 Dearing). Nur komplett für 750,- DM.  
 Normann, ☎ 0541/387592

Cassettenlaufwerk **HP 82161 A** incl. 9 Cassetten,  
 550,- DM, Doppelmodul MATH/MACHINE DESIGN  
 incl. Handbücher 100,- DM, Thomas Mareis, Cranach-  
 straße 3, 8000 München 40, ☎ 089/1296844

Wegen Systemwechsel zu verkaufen: **IL Cassetten-**  
**laufwerk** inkl. 3 Cassetten für 550,- DM; **Thinkjet IL**  
 580,- DM; Plottermodul 90,- DM; Buch „Das HP IL  
 System“ für 15 DM. Alles original verpackt, absolut neu-  
 wertig aus 1. Hand. ☎ 0421/656439 Bremen.

**HP 110** mit Lotus 123, HPLink Memomaker, Turbo Pascal  
 2000,- VB; Thinkjet-IL 700,- DM; IL-Karte für Epson-  
 Drucker 350,- DM; HP 82166A IL-Converter 350,- DM;  
 HP75-Visicale 300,- VB. Achim Bähr, Holzkirchener  
 Str. 20a, 8029 Arget, ☎ 08104/ 484. bzw 089/21952850  
 tagsüber.

Verkaufe **X-Function-Modul** mit kopierter Anleitung,  
 90,- DM, Dennis Föh ☎ 04721/24129

Suche HP-9114 A oder B; Software für HP-110 (z.B.  
 GW-Basic) ☎ 07131/484738

Verkaufe Plotter- und Mathemodul (je 20,- DM); Akku neu,  
 82033A (20,- DM); 50 blanko Tastaturmasken, 2 Rollen  
 Thermopapier, Modultasche, Barcodeaufkleber, HP41  
 Ständer (zus. 30,- DM).  
 Programmsammlungen: Statistik, Geometry, High Level  
 Math, Games II, Games, Business, Time Module Sol.  
 (je 10,- DM, zus. 50,- DM);  
 Wickes, Jarrett 2x Synthe. Progr.; Qick-Ref.-Card; 2 Byte-  
 Tabellen; Gosmann, Anwenderhandbuch; Thorssen Bar-  
 codes (je 10,- DM, zus. 40,- DM).  
 Michael Bering, Gartenstraße 9, 6929 Angelbachtal,  
 ☎ 07265/7392

Verkaufe **HP-2225B** THINK JET mit IL Schnittstelle (f.  
 HP41/71/75 m. entsp. Interface), 555,- DM; HP-41  
 STANDARD/GAMES MODUL in einem Gehäuse,  
 55,- DM; HP-41 Barcodeleser Zub. 10,- DM; TI 58/59  
 Buch, kein 08/15 Schmöcker, etwas f. Freaks, 20,- DM;  
 Preise VHB; ☎ 07131/484738

Verkaufe **HP-71 B** mit IL-Modul zusammen DM 900,-,  
 HP82161A Cassettenlaufwerk mit 10 Cass. DM 650,-,  
 HP82162A IL-Thermodrucker DM 500,-, HP2225B Think-  
 Jet neuwertig DM 900,-, 8 kByte-Modul f. HP-75 (neu)  
 DM 100,-; Volker Lang (130) ☎ 07138/5349, ab 18 Uhr.

Verkaufe **HP 75C** mit Text-Formatter Modul (in bestem  
 Zustand, jedoch mit altem Akku) für DM 450,-  
 Cassettenlaufwerk (IL) HP 82161A für DM 450,-  
 Video Interface HP für DM 150,-  
 IL-Modul HP 82160A für HP 41 für DM 120,-  
 Stefan Schmidt (2437), Schillerstraße 24, 8972 Sonthofen  
 ☎ 08321/4794

Verkaufe:  
**IL-ThinkJet HP2225B** mit ca. 1000 Bl.  
 Endlospapier 12" DM 950,-  
**IL-Diskettenlaufwerk HP-9114B**  
 mit 2 Disketten DM 1070,-  
**IL-Plotter HP 7470A** mit div.  
 Staedtler-Tuschespitzen, 0,3-0,7 und  
 Marsonic 580 U-Schallreinigungsgerät DM 2125,-  
**HP-75C mit I/O-ROM** und Graphics-  
 Solutions auf M-Karte DM 1500,-  
**Plottermodul für HP-41** DM 180,-  
**HP 16** DM 195,-  
**Real-Estate-Pac, Home-Management-**  
**Pac, Securities-Pac und Advantage-Pac** je DM 65,-  
 Alle Geräte mit Original-Bedienungshandbüchern.  
 Konrad Albers, Südring 42, 2300 Klausdorf/Schwentine  
 Tel.: tagsüber 0431/92055 nach 18.00 Uhr 0431/79494

**Schneider PC 1512**  
 neu, original verpackt, 2 Disk-Laufw., Monitor DM 990,-  
**Sanjo MPC 555 2 DL**, Monitor, orig. verp. DM 1190,-  
 Jeans-Tankstelle GmbH, Postf. 1449, 7880 Bad Säckingen  
 Tel.: 07761/1446

## Regionalgruppen

### Karlsruhe

In Karlsruhe trifft man sich jeden 2. Mitt-  
 woch eines Monats zum Erfahrungsges-  
 austausch.

Kontakt: Stefan Schwall, (0721) 576756

### München

Auch dort trifft man sich an jedem 2.  
 Mittwoch eines Monats in der Gaststätte  
 Schützenring, Weißenburger Platz, 8000  
 München, S-Bahn-Haltestelle Rosen-  
 heimerplatz

Kontakt: Victor Lecoq, (089) 789379

### Rhein-Main

In Raunheim (bei Rüsselsheim) treffen  
 sich die Mitglieder an jedem letzten  
 Samstag eines Monats.

Kontakt:

Andreas Eschmann, (06142) 46642

## Regionalgruppe Köln

Welche Vorteile bietet eigentlich ein Besuch  
 bei der Regionalgruppe Köln, die sich regel-  
 mäßig einmal pro Monat in Leverkusen trifft?

Nun, ich möchte hier einige Punkte auf-  
 führen:

- Austausch von Programmen
- Erfahrungsaustausch über alle Hardware
- Zugang zu anderen Clubs (PPC, CHHU)
- Zugang zur amerikanischen Literatur
- fast alle Literatur über HP 41/71 (z.B. IDS)
- Brennen von Eproms
- Bezug von Programmen aus der US User-  
Library
- gemeinsamer Kauf von Hardware in USA  
(sehr günstig)
- eine tolle, wenn auch konzentrierte At-  
mosphäre

Zudem will die Regionalgruppe Köln im  
 Rahmen der Treffen kurze Referate über  
 Probleme der HP-Computer halten. Ged-  
 acht ist, daß ein Referent einzelne Aspekte  
 der Programmierung oder Hardware für alle  
 verständlich erklärt. Als Themen stehen an:

- Wie bediene ich überhaupt meinen HP?  
(Vom Einschalten bis zum Ausschalten;  
alles)
- HP-Computer und Video-Interface
- HP und Speichererweiterungen; RAM-  
DISCS
- HP-71 und Data bzw. Text-Files
- HP-71 Assembler, LEX-Files
- HP-Module, ein Überblick
- Graphik auf Think Jet

Wie Ihr seht, eigentlich eine ganze Menge.  
 Was müßt Ihr jetzt noch tun? Sucht Euch  
 den nächsten Termin und im Stadtplan  
 Rheindorf heraus!

Bis bald, wir freuen uns über jeden Interes-  
 senten.

### Termine für das Regionaltreffen im Jahr 1988

Alle Termine jeweils am zweiten Sonntag  
 im Monat um 15.00 Uhr

- 13. März
- 10. April
- 8. Mai
- 12. Juni
- 10. Juli
- 14. August
- 11. September
- 9. Oktober
- 13. November
- 11. Dezember

### Im Wiesdorfer Treff

Hauptstraße 133-139  
 5090 Leverkusen 1 (Wiesdorf)

(Bitte beim Pförtner anmelden)

Mit den öffentlichen Verkehrsmitteln ist der  
 Wiesdorfer Treff: Linie 211, Haltestelle  
 „Adolfstraße“ zu erreichen.

Terminänderungen werden nach Möglich-  
 keit im PRISMA bekanntgegeben.

Für Rückfragen stehen Euch zur Verfügung:

Werner Weith, Telefon 0201/7 25 33 05  
 dienstlich 9 bis 16 Uhr  
 02126/24005 privat

Frank Ortman, Telefon 0214/272 18

Mit freundlichem Gruß  
 Frank Ortman

## Wichtige Clubadressen

Der CCD kann seinen Mitgliedern schon seit längerem ein paar interessante "Dienstleistungen" anbieten, jedoch werden diese - im Verhältnis zur Mitgliederzahl - selten genutzt. Das liegt möglicherweise daran, daß diese Dienste nur bei einer näheren Betrachtung der Clubadressen (ab 1988 auf der vorletzten Seite in jedem Heft) entdeckt worden sind. Dazu gehören der PRISMA-Nachsendedienst, die Programm-Bibliotheken, der "Service" für HP Serie 80, CP/M und MS-DOS (Personal Computer), sowie die Ansprechpartner für einzelne Fachgebiete und oder Regionalgruppen.

Alle diese "Dienste" sind nur möglich, weil aktive Clubmitglieder ihre Mitarbeit ehrenamtlich zur Verfügung stellen. Diese Arbeit geschieht überwiegend nach "Feierabend" während der persönlichen Freizeit. Deshalb wäre es auch verkehrt, dahinter einen professionellen Service zu erwarten. Es kann deshalb schon mal einige Zeit in Anspruch nehmen, bis eine Anfrage beantwortet wird. Trotzdem kann der Kontakt zu anderen Mitgliedern oft weiterhelfen,

## Impressum

**Titel:**  
PRISMA  
**Herausgeber:**  
CCD-Computerclub Deutschland e.V.  
Postfach 11 04 11  
Schwalbacher Straße 50  
6000 Frankfurt am Main 1  
**Verantwortlicher Redakteur:**  
Alf-Norman Tietze (ant)  
**Redaktion:**  
Hans Jürgen Hübner (hjh)  
Klaus Kaiser (kk)  
Michael Krockner (mik)  
Martin Meyer (mm)  
Henry Schimmer (hs)  
Dieter Wolf (dw)  
**Herstellung:**  
CCD e.V.  
**Manuskripte:**  
Manuskripte werden gerne von der Redaktion angenommen. Honorare werden in der Regel nicht gezahlt. Die Zustimmung des Verfassers zum Abdruck wird vorausgesetzt. Für alle Veröffentlichungen wird weder durch den Verein noch durch seine Mitglieder eine irgendwie geartete Garantie übernommen.

**Druck und Weiterverarbeitung:**  
Reha Werkstatt Rödelheim  
Biedenkopfer Weg 40 a, 6000 Frankfurt  
**Anzeigenpreise:**  
Es gilt unsere Anzeigenpreisliste 3 vom Juni 1987  
**Erscheinungsweise:**  
PRISMA erscheint jeden 2. Monat.  
**Auflage:**  
3000  
**Bezug:**  
PRISMA wird von allen Mitgliedern des CCD ohne Anforderung übersandt. Ein Anspruch auf eine Mindestzahl von Ausgaben besteht nicht. Der Bezugspreis ist im Mitgliedsbeitrag enthalten.

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

wenn es darum geht, irgendwelche Computer- oder Softwareprobleme zu lösen.

Aus diesem Grund empfehle ich die Möglichkeit, über die Clubadressen weitere Kontakte zu anderen Mitgliedern zu knüpfen, häufiger zu nutzen - ganz besonders gilt hier das Angebot, die Treffen der Regionalgruppen zu besuchen.

... und wenn einer nur einen kennt, der einen anderen kennt, der mal einen gesehen hat, der weiß wie's gemacht wird.

Happy Programming

Alf-Norman Tietze  
(verantwortlicher Redakteur)

## Inhalt

<b>Clubnachrichten</b>	
Clubbörse	2
Regionalgruppe Köln	2
Impressum	3
"Wichtige Clubadressen"	3
Mitglieder-Support	4
Offener Brief (Koppel)	4
Clubreise nach USA	5
NEU: Marys II	5
Antwortbrief	7
<b>Grundlagen</b>	
Technisch-wissenschaftliche Textverarbeitung auf Personal Computern	8
<b>MS-DOS</b>	
MS-DOS Referenzblätter	20
<b>Serie 80</b>	
HP-Serie 80 Story	11
<b>Serie 70</b>	
Lineare Netzwerke	13
Update zum HP-71 Benutzerhandbuch	17
Telefonverzeichnis	17
Gas	18
Lexfile: FINDLEX	19
<b>Serie 40</b>	
Neue HP Taschenrechner	22
Die Barcodes des HP-41	23
Dreiecksberechnungen	27
Jacobi-Verfahren	30
Messen mit HP-IL	33
Tips & Utilities	35
Tips & Tricks für HP-41 Benutzer	36
20/40-stellige Gleitpunkt-Arithmetik	37
Steuer '88	40
CAT-Bug	42
Tschebyscheff Bandpaß	44
<b>Barcodes</b>	46
<b>Clubadressen</b> in der nächsten PRISMA-Ausgabe	

## Mitglieder- versammlung

Am Samstag, den 16. April 1988  
um 11 Uhr,  
im Bürgerhaus Frankfurt-Nordwest,  
Niddacorso 5,  
6000 Frankfurt am Main  
(Nordweststadt)

Tagesordnung:

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

Liebe Mitglieder,

mindestens einmal im Jahr, möglichst im 1. Quartal, soll die ordentliche Mitgliederversammlung stattfinden. Eis und Schnee im Februar, CEBIT und Hannover-Messe und sehr frühe Osterferien haben uns jedoch Samstag, den 16. April, als geeignetesten Termin erscheinen lassen. Frankfurt als zentral gelegene Stadt, an der auch unsere Geschäftsstelle ist, bietet sich geradezu als Tagungsort an.

Das Bürgerhaus in Frankfurt Nordwest haben wir als Tagungsort ausgesucht, da wir dort während der Mittagsunterbrechung der Mitgliederversammlung gemeinsam essen können und da wir dort nach der Versammlung in einem Nebenraum Gelegenheit haben, unsere diversen "Kisten" auspacken oder auch nur so einen Erfahrungsaustausch über unser Hobby durchzuführen.

Die persönliche Kontaktnahme mit anderen Mitgliedern ist schon bisher vielen Mitgliedern Grund gewesen, auch "meilenweite" Wege in Kauf zu nehmen, um zum Treffen zu kommen. Die Einladung gilt, nehmt sie wahr und kommt!

Wolfgang Fritz  
1. Vorsitzender

Es befinden sich ausreichende Parkmöglichkeiten im "Untergeschoß" des Nordwest-Zentrums, wo die Mitgliederversammlung auch stattfindet.

## Mitglieder-Support

Der CCD als ein bundesweit - ohne Unterorganisationen - tätiger Computerclub muß Möglichkeiten zur Unterstützung seiner Mitglieder bei Problemlösungen im Hard- und Softwarebereich wie auch bei der Entwicklung spezieller Anwendungen für die im Club verbreiteten Computer schaffen.

Sicherlich stellt unsere Zeitschrift "Prisma" eine geeignete Basis für den Erfahrungsaustausch dar. Sie ist schlechthin das Medium, um der breiten Öffentlichkeit Meinungen und Beiträge vorzustellen. Speziellere Lösungen, die vielleicht auf bestimmte Einzelfragen eingehen, finden dafür nicht ihren Platz im "Prisma".

Nun gibt es seit langer Zeit Regionalgruppen im CCD. Das sind Vereinigungen interessierter Mitglieder einer Region, die sich regelmäßig zum Gedankenaustausch treffen. Hervorgegangen sind diese Treffs aus dem anfänglichen Bemühen, die Programmierung des HP41 zu lernen und zu verbessern. Heute ist es so, daß in den bestehenden Gruppen auch andere Computersysteme bis zum PC vertreten sind. Einzelne interessierte Mitglieder haben

sich in lobenswerter Weise dafür eingesetzt, Regionalgruppen zu organisieren oder bestimmte Fachgebiete zu betreuen (siehe Editorial "Wichtige Clubadressen"). So gibt es in verschiedenen Regionen diese Einrichtungen. In manchen Gebieten, die sich durch einen hohen Mitgliederbestand auszeichnen, gibt es solche Einrichtungen noch nicht. Deshalb möchte ich, als das für die Mitgliederwerbung verantwortliche Vorstandsmitglied, engagierte Mitglieder auffordern Regionalgruppen zu organisieren.

Die Erfahrung hat gezeigt, daß hier auf kommunale Räume als Veranstaltungsort zurückgegriffen werden kann. Kosten, die bei der Anmietung solcher Räume entstehen, übernimmt der CCD nur nach vorheriger Rücksprache (!). Auch wird diesen Initiatoren Unterstützung dadurch gewährt, daß die Mitgliederverwaltung vorbereitete und vom Vorstand genehmigte Werbeaktionen durch Versendung an die im Einzugsgebiet solcher Veranstaltungen wohnenden Mitglieder durchführt. Für spezielle Fragen in einem solchen Zusammenhang stehe ich bereitwilligen Initiatoren

in den Abendstunden telefonisch unter 0201-263045 zur Verfügung.

Wer fühlt sich angesprochen im Großraum Ruhrgebiet, z.B. in Bochum eine Regionalgruppe Rhein-Ruhr aufzuziehen? Oder in Stuttgart, in Berlin usw.. Das geht auch in nicht so dicht besiedelten Regionen, wie beispielsweise in Bielefeld.

Allen, die sich angesprochen fühlen, kann ja der Katalog des vorstehenden Aufrufs als Einstieg in die Sache dienen. Diese Themen interessieren bestimmt nicht nur in Köln, sondern ganz bestimmt auch anderswo.

Ich meine, daß der CCD nur vom Geben und Nehmen der Mitglieder - und das ausdrücklich im wohlmeinenden Sinn - richtig existieren kann. Das bringt Schwung und macht Außenstehende auf den Club aufmerksam und sorgt für neue Mitglieder. Wer macht mit, ich helfe gerne mit Ideen und Unterstützung durch den Vorstand.

Erich H. Klee (1170)  
- 2. Vorsitzender -

## Offener Brief

Liebe CCD'ler

nun bin ich schon über zwei Jahre Mitglied im CCD und habe es noch nicht ein einziges Mal geschafft, wenigstens einen kleinen Beitrag zu schreiben und eventuell das eine oder andere Programm beizusteuern. Ich denke, daß ich wie viele andere Mitglieder die gleiche auf etwas wackligen Füßen stehende Ausrede benutze, wenn ich sage, daß ich einfach keine Zeit habe, um ab und zu einen Beitrag zu schreiben. Nun, damit in Zukunft wenigstens ein neuer Name unter den erscheinenden Artikeln steht, habe ich mir für's neue Jahr vorgenommen, etwas aktiver zu werden.

Um dem Problem der nachlassenden Programmbeiträge entgegenzutreten kann ich einige Vorschläge machen:

Prisma ist eine Zeitschrift von Lesern für Leser. Es verlangt niemand Perfektion. Ein eingesandtes Programm wird mit Sicherheit auch dann positiv aufgenommen werden, wenn es nicht mit dem raffiniertesten Techniken synthetischer Programmierung ausgestattet ist (betrifft Serie 40). Serie 80 Beiträge müssen auch nicht unbedingt in Assembler geschrieben sein.

Das eine oder andere Programme was u.U. schon längst in der Schublade eines Mitgliederschreibtisches verschwunden ist kann anderen Mitgliedern vielleicht noch sehr nützlich sein. Es lohnt sich also bestimmt, mal über ältere Programmbestände nachzudenken.

Mit Sicherheit wird die CCD-Redaktion auch einen Erfahrungs- oder Testbericht über eine Neuheit auf dem von Prisma betreuten Sektor veröffentlichen.

Wenn sich jemand als Fachmann für einen bestimmten Themenkreis fühlt, so kann er meiner Ansicht nach durchaus an die Redaktion herantreten und fragen, ob Interesse an einem Beitrag vorhanden ist.

Ich glaube, der CCD hat mit seinen Mitgliedern ein undenkbar großes, allerdings zur Zeit noch schlummerndes Potential an Fachwissen in den verschiedensten Fachrichtungen. Wenn dieser Dornröschenschlaf einmal beendet sein sollte, so ist der CCD meiner Ansicht nach durchaus in der Lage, den Computersektor - zumindest auf dem deutschen Markt - nicht unerheblich zu beeinflussen. Schließlich ist der CCD als unparteiischer und im Konzept recht einmaliger Verein bekannt.

Nun noch ein Wort zur Ausgabe Nr. 6/1987. Auf der vierten Seite steht ein kleiner Aufruf, im Club Aldiana als Computerlehrer zu arbeiten. Ich habe keine Ahnung, was näheres hinter diesem Aufruf steckt. Wenn sich allerdings meine Vermutung bewahrheitet, daß auf diese Weise Animatoure gesucht werden, so muß ich sagen, daß ich das sehr schwach finde. In meinem letzten Urlaub - ein Clubaufenthalt - mußte ich feststellen, daß die dort tätigen Animatoure neben verschieden angenehmen Seiten, die sie in Anspruch nehmen können, schwer arbeiten müssen. An dieser Anzeige gefal-

len mir zwei Aspekte überhaupt nicht:

Zum einen kommt NUR-Touristik (mit Sicherheit kein finanzschwacher Konzern) auf diesem Weg zu einem kostenlosen Inserat. Auf diese trickreiche Vorgehensweise sollte die Redaktion in Zukunft besser achten, schließlich müssen alle „Professionellen“ für Ihre Anzeigen in harter DM bezahlen.

Zum Anderen werden (zumindest geht das aus dem Inserat hervor) hochgradige Fachleute mit einem Pseudourlaub und einem leistungsbezogenen Entgelt (und das ist nicht so hoch, wie man erwarten könnte) für eine Sache angeworben, die sie im nachhinein u.U. verteufeln würden. Wer sich also auf dieses Inserat hin melden sollte, der sollte seinen gesunden Menschenverstand nicht an Betracht eines bezahlten Urlaubes außer Kraft setzen.

MfG Andre Koppel (CCD 3030)

### Anmerkung der Redaktion:

„Ein Mann - ein Wort“, der neue Name ist in diesem Heft unter dem Artikel der HP-Serie 80 Rechner und ihrer Geschichte zu finden. Die Redaktion freut sich über jeden Beitrag, besonders dann, wenn er auch das behandelte Problem gut beschreibt. Vielen Dank übrigens für den Hinweis, das Angebot mit dem Computerurlaub kritisch zu prüfen. Für Leute die sonst allerdings keine Möglichkeiten für Urlaubsreisen haben, kann es durchaus ein interessantes Angebot sein.

# Clubreise nach USA

## zur Hand-Held-Conference '88

Mit diesem kurzen Beitrag möchte ich noch einmal auf die ausgezeichnete Möglichkeit einer Reise in die USA hinweisen, die anders als jeder individuelle oder gar Pauschal-Trip ist. Jetzt wo die offizielle Einladung von Corvallis MicroTechnology vorliegt, kann auch gesagt werden was uns dort erwartet.

Neben der eigentlichen Konferenz mit vielen interessanten Beiträgen zu neuen Anwendungen, Programmieretechniken und Hard- wie Softwareentwicklung, die sich über drei Tage vom 4. bis zum 6. August verteilen, findet eine Computer-Messe statt. Hier bietet sich wirklich die einmalige Gelegenheit, Neuigkeiten auf dem PC- und Hand-Held-Sektor zu entdecken; denn bis diese hier in Europa erhältlich sind, ist bekanntlich ein halbes, manchmal sogar ein ganzes Jahr vergangen.

Darüberhinaus findet für Begleitpersonen während der Konferenz eine interessante, dreitägige Bustour durch Naturparks zum Columbia River statt. Die Kosten dafür liegen bei rund 200 Dollar. Den Ausführungen des Veranstalters - Corvallis MicroTechnology - ist zu entnehmen, daß die Teilnehmer etwas Außergewöhnliches zu sehen bekommen.

Neben den in Heft 6/87 vorgestellten Programmen, gibt es jetzt noch eine weitere Reisealternative. Flug von

Amsterdam nach Seattle im Großraumflugzeug, von dort Anschlußflug mit inneramerikanischer Linie nach Portland und Bustransfer nach Corvallis, Abflug am 31.07.88, Rückflug am 07.08.88 mit Ankunft in Amsterdam am 08.08.88, Unterbringung im Doppelzimmer in den von Corvallis MicroTechnology reservierten Hotels zu einem Preis von DM 2.695,00. Dieser Preis entspricht fast dem eines normalen Tickets für den Hin- und Rückflug nach Californien von Europa aus. Für eine 10-tägige USA-Reise nach Californien in der Hochsaison (Der Sommer ist in den USA immer die Reisesaison!) zahlt man sogar noch etwas mehr. Aber es kommt uns ja nicht darauf an, so billig wie möglich nach Californien zu verreisen, sondern zu erschwinglichen Kosten ein nicht alltägliches Treffen amerikanischer und internationaler Computerexperten zu besuchen.

In diesem Sinne, darf ich Euch noch einmal alle auffordern, regen Gebrauch vom anhängenden Coupon zu machen und vielleicht Urlaubsgelüste an europäischen Badestränden gegen die USA-Club-Reise einzutauschen. übrigens verpflichtet die Einsendung des Coupons zu nichts. Dafür bekommt man zunächst einmal das vollständige Reiseangebot, um die endgültige Entscheidung für die Teilnahme und den zu buchenden Reisevorschlag zu treffen.

Also bis zum August in Corvallis, spätestens ...!

Erich H. Klee

# Marys II

## Bekommt der HP41 einen großen Bruder?

Irgendwie bekannt kommt einem dieses Kästchen mit seinen vielen Tasten doch vor, was war es denn gleich . . .

. . . Natürlich, der HP41 war aber bei weitem schöner, die Tastatur erinnert mich an die neuerdings auch bei HP in Mode gekommenen Gummitastaturen, die zwar billiger in der Fertigung sind, ihren Hauptzweck aber, nämlich die gute Bedienbarkeit des Gerätes, in keinsten Weise mehr erfüllen, sie sind zum permanenten Ärgernis der Techniker geworden.

Die Kipptasten, die das erste Mal 1980 beim HP41 verwendet worden waren, sind und bleiben wohl eine Ausnahme an Ergonomie für Handhelds von HP, sie sind vermutlich wie alle guten Erfindungen von HP (z.B. das Plotterprinzip oder auch deren Tintenstrahl-druckprinzip) patentrechtlich geschützt.

Eine optische Verwandtschaft zum HP41 ist durchaus gewollt, das Gerät soll eigentlich ein Nachfolger für diesen werden, einige seiner Features zeigen dies auch recht deutlich, nicht nur fast dieselbe Anzahl und Anordnung von Tasten.

Ein Schwachpunkt des HP41, das Display, wurde, wie man sehen kann, deutlich verbessert, volle Graphikfähigkeit auf 64\*128 Punkten oder 8\*21 Zeichen.

Wenn man die Zeile für die Softkeys abzieht, eine für so komplexe Geräte sehr wichtige Eigenschaft, so bleibt immer noch genügend Platz für die Informationsdarstellung übrig, fast doppelt soviel wie beim HP18C oder HP28C.

ZM-Travel Service  
- Frau Alexa Wehmer -  
Hartmann Reisen GmbH  
Rolandstraße 9

D-4300 Essen 1

Absender:

Name \_\_\_\_\_

Vorname \_\_\_\_\_

Straße \_\_\_\_\_

PLZ/Ort \_\_\_\_\_

Telefon \_\_\_\_\_



So, nach all diesen theoretischen Überlegungen nun ein kurzer Steckbrief dieser kleinen Schatztruhe:

Prozessor:  
80C88

Betriebssystem:  
spezielles mit DOS-ähnlichen Befehlen, über Softkeys aufrufbar

Anzeige:  
8\*21 Zeilen, davon eine für die Softkeys, (64\*128 Punkte grafikfähig)

Speicher:  
128k RAM, erweiterbar auf 768k,  
128k ROM für das Betriebssystem und KERMIT

Hardware:  
Echtzeituhr mit Alarmfunktion eingebaut, Systemtakt 5 MHz,  
2 RS232 Schnittstellen an speziellen Stekkern (110-19200 Baud),  
4 Speichererweiterungsports (<> HP41 und HP71),  
1 Systemerweiterungsport

Software:  
Eigene Software kann in Einsteckmodulen gespeichert werden, diese kann auf einem IBM-PC entwickelt werden.

Das Betriebssystem kennt folgende Befehle:

CD Wechsel des Filesystems  
COPY Files kopieren  
DEL Files löschen

EXIT Verlassen des Kommandointerpreters und Ausschalten  
FORMAT RAM für Benutzung vorbereiten?  
RUN Programmstart  
VIEW Anzeige eines Files?

? heißt, daß ich nicht weis, was genau dahinter steckt.

Man sieht hier schon die Verwandtschaft zum DOS auf den IBMs, BATCH-Files sollen ebenfalls möglich sein.

Folgende Standartprogramme sind ebenfalls im ROM enthalten:

CHKDSK Massenspeichertest in Parameteranzeige  
DATE Anzeige/Änderung des Datums  
KERMIT Filetransfer-Utility zur Übertragung von Daten/Programmen über die RS232 Schnittstelle  
MODE Schnittstellenparameter setzen  
TIME Anzeige/Änderung der Zeit

KERMIT ist ein Programm, das es auf so gut wie jedem Rechner rund um den Erdball inklusive Großrechnern gibt, man kann mit seiner Hilfe Dateien von einem System zu einem anderen über die serielle Schnittstelle schicken. Dieses Programm wird zur Datenübertragung auf einen anderen Rechner verwendet, damit sollen auch z.B. HP41 Programme übertragen werden, dies ist aber anscheinend nicht so ganz einfach, so wie es in den bisherigen Veröffentlichungen zu sehen war:

Man muß dazu die Daten über die HP IL-Schnittstelle zu einem IBM-PC schicken,

der braucht natürlich auch eine IL-Karte. Dieser übersetzt dann die Programmdateien, die in ASCII vorliegen und schickt sie dem Marys II, der Weisheit letzter Schluß kann das wohl auch nicht sein!

Marys II kann mit oder ohne diese sogenannte HP41-Emulation gekauft werden, die sich dann im Betriebssystem-ROM befinden soll, über deren Brauchbarkeit kann man aber erst nach einem ausführlichen Test eine Aussage machen. Laut Beschreibung werden im Gegensatz zum HP41-Translator-PAC des HP71 alle Funktionen des HP41CX unterstützt, d.h. auch alle TIME und X-Function Funktionen, das wiederum hört sich gut an.

Ich hoffe, daß die Redaktion ein Gerät zum Test bekommt, die bisher vorliegenden Informationen sind doch zu stark von Werbung geprägt, ganz billig ist der Spaß ja auch nicht gerade, man muß schon weit über tausend Mark für diesen Rechner berappen.

Mit Fragen wendet Euch bitte an Stefan Schwall, der dieses Gerät hier in Deutschland vertreibt, seine Adresse steht unter Beirat im PRISMA ganz hinten. Alles, was ich bis jetzt geschrieben habe ergründet sich aus Werbeanzeigen und Berichten in Handheld Solutions VOL. I+II, einer Zeitschrift der CMT (Corvallis Microtechnology, Oregon).

Martin Meyer (1000)  
Redaktion

## Antwort zum offenen Brief in Prisma 6/87

Vielleicht vorab kurz etwas über meine Person: Ich bin unter den Clubadressen als Betreuer des Bereichs Vermessungswesen zu finden. Trotz meiner relativ hohen Mitgliedsnummer (2719) habe ich schon 1975 mit einem HP 45 angefangen. Über den HP 67 und HP 41 bin ich zu einem recht umfangreich ausgebauten HP 71B-System gekommen.

Herr Speckert hat in seinem Brief einiges zur Zukunft des Clubs niedergeschrieben. Ich möchte an dieser Stelle auch einiges sagen. In vielen Punkten hat Heinz Speckert Recht, insbesondere was die Firmenpolitik von Hewlett Packard betrifft. Ich selbst bin auf der Messe zweimal im Gedränge um den HP-Stand gelaufen, bis ich einen vereinsamen HP 28C entdecken konnte. Bei der Vielzahl von IL-Nutzern ein trauriges Zeugnis, das HP sich da ausgestellt hat. Inzwischen werden nämlich auch viele Vermessungsgeräte über Interface und IL-Modul am HP 71B betrieben. Auch ich hoffe, daß der HP 71 mit mehr Grundspeicherkapazität und größerem Display eine Weiterentwicklung findet.

Die Probleme, die der Club momentan mit den Mitgliederzahlen hat, sind sicherlich auch auf die Entwicklung auf dem Computermarkt zurückzuführen. Wenn ich alle Geräte meiner IL-Loop und die Speichererweiterungen einmal vom Preis her aufaddiere, wird mir schon noch leicht schwindelig.

Bei einem Preisvergleich mit Geräten wie Schneider, Commodore oder Atari wird natürlich so mancher schwach. Bei mir fallen bisher immer noch das HP 71B-Basic und die geringen Abmessungen bei Vergleichen stark ins Gewicht. Die geringen Abmessungen und der Batteriebetrieb spielen im Außendienst eine ganz besondere Rolle.

Die Gestaltung von Prisma hat sich meiner Meinung nach zum Jahreswechsel 87/88 recht gut entwickelt und die Verzögerungen sind zu entschuldigen, da ja auch alles auf freiwilliger Basis gemacht wird. Es ist nur auch sehr schwer, für alle Nutzer etwas anzubieten. Ich habe bei der Führung der Programmsammlung Vermessungswesen dabei auch einige Erfahrungen gemacht. Bis auf wenige Ausnahmen wollen alle etwas haben, aber kaum jemand liefert gut dokumentierte Programme. Die Weitergabe von Programmen ist auch nicht immer ganz einfach. Man glaubt gar nicht, wieviele Kartenleser es am HP 41 noch gibt. Zum Schluß war ich teilweise soweit, daß ich mit dem in Prisma vorgestellten Programm LIST 41 Listings erstellt und verschickt habe. Vorher mußte aber noch der kleine „Bug“ (siehe Prisma 6/87) beseitigt werden. Die Führung der Sammlung soll aber aufrecht erhalten werden.

Eines habe ich aber bei der Betreuung des Fachbereichs Vermessung gelernt: Man

darf PRISMA nicht als einzige Informationsquelle ansehen.

Kontakte, die direkt mit den Artikelschreibern geknüpft werden, sind meist viel ergiebiger. Hier läuft der Austausch von Programmen recht gut, wenn man erst einmal jemand mit etwa gleicher Geräteausstattung gefunden hat. Ein großes Problem ist aber die Dokumentation von den Programmen. Auch Lexfiledokumentation fehlen sehr oft. Ich habe vor, in dieser Richtung einiges zu machen, um Mitgliedern auf Anfrage mitzuteilen, was es überhaupt gibt. Sicherlich gibt es im CCD HP 71 Lexfilespezialisten, die einiges auf die Beine stellen können. Ich selbst würde mich als „guten Anfänger“ einstufen und bin auch bereit Mitgliedern im Lexfilesektor zu helfen. Vielleicht wird aber auch etwas Zurückhaltung geübt, weil ein wirtschaftliches Interesse eine Rolle spielt. Außerdem dürfte auch das Urheberrecht dazu führen, daß bei Lexfiles nicht so viel Gutes veröffentlicht wird.

Ich hatte in der Prisma auch einmal etwas von einem geplanten EPROM mit Lexfiles gelesen. Davon hat man aber nichts mehr gehört. Aus dieser Ecke hätte doch eigentlich ein Verzeichnis kommen müssen, was an Lexfiles vorhanden ist. Konsequenz: Verzeichnis selbermachen oder man versinkt im Lexfile-Befehlssalat. Und dann wären da noch meine Frau und drei Kinder und mein Brötchengeber . . . .

Ulrich Kulle (2719)

# Technisch-wissenschaftliche Textverarbeitung auf Personal Computern

## 1. Einleitung

Während für die Bearbeitung „herkömmlicher“ Texte auf IBM-PC's und Kompatiblen inzwischen eine kaum noch übersehbare Fülle von Programmen angeboten wird, waren Programme zur Erstellung technisch-wissenschaftlicher Texte bisher Mangelware. Diese Art von Texten, die viele Sonderzeichen wie griechische Buchstaben und mathematische Symbole, Brüche, mehrstufige Indizes und Exponenten u.ä. enthalten, stellen ganz andere Anforderungen an Hard- und Software als „normale“ Texte.

Seit einiger Zeit sind jedoch Programme erhältlich, die speziell auf diese Anforderungen zugeschnitten sind. Vier solche Systeme, nämlich ScienTEX, ChiWriter, T<sup>3</sup> und PC-TEX, wurden von den Autoren im Rahmen ihrer Tätigkeit beim Microcomputer Beratungs- und Informationsteam (Micro-BIT) des Rechenzentrums einer eingehenden Betrachtung unterzogen.

## 2. WYSIWYG-Systeme

Die ersten drei der genannten Programme, also ScienTEX, Chiwriter und T<sup>3</sup> arbeiten nach dem What-You-See-Is-What-You-Get-Prinzip (WYSIWYG-Prinzip), d.h. alle Formeln und Sonderzeichen erscheinen bei der Bearbeitung eines Textes so auf dem Bildschirm, wie sie nach dem Druck auch auf dem Papier zu sehen sind.

Es handelt sich bei diesen Systemen um vollständige Textverarbeitungsprogramme, die alle Bearbeitungsschritte der Editierung und Formatierung technisch-wissenschaftlicher Texte ermöglichen. Hierzu verfügen sie sowohl über alle wichtigen Editierfunktionen wie Suchen und Ersetzen von Zeichenfolgen, Verschieben und Kopieren von Textblöcken u.ä., als auch über Formatierungsfunktionen wie Zeilen- und Seitenumbruch, Absatzformatierung und dergleichen mehr. Außerdem bieten sie die Möglichkeit, Zeichen aus mehreren verschiedenen Zeichensätzen zu verwenden. Zur Standardausstattung gehören ein Satz kursiver und ein Satz griechischer Buchstaben sowie ein oder zwei Sätze mathematischer Symbole, bei ScienTEX und T<sup>3</sup> auch ein Satz zum Aufbau chemischer Strukturformeln.

Darüber hinaus können mit Hilfe mitgelieferter Editoren die vorhandenen Zeichen beliebig verändert oder auch ganz neue Zeichen erstellt werden. Während es hierbei nur kleine Unterschiede zwischen den drei Programmen gibt, unterscheiden sie sich erheblich in den Benutzeroberflächen sowie der Art und Weise, in der mathematische Formeln eingegeben und editiert werden können. Auf diese beiden Punkte wird daher im weiteren besonders eingegangen.

### 2.1 ScienTEX

#### a) Allgemeines

ScienTEX benötigt als Mindest-Hardwarekonfiguration einen IBM-PC oder Kompat-

blen mit 2 Diskettenlaufwerken, 256 kByte Hauptspeicher und einer Farbgrafik- oder Hercules-Karte und kann mit dieser Konfiguration problemlos betrieben werden.

Man kann ScienTEX in drei Versionen installieren, einer mathematischen, einer chemischen und einer sogenannten multilingualen Version. In jeder Version stehen insgesamt 5 Zeichensätze zu je 95 Zeichen zur Verfügung, wobei allen Versionen jeweils ein normaler, ein kursiver und ein griechischer Zeichensatz gemeinsam ist.

Das Programm ist kopiergeschützt durch einen Zwischenstecker, den man dem Drucker kabel vorsteckt. Ohne diesen Stecker ist eine Speicherung von Texten nicht möglich.

#### b) Benutzeroberfläche

Die verschiedenen Funktionen von ScienTEX werden über die Funktionstasten bzw. Kombinationen der Funktionstasten mit der Control- oder Alt-Taste ausgelöst, wobei zwischen einem Editor- und einem Funktionsmodus unterschieden wird.

Im Editiermodus dienen die Funktionstasten zur Auswahl des zu benutzenden Zeichensatzes, zum Setzen und Löschen der Randmarkierungen sowie der Vergabe von Attributen wie Fettdruck, Unterstreichen etc.

Im Funktionsmodus können weitergehende Funktionen zur Formatierung von Texten, Blockbearbeitung, Drucken, Dateiverwaltung usw. aufgerufen werden. Dabei ist die Tastenbelegung zum Teil mehrstufig ausgeführt, d.h. nach Auswahl einer Funktion wechselt die Belegung, um die Eingabe eventueller Optionen der gewählten Funktion zu ermöglichen. Obwohl die jeweilige Funktionstastenbelegung am Bildschirm angezeigt wird, ist die Bedienung auf diese Art doch recht umständlich und nur schwer zu merken. Dies gilt vor allem für die Tastenkombinationen des Editiermodus (z.B. Ctrl-F1 zum Setzen eines Tabulators) und der Cursorsteuerung (z.B. Ctrl-PgDn zum Springen zum nächsten Wort), hinter deren Wirkung keine erkennbare Systematik steckt.

#### c) Formeleditierung

Zur Eingabe von Formeln muß zunächst in einen speziellen Formelmodus umgeschaltet werden. Dann markiert man mit Hilfe des Cursors auf dem Bildschirm ein Feld, das groß genug für die zu erstellende Formel ist. Dieses Feld läßt sich auch im nachhinein noch erweitern. Innerhalb des Feldes kann man den Cursor nun halbzeilenweise auf- und abwärtsbewegen und auf diese Weise z.B. beliebig komplexe Indizes und Exponenten aufbauen.

Die fertige Formel besteht dann aus einer Anzahl voneinander unabhängiger Zeilen im Halbzeilenabstand. Das hat zur Folge, daß man innerhalb einer solchen Zeile, also beispielsweise einer Indexstufe, Verschiebungen vornehmen kann, ohne daß die anderen Zeichen davon betroffen werden, oder daß man eine solche Zeile problemlos

zwischen bereits bestehende einfügen kann. Es bedeutet aber auch, daß die Formel nicht als Ganzes horizontal verschoben werden kann, sondern jede Zeile, die zur Formel gehört, einzeln bewegt werden muß.

#### d) Sonstiges

ScienTEX verfügt im Gegensatz zu den beiden anderen Programmen über die Möglichkeit des Hintergrunddrucks, d.h. man kann während des Druckens mit der Textbearbeitung fortfahren. Außerdem bietet ScienTEX eine Mail-Merge-ähnliche Einrichtung, mit deren Hilfe man zwei Texte während des Drucks miteinander mischen und so beispielsweise Serienbriefe drucken kann. Allerdings müssen mit dieser Funktion auch Kopf- und Fußzeilen sowie Fußnoten realisiert werden, was fehleranfällig und vor allem sehr umständlich ist.

## 2.2 ChiWriter

### a) Allgemeines

Voraussetzung zum Betrieb von ChiWriter ist ein IBM-PC oder Kompatibler mit 2 Diskettenlaufwerken, 256 kByte Hauptspeicher und einer Farbgrafik-Karte. Mit dieser Konfiguration kann ohne Schwierigkeiten gearbeitet werden, allerdings empfiehlt sich die Anschaffung eines Treibers für die Hercules oder EGA-Karte, deren höhere Bildschirmauflösung die Augen bei längerem Arbeiten schont.

Es stehen 14 Zeichensätze à 95 Zeichen zur Verfügung, die Maximalzahl der verwendbaren Zeichensätze ist 20. Das Programm ist nicht kopiergeschützt.

### b) Benutzeroberfläche

Die Steuerbefehle von ChiWriter lassen sich in drei Gruppen einteilen, nämlich in solche, die über die Funktionstasten ausgewählt werden, solche, die durch Kombinationen der Control- und einer Buchstabentaste ausgelöst werden, und solche, die über ein Menü am unteren Bildschirmrand aufgerufen werden.

Die Funktionstasten dienen ausschließlich zur Auswahl des zu benutzenden Zeichensatzes. Mit Hilfe der Control-Kombinationen werden die direkt bei der Texteingabe und -formatierung benötigten Funktionen aufgerufen, wobei das einzugebende Zeichen in mnemonischem Zusammenhang mit dem jeweiligen Befehl steht (z.B. Ctrl-C zum Zentrieren einer Zeile, Ctrl-[ zum Setzen des linken Randes etc.).

Über das Menü schließlich können weitergehende Operationen z.B. zur Blockbearbeitung, zum Drucken, zur Dateiverwaltung usw. ausgeführt werden. Hierzu aktiviert man das Menü zunächst mit der Escape-Taste und löst dann durch Eingabe des Anfangsbuchstabens des jeweiligen Menüpunktes die entsprechende Funktion aus. Die Menüpunkte können aber auch mit der

Alt-Taste direkt aufgerufen werden ohne vorher auf das Menü umzuschalten. Auf diese Art und Weise läßt sich ChiWriter schnell, bequem und ohne großen Lernaufwand bedienen.

**c) Formeleditierung**

Im Gegensatz zu ScienTEX kennt ChiWriter keinen besonderen Formelmodus, sondern es können Formeln direkt in den laufenden Text eingegeben werden. Bei ChiWriter besteht eine Zeile aus einer Hauptebene sowie einer unbegrenzten Anzahl von Index- und Exponentenebenen im Halbzeilenabstand. Mit Hilfe der Tastenkombinationen Ctrl-PgUp und Ctrl-PgDn kann der Cursor jederzeit auf diesen Ebenen auf- bzw. abwärts bewegt werden, wobei automatisch eine neue Ebene hinzugefügt wird, wenn man sich in der obersten oder untersten befindet. Zwischen zwei Ebenen einer Zeile kann man mit Ctrl-A (für "add") eine neue einfügen und mit Ctrl-Z (für "zap") eine Ebene löschen.

Wichtig ist hierbei, daß eine so aufgebaute Formel in einer einzigen Zeile steht und beim Einfügen oder Löschen von Zeichen unmittelbar vor der Formel als Einheit behandelt wird, d.h. sämtliche zugehörigen Index- und Exponentenebenen werden mitverschoben. Dies wird dadurch erreicht, daß alle Spalten einer Zeile synchron bewegt werden (bei  $\tau_2^1$  z.B. befinden sich der Exponent 2 und der Index 1 in einer Spalte). Das hat aber auch zur Folge, daß beispielsweise eine solche Spalte immer als ganzes gelöscht wird, selbst wenn nur ein Zeichen auf irgendeiner Ebene innerhalb dieser Spalte gelöscht werden soll.

Effekte dieser Art können durch einen sogenannten "unsynchronisierten" Modus vermieden werden. In diesem Modus wirken sich alle Operationen wie Löschen, Einfügen und insbesondere auch die Blockbearbeitung nur in der (den) betroffenen Ebene(n) aus. Während man im synchronisierten Modus nur alle Zeilenebenen umfassende Spaltenfolgen als Block definieren kann, ist es im unsynchronisierten Modus möglich, ein beliebiges Rechteck innerhalb der Zeilenebenen zu markieren und weiterzuverarbeiten und so z. B. den Nenner eines Bruches in eine andere Formel hineinzukopieren.

**d) Sonstiges**

Eine weitere interessante Eigenschaft von ChiWriter ist die Möglichkeit Makros, sogenannte "Key Sequences", anzulegen. In einer solchen "Key Sequence" läßt sich eine beliebige Folge von Tastendrücken abspeichern und jederzeit wieder aufrufen. Diese Funktion ist überaus nützlich um z.B. komplexe Ausdrücke, die oft im Text vorkommen, nicht jedesmal neu eingeben zu müssen oder auch um häufige Bedienungsvorgänge zu automatisieren.

Als Nachteile von ChiWriter wären in erster Linie die fehlende Trennfunktion sowie die nach Formatänderungen notwendige manuelle Neuformatierung der betroffenen Absätze zu nennen. Außerdem führt die Verwendung einer deutschen Tastaturbelegung zu Problemen bei der Systembedienung, weil dadurch bestimmte Tastenkombinationen (z.B. Ctrl-]) nicht mehr ohne weiteres ausgelöst werden können. Über diese Nachteile tröstet jedoch der günstige Preis des Programmes hinweg: die Grundversion mit Treibern für die Farbgrafik-Karte und einige 9-Nadel-Drucker kostet 80 US\$.

**2.3 T<sup>3</sup>**

**a) Allgemeines**

Für den Betrieb von T<sup>3</sup> benötigt man einen IBM-PC oder Kompatiblen mit 2 Diskettenlaufwerken, 512 kByte Hauptspeicher und einer Grafikkarte (Farbgrafik, Hercules, EGA, Olivetti, HP Vectra u.a.). Diese Mindestkonfiguration ist jedoch nicht sinnvoll, schon allein wegen der Größe des Systems (ungefähr 1,5 MByte) empfiehlt sich eine Festplatte, und als Hauptspeicher sollten 640 kByte vorhanden sein.

Zum Lieferumfang gehören 9 Zeichensätze zu je 128 Zeichen. In einem Text dürfen maximal 8 verschiedene Zeichensätze verwendet werden, die Gesamtzahl der Zeichensätze im System ist nicht begrenzt. Das Programm ist durch eine Key-Diskette, die beim Start überprüft wird, kopiergeschützt.

**b) Benutzeroberfläche**

Die Hauptbedienungselemente von T<sup>3</sup> sind Pop-Up-Menüs, sogenannte Formulare sowie die Funktionstasten.

In den Menüs wird ausgewählt, indem man einen Inversbalken mit Hilfe der Cursor-tasten bewegt und die Auswahl mit der Plus-Taste des Ziffernblocks bestätigt.

Die Formulare enthalten Felder, die der Benutzer ausfüllen muß, z.B. mit dem Namen eines Textes. Durch Drücken der Funktionstaste F9 (der sog. "Menü-Taste") in einem solchen Feld wird ein Menü mit den möglichen Einträgen für dieses Feld aufgerufen. Die einzelnen Bedienungsschritte folgen dabei einem im ganzen Programm konsequent durchgehaltenen Schema: immer wird zuerst ein Objekt ausgewählt (T<sup>3</sup>-Objekte sind z.B. Texte, Tastaturbelegungen, Zeichensätze, Formate usw.) und dann die Operation, die mit diesem Objekt durchgeführt werden soll (z.B. Bearbeiten, Kopieren, Löschen usw.). Dadurch ist die Bedienung von T<sup>3</sup> trotz des großen Umfangs des Programmes und seiner vielen Funktionen leicht erlernbar. Allerdings fällt die Umgewöhnung von einem anderen Text-

system auf T<sup>3</sup> nicht ganz leicht, weil man sich mit einigen neuen, ungewöhnlichen Konzepten vertraut machen muß. Doch gerade diese Konzepte, wie z.B. das genannte Auswahlprinzip, machen T<sup>3</sup> zu einem Beispiel für die benutzerfreundliche Gestaltung eines Programmes.

**c) Formeleditierung**

Die Formeleditierung mit T<sup>3</sup> ist derjenigen mit ChiWriter sehr ähnlich. Auch bei T<sup>3</sup> ist eine Zeile aus mehreren Ebenen aufgebaut, in denen man sich mit Hilfe des PgUp- und der PgDn-Taste halbzeilenweise bewegen kann. Die Anzahl der Ebenen ist auf insgesamt 25, 12 Index-, 1 Haupt- und 12 Exponentenebenen beschränkt, was für die meisten Zwecke aber mehr als ausreichend ist.

Die Spalten einer Zeile sind synchronisiert, d.h. beim Einfügen und Löschen wird eine Formel als Einheit behandelt, eine Spalte ist von den genannten Operationen immer als ganzes betroffen. Einen unsynchronisierten Modus wie ChiWriter kennt T<sup>3</sup> allerdings nicht, so daß innerhalb einer Ebene keine Verschiebungen möglich sind.

**d) Sonstiges**

T<sup>3</sup> verfügt wie ChiWriter über die Möglichkeit, Makros anzulegen. Diese heißen auch bei T<sup>3</sup> "Key Sequences" und funktionieren genau wie bei ChiWriter. Außerdem stehen dem Benutzer bei der Bearbeitung eines Textes zwei Tastaturbelegungen zur Verfügung, die er sich völlig frei zusammenstellen kann. Auf diese Weise lassen sich ohne Schwierigkeiten bestimmte nationale Tastaturbelegungen erzeugen, was das System auch für die Erstellung fremdsprachlicher Texte interessant macht.

Schließlich bietet T<sup>3</sup> noch eine (halbautomatische) Trennfunktion, sowie die Möglichkeit, einmal definierte Absatz- und Seitenformate zu speichern und in anderen Texten wiederzuverwenden. Damit kann man seinen Texten leicht ein einheitliches Äußeres geben, und es lassen sich so auch bestimmte Klassen von Texten mit unterschiedlichen Formaten bilden, z.B. Briefe, Berichte, Kurzmitteilungen etc.

**2.4 Zusammenfassung**

Von den betrachteten Programmen ist ChiWriter auf dem Gebiet der Formeleditierung eindeutig das beste. Es erlaubt dem Benutzer, sowohl eine Formel als Einheit zu

	ScienTex	T <sup>3</sup>	ChiWriter
Vorteile	Hintergrunddruck Serienbriefe	hervorragende Benutzeroberfläche vielfältige Formatierungsmöglichkeiten Trennhilfe ausführliche Dokumentation	niedriger Preis sehr gute Formelbearbeitung einfache Bedienung
Nachteile	komplizierte Bedienung Formelbearbeitung umständlich keine Trennhilfe unübersichtliches Handbuch teuer	teuer	umständliche Änderung von Formaten keine Trennhilfe

behandeln (was ScienTEX nicht kann), als auch innerhalb einer Formel Verschiebungen vorzunehmen (was mit T<sup>3</sup> nicht möglich ist). Durch seine einfache Handhabung, den ausreichenden Funktionsumfang und seinen erstaunlich niedrigen Preis ist es in der Lage, ein Programm wie beispielsweise WordStar vollauf zu ersetzen und bietet darüber hinaus noch die Fähigkeit, Formeln zu schreiben.

T<sup>3</sup> überzeugt durch seine beispielhafte Benutzeroberfläche, die übersichtliche und ausführliche Dokumentation sowie den vielfältigen Formatierungsfunktionen und der Möglichkeit einmal festgelegte Formate wiederzuverwenden.

ScienTEX schließlich fällt in diesem Vergleich etwas ab. Zwar verfügt es als einziges der untersuchten Systeme über die Fähigkeit des Hintergrunddrucks, doch diese Eigenschaft kann die schwerfällige Bedienung, die Mängel bei der Formeleditierung sowie die fehlende Makrofunktion nicht ausgleichen.

**3. Reine Formatier / Textsatzsysteme**

**3.1 T<sup>3</sup>EX**

**a) Allgemeines**

Bei der Besprechung von Textverarbeitungssystemen für technisch wissenschaftliche Anwendungen sollte das Ende der siebziger Jahre von Donald E. Knuth an der Stanford University entwickelte T<sup>3</sup>EX nicht fehlen. Bekannt geworden auf Großrechnern, ist es seit einigen Jahren auch für IBM-PC's und Kompatible verfügbar. Bevor wir auf einige Einzelheiten eingehen, ein grundlegender Unterschied: T<sup>3</sup>EX ist ein reines Formatierungssystem (Textsatzsystem). Zum Editieren ist ein Editor notwendig (z.B. Wordstar, WordPerfect, SPFFPC . . .).

Minimalkonfiguration: PC/XT oder AT (bzw. Kompatibler) mit 10 MB Platte und 360 k Floppylaufwerk. Arbeitsspeicher mindestens 512 kB für T<sup>3</sup>EX, 640 kB für L<sup>A</sup>T<sup>3</sup>EX.

**b) Benutzeroberfläche**

Eine Belegung der Funktionstasten wie bei den bisher besprochenen Systemen gibt es bei T<sup>3</sup>EX nicht. Befehle an den Formatierer werden als Kommandos beginnend mit einem "/" in den Text geschrieben. Es kann durchaus sein, daß ein Text mit anspruchsvollen Layout auf den ersten 1-2 Seiten (Eingabetext) nur aus T<sup>3</sup>EX Kommandos besteht. Ein WYSIWYG gibt es nicht (käuflich erwerbbar sind allerdings "Screen Previews" zum Ansehen von formatierten Texten). Zum Erstellen qualitativ hochwertiger Dokumente sollte eine Einarbeitungszeit von ca. 2 Wochen eingeplant werden.

**c) Formeleditierung**

Dieses funktioniert bei T<sup>3</sup>EX durch das Umschalten in den Math-Modus durch "\$", wenn der Ausdruck im Text eingebunden ist, oder durch "\$\$", wenn der Ausdruck allein in einer Zeile stehen soll. Der Ausdruck wird geschrieben aus T<sup>3</sup>EX-Befehlen und Zeichen. "\$" oder "\$\$" beenden den Math-Modus auch wieder.

Die Größe bzw. Länge von mathematischen Zeichen wird automatisch der Syntax angepaßt. Zwischenräume zwischen Text und Formel werden automatisch richtig generiert. Dazu liegen T<sup>3</sup>EX Regeln zugrunde, die den typographischen Ansprüchen der Herausgeber von mathematisch-wissenschaftlichen Fachliteratur genügen (einige Verlage akzeptieren bzw. wünschen T<sup>3</sup>EX-geschriebene Texte als Vorlage).

Beispiel:

$$\sum_{n=0}^{\infty} a_n z^n \quad \text{für} \quad |z| < \left( \limsup_{n \rightarrow \infty} \sqrt[n]{|a_n|} \right)^{-1}$$

wurde erzeugt durch:

```


$$\sum_{n=0}^{\infty} a_n z^n$$


$$\left| z \right| < \left( \limsup_{n \rightarrow \infty} \sqrt[n]{|a_n|} \right)^{-1}$$


```

**d) Sonstiges**

1) T<sup>3</sup>EX zeichnet sich durch die Implementierung von Algorithmen aus, die versuchen Abschnitte des Textes möglichst optimal zu gestalten. Dabei wird das Verhältnis zwischen Trennungen und Zwischenräumen zwischen Wörtern optimiert. Trennungen werden von T<sup>3</sup>EX selbst durchgeführt (momentan amerikanischer Trennungssatz, deutscher ist in Vorbereitung, wobei beim amerikanischen ca. 80 Prozent richtig getrennt wird). Kann T<sup>3</sup>EX Worte nicht trennen, so gibt man Trennvorschläge ein.

2) Durch das Schreiben von Makros in T<sup>3</sup>EX und/oder durch Verwendung der mitgelieferten Makroerweiterungen (plain, AMS-T<sup>3</sup>EX oder L<sup>A</sup>T<sup>3</sup>EX) kann z.B. folgendes mit T<sup>3</sup>EX gemacht werden:

- Text in Zwei- oder Mehrspaltenformat
- Indexliste erzeugen
- Seitenzahl oben / unten / rechts / mitte / links
- Automatische Kapitelnumerierung
- Kopfzeilen, Fußnoten . . .

3) Im Gegensatz zu den bisher besprochenen Systemen, gibt es T<sup>3</sup>EX auf nahezu allen Rechnern und Betriebssystemen (Beispiele: APPLE MacIntosh, PC unter DOS, . . . , VAX unter VMS, Siemens 7881 unter BS3000, . . .). Ein Austausch der Texte ist auf mehreren Stufen möglich:

T<sup>3</sup>EX-Eingabetext und DVI-File (DVI = DeVice Independent, eine Datei, die Informationen darüber enthält, an welcher Stelle einer Seite welches Zeichen aus welchem Font plaziert wird). Diese Möglichkeit erlaubt es auch andere Drucker, als die am PC angeschlossenen, zu nutzen. Bestehen die Übertragungsmöglichkeiten, so ist auch das Verschicken von fertigen Druckdateien möglich.

4) Druckertreiber gibt es für nahezu alle T<sup>3</sup>EX-fähigen (Graphikfähigkeit und/oder ladbare Zeichensätze) Drucker (Beispiel: EPSON, HP-Laserjet . . .).

**4. Zusammenfassung**

Für jemanden, der ab und zu einen Text schreibt, der auch "Formeln" enthält (für den die Einarbeitungszeit in T<sup>3</sup>EX also in keinem Verhältnis zum Nutzen steht), oder

Eigenschaft	WYSIWYG	T <sup>3</sup> EX
Editorfunktionen	ja	keine, da reines Formatierungssystem
Lernaufwand	gering	hoch
Dokumentation	T <sup>3</sup> - sehr gut ChiWriter - gut ScienTex - verwirrend	T <sup>3</sup> EX-Book - schwer verständlich, deutsche Einführungen gut
"On-line Hilfe"	gut	kaum, dafür eindeutige Fehlermeldungen
Dokumentenerstellung	nach eigenen Geschmack, "Augenmaß", keine Hilfe-stellung des Programms	strukturierte Dokumentenerstellung möglich
Formatiereigenschaften	kaum	sehr gut, anerkannt von renommierten Verlagen
Kopf- und Fußzeilen	ja	ja
Inhaltsverzeichnis	von Hand	automatisch
Schlagwortverzeichnis	von Hand	automatisch
Layout von Formeln	gut	sehr gut
Korrekturen in Formeln	einfach - problematisch je nach Programm	einfach
Trennung	Programme geben z.T. Trennhilfen. Trennung manuell	automatisch (95% richtig)
Verfügbarkeit	nur auf IBM-PC's und Kompatible	auf fast allen Rechnern - übertragbare Texte

jemand der Wert auf eine bedienungsfreundliche Benutzeroberfläche legt und mit den Möglichkeiten von T<sup>3</sup>, ChiWriter oder Scien-TEX zufrieden ist, dem seien diese Pakete empfohlen. Wer aber auf die Vorteile eines Textsatzsystemes (wie es auf PC's schon einige gibt, die leider nicht für Formeln u.a. geeignet sind) nicht verzichten will, dem bleibt momentan nur das Arbeiten mit den "benutzerunfreundlichen" TEX.

Ein großer Vorteil ist neben den oben angeführten Fähigkeiten auch die Kompatibilität von TEX auf dem PC und auf anderen Rechnern, während die "WYSIWYG"-Programme im Allgemeinen auf keiner anderen Anlage verfügbar sind.

5. Literatur

[1] Knuth, D.E.: The TEXbook, Addison-Wesley publishing Company, 1984.  
 [2] Spivak, M.: PC TEX Manual, 1985  
 [3] Wagner, J.: Textverarbeitung an einem wissenschaftlichen Rechenzentrum, Rechenzentrum der Universität Karlsruhe, 1985  
 [4] Heinz, A., Meyer, K.: DocEdit – Ein Editor zur Syntaxgesteuerten Erstellung strukturierter Dokumente, bisher unveröffentlicht, Universität Karlsruhe, Institut für Angewandte Informatik, 1987.  
 [5] T<sup>3</sup> Manual  
 [6] ChiWriter Manual

Michael Helmich  
 Rainer Rupprecht  
 Rechenzentrum Universität Karlsruhe  
 Postfach 6980  
 7500 Karlsruhe 1  
 West-Germany

# HP-Serie 80 Story

## Die Geschichte der HP-Serie 80

Liebe CCD'ler,

ich möchte Euch mit einem kleinen Beitrag über die HP-Serie 80 etwas über ein Computer- und Mikroprozessorsystem erzählen, welches zwar nie den Durchbruch auf dem breiten internationalen Markt geschafft hat, aber auf dem Nischenmarkt der Meßgerätesteuerung und Datenauswertung Computergeschichte geschrieben hat, wie meines Wissens nach kein anderes System. Ich fühle mich zwar kompetent genug, über die Serie 80 etwas zu schreiben, aber da auch ich nicht „Zwerg Allwissend“ bin, können natürlich einige Angaben ungenau und manche Vermutungen falsch sein. Schließlich dürfte wohl allen bekannt sein, daß man von HP nur selten die Information bekommt, die man braucht (sehr miserabler Service, den die Händler auch noch teuer bezahlen müssen).

Ihr könnt die folgende Story auch als Nachruf betrachten, denn im November 1987 hat HP die Produktion der Serie 80 eingestellt, und damit von der Fachwelt nahezu unbemerkt eine nicht mehr zu schließende Lücke in den genannten Anwendungsbe- reich gerissen.

Die Entwicklung der Serie 80 begann im Jahr 1975, wobei mir allerdings unbekannt ist, ob der in den 80zigern verwendete Mikroprozessor ohne Namen damals bereits zur Verfügung stand oder ob er erst im Laufe der Systementwicklung fertigungsreif wurde. Ich gehe zuerst auf die von vielen Assemblerprogrammierern als dubios verkannte CPU ein.

Die technischen Daten der CPU sind folgende:

- 8 Bit Datenbreite
- 16 Bit adressierbarer Speicherbereich
- Registerorientiert mit insgesamt
  - 16 Registern a 2 Byte
  - 4 Registern a 8 Byte
- Taktfrequenz 625 kHz
- 254 Befehle, davon
  - 128 Befehle um die zu verwendenden Register zu adressieren
  - 128 Datenmanipulationsbefehle
- BCD oder Binäremodus (Die CPU kann alle arithmetischen Anweisungen mit vier Bit Breite – selbst

- bei 8-Byte-Registern – mit allen benötigten BCD-Überträgen korrekt ausführen)
- 8 Statusflaggen
- 1 4-Bit-Register (entspricht dem CARRY-Flag bei BCD-Operationen)
- Befehlsausführungszeit zwischen 1 und 15 Taktzyklen

Die technischen Daten an sich sind schon recht beeindruckend (abgesehen vom Takt), was aber noch viel erstaunlicher ist, sieht man erst, wenn man sich anschaut, was die Befehle mit den Registern alles machen können. Man kann sich den Registersatz als 64 direkt nebeneinander liegende 1-Byte-Register vorstellen, wobei jedes Register als Ziel oder als Startadresse einer Operation verwendet werden kann. Die CPU baut nun logische Grenzen zwischen den Registern auf. Und zwar sind die ersten 32 Register zu 2er paaren zusammengefaßt, und die folgenden 32 Register zu paaren a 8. Wenn ein arithmetischer Befehl ausgeführt wird, bei dem die Zieladresse gleich der Startadresse ist (z.B. Increment) so arbeitet der Befehl ab dem momentan aktiven Register bis zur nächsten Registergrenze, und die kann wie besagt 1 bis acht Byte entfernt sein. HP bezeichnet den Zeiger, der die Registerstartadresse definiert als DRP (Data-Register-Pointer), wo dieser Name herkommt, ist mir allerdings reichlich unklar. Werden nun mehrere Register zusammenaddiert oder der Inhalt eines Registersatzes wird in andere Register verschoben, so muß der CPU noch das erste Zielregister angegeben werden. Bei der Abarbeitung eines solchen Mehrregisterbefehles geht die CPU nun so vor, daß sie nur die nächste Registergrenze des Startregisterzeigers beachtet, die Grenzen der Zielregister dagegen werden einfach ignoriert. Man kann also durchaus ein acht Byte langes Wort auf vier aufeinanderfolgende 2-Byte-Register verteilen, und das, man lese und staune, mit einem einzigen Befehl. Da nun alle CPU-Befehle mit jeder beliebigen Registerkombination arbeiten, ist die CPU mit ihren mageren 128 datenverändernden Befehlen trotzdem ein Superchip. Betrachtet man zum Beispiel die Befehle zur Stapelbearbeitung, so stellt man fest, daß man nicht nur theoretisch sondern auch praktisch insgesamt 32 Stapel gleichzeitig bearbeiten kann, ohne Registerinhalte in den Speicher verlagern zu müssen, denn

jeder Stapel hat eine Breite von 2 Byte (der Sinn einer solchen Anwendung sei dahingestellt). In Berlin ist ein guter Freund von mir dabei, sich mit der mageren Registeranzahl der 80C86-CPU von Intel herumzuschlagen, vorher programmierte er die Serie-80-CPU in Assembler. Es ist interessant, von ihm zu hören, wo er seine Probleme hat. Die einfache Struktur der Befehle (es gibt weder Multiplikation, noch Division, speicherverschiebende Befehle und dergleichen), sowie die kurzen Ausführungszeiten, ein Takt-Zyklus für alle 1-Byte-Befehle, lassen darauf schließen, daß die CPU wohl nicht mikroprogrammiert ist (siehe PRISMA 6/87), sondern sämtliche Funktionen fest verdrahtet enthält. Wenn man dieses Konzept etwas näher betrachtet, so stellt man fest, daß die RISC-Architektur wohl doch nicht so neuen Datums ist. Ein Entwickler von C-Compilern würde an Betracht dieser CPU wahrscheinlich feuchte Augen bekommen, denn dieses Glanzstück wurde der Welt nur in Form eines Computers zur Verfügung gestellt, sie war nie einzeln erhältlich.

Allerdings war die Struktur dieses Computersystems derart überragend, daß man es wohl nur dem hohen Preis und einer eigenartigen Vertriebspolitik zu verdanken hat, daß der Welterfolg sich nicht einstellte.

Die HP-Entwicklungsingenieure haben um diese CPU herum ein Computersystem aufgebaut, welches meiner Ansicht nach selbst heute noch – auf vergleichbaren Sektoren – seinesgleichen sucht. Unter strikter Ausnutzung der Registerorientierung wurde das gesamte BASIC-Betriebssystem des Computers entwickelt. So werden zum Beispiel die Gleitkommazahlen bei den meisten Berechnungen (Addition, Multiplikation, Umwandlung etc.) komplett innerhalb der CPU manipuliert. Der Computerspeicher wird also nur zur Ablage von Programmen und Programmdateien benötigt, abgesehen vom Operationsstapel auf dem die Zwischenergebnisse der BASIC-Ausdrücke gelagert werden. Aus diesem Programmkonzept resultiert die für die niedrige Taktfrequenz sehr hohe Rechengeschwindigkeit (ein HP-200 mit dem 68000-Prozessor von Motorola ist im Schnitt nur 10 bis maximal 20 Mal so schnell, obwohl er mit mehr als dem 10-fachen Takt gefahren wird, begrenzte mathematische Koprozessor-Funktionen auf seinem Chip vereinigt und ein ein 16/32-Bit Prozessor ist).

Da das BASIC-Betriebssystem der Serie-80-Rechner unvergleichbar groß ist (ein ausgebaute 86er hat ein über 100 kB großes Betriebssystem), hat HP schon damals (1975) auf eine sehr elegante Adressierungstechnik zurückgegriffen. Es blieb ihnen auch nichts anderes übrig, denn die 80er können nur 64 kB Programmcodes adressieren. Der Speicher wurde in einen herkömmlichen ROM-, einen Switch-ROM-Bereich und in RAM aufgeteilt. 24 kB ROM beinhalten das Basic-System der Rechner. 8 kB sind für selektierbare ROM's reserviert und 32 kB des Speichers stehen dem BASIC zur Verfügung. Den Switch-Bereich kann man sich folgendermaßen vorstellen: Viele 8 kB-ROM's adressieren übereinander den gleichen Speicher. Wenn nun der Programmcodes eines dieser ROM's abgearbeitet werden soll, so wird dieses ROM selektiert und alle anderen werden stillgelegt. Da die ROM-Switch-Routine im unteren „ständig vorhandenen“ ROM-Bereich steht, ist es problemlos möglich von einem Switch-ROM in ein anderes zu verzweigen. Die Struktur des Betriebssystems wurde nun so organisiert, daß die lebenswichtigen und häufig benötigten Routinen (Interrupt-Handling, Programmabarbeitung, Gleitkommaoperationen) im unteren ROM-Bereich liegen. Diese Organisation bringt große Geschwindigkeitsvorteile mit sich. Weiterhin hat man alle wichtigen Interrupt-Routinen sozusagen nach außen geführt. Dabei springt das System, bevor es einen Interrupt bearbeitet, ersteinmal in's RAM, wo normalerweise ein Rücksprungbefehl steht, so daß das System sofort mit der Interrupthandhabung fortfährt. Nun ist es jedoch jederzeit möglich, den Rücksprungbefehl im RAM durch einen Sprung in eine neue Interrupt-Routine zu ersetzen. Auf diesem Wege hat es HP geschafft, ein derart offenes System zu gestalten, daß noch Jahre später neue Peripherie an das System angeschlossen werden konnte, an die man bei der Entwicklung überhaupt noch nicht gedacht hatte. Hierbei ist noch nachzutragen, daß die CPU keine Softwareinterrupts und nur einen Hardwareinterrupt, der dem Programmierer allerdings 100%tig verborgen bleibt, unterstützt. Alle Prozesse, die in die Interrupt-Kategorie fallen, werden dadurch eingeleitet, daß ein externer Baustein eine Information in's RAM schreibt und danach einen Hardwareinterrupt am Prozessor einleitet. Den großen Bildschirmspeicher von 16 kB konnte HP aus Platzgründen nicht im Systemspeicher unterbringen, so daß hier zum ersten Mal in diesem System Multiprozessorbetrieb nötig wurde. Der Bildschirm wird über einen eigenen Spezialcontroller gesteuert. Das Betriebssystem kommuniziert mit diesem Controller über vier Speicherstellen im RAM. Dem Prozessor wird dabei vorgegaukelt, daß es sich um normales RAM handelt, allerdings steuert das Schreiben an diese Speicherstellen alle Bildschirmaktivitäten. Es brauchte übrigens volle 10 Jahre, bis IBM mit seiner EGA-Karte ein gleiches Konzept verwirklichte. Ein großer Nachteil des CRT-Controllers liegt allerdings darin, daß der verwendete Typ nur im vertical-Trace in der Lage ist, Steuerungsparameter und Daten entgegenzunehmen. Daher ist es bei der Serie 80 auch nicht möglich, wirklich schnelle Grafik aufzubauen, denn das System muß 50 mal in der Sekunde auf den CRT-Chip warten. Hätte man zudem den

CRT-Chip, wie alle anderen externen Systemprozessoren auch, nicht auf dem Mother-Board eingebaut, sondern in einem Rechnerlot verfrachtet, so wären die grafischen Möglichkeiten des Systems noch erheblich zu steigern gewesen. Wie bereits kurz erwähnt, wird die gesamte externe Kommunikation über eigene intelligente Einsteckkarten und somit über eigene Prozessoren geführt. Ein voll ausgebaute Serie-80-Rechner (zum Beispiel HP-IB, -IL, RS232, GPIO, Speech-Modul) wird somit zu einem riesigen Multiprozessorsystem, bei dem sich die Haupt-CPU ausschließlich um ihren eigenen Kram kümmert, da ihr die vielen Sklaven die ganze Kommunikation abnehmen. So wie hier beschrieben kam der HP-85 Anno 1979 auf den Markt. Vier Jahre waren also nötig, um diese Maschine unter dem Decknamen Capricorn (Steinbock) zu entwickeln. Das Betriebssystem wurde wahrscheinlich parallel zur Hardware auf einem Computer der gehobenen Datentechnik entwickelt. Das schließe ich aus dem Aussehen der Assemblerlistings. Bei dem verwendeten Entwicklungscomputer hat es sich mit Sicherheit um keine kleine Maschine gehandelt, denn wer einmal versucht die Serie 80-CPU auf einem anderen Rechner zu simulieren wird mit Sicherheit auf einige größere Schwierigkeiten stoßen. Es dürfte in dieser Entwicklungszeit bei HP ähnlich zugegangen sein, wie es Tracy Kidder in seinem Buch „Die Seele einer neuen Maschine“ beschreibt.

Der Erfolg im angepeilten Bereich stellte sich für den HP-85 nach kurzer Zeit ein, was wohl nicht zuletzt daran lag, daß HP mit dem 85er endlich einen Computer auf den Markt brachte, mit dem die firmeneigenen Meß- und Steuergeräte wunderbar betrieben werden konnten.

Etwa im Jahr 1982 hat HP dann den Entschluß gefaßt, trotz IBM's Eintritt in den Personal-Sektor, die HP-85-Struktur nochmals zu überarbeiten. Wieder setzte sich ein Team zusammen und entwickelte unter dem Decknamen Gemini (Zwilling) den HP-87. Drei Schwachstellen des HP-85 hatten sich mittlerweile herauskristallisiert: der freie Speicher war zu klein (maximal 32 kB), der Bildschirm war zu klein (15 Zeilen a 32 Zeichen) und das HP-Standard-Interface (HP-IB) war nicht von vornherein in das System integriert worden. Bildschirm und HP-IB stellten kein Problem dar.

Schwieriger war allerdings die Vergrößerung des Systemspeichers. Schließlich war die Grenze von 64 kB für den 8-Bit-Prozessor nicht nur eine Schallmauer. Aber HP löste das Problem äußerst elegant durch einen tiefen Griff in die Trickkiste. Es wurden zwei weitere Controller in das System integriert. Wie auch mit allen anderen externen Bausteinen kommuniziert die CPU mit den beiden neuen Prozessoren über verschiedene RAM-Stellen. Mit den beiden neuen Bausteinen führte HP meiner Ansicht nach als erste Firma EMC's (Extended Memory Controller) auf dem PC-Markt ein. Die verwendeten EMC's sind in der Lage, einen theoretisch beliebig großen Speicherbereich (praktisch gibt's allerdings Probleme mit der Stromversorgung, so daß nur 770 kB adressiert werden können) als Stapel zu adressieren. Die CPU braucht einem EMC nur mitzuteilen, von wo sie die

Daten herhaben möchte und kann dann hintereinander von diesem Pseudostapel beliebig viel Datenbytes laden. Nach Einführung der EMS's konnte das Speicherproblem ad Acta gelegt werden. Einzig das Betriebssystem (heute BIOS) mußte in 64 kB untergebracht werden, was durch die Switch-ROM's kein Problem war. BASIC-Programme und Daten standen nunmehr im externen Speicher. Um das Konzept zu verwirklichen, brauchte nun nur noch das Betriebssystem des HP-85 so umgeschrieben zu werden, das es auf BASIC-Programme via EMC zugriff. Bei dieser Gelegenheit hat man noch ein paar andere Änderungen in das System eingebracht. Alle Speicherverwaltungsparameter wurden etwas größer dimensioniert. Variablenamen der Programme durften nunmehr 32 Zeichen lang sein. Die Verzweigung zu Zeilenmarken statt ausschließlich zu Zeilennummern rundete das neue System ab. Und wie erwartet schlug der HP-87 im angepeilten Zielbereich auch wie eine Bombe ein.

Im Serie 80-Konzept tat sich nach der Einführung des HP-87 kaum noch etwas. Einzig erwähnenswert ist wohl der HP-86B, bei dem einige Macken des 87-Systems beseitigt wurden. Ferner wurde der Monitor als externes Gerät nach außen verlagert und der Rechner wurde mit seinem Language-ROM im europäischen Bereich vielsprachig. Mit diesem Gerät endete eine leistungsstarke Produktfamilie.

Der HP-75 (Deckname Kanguruh) war meiner Ansicht nach von Anfang an ein totgeborenes Kind, denn HP hat diesem an sich recht guten Computer einige schwerwiegende Fehler eingepflanzt. So z.B. die absolute Speichergrenze von 24 kB (mager, mager) und die fehlende offene Architektur (keine Möglichkeit auf vernünftige Weise einen Monitor anzuschließen). Der HP-75 als erster Laptop mit der Serie 80-CPU (allerdings in CMOS) hat sich aus dem Markt katapultiert, bevor er sich etablieren konnte.

Nun bin ich am Ende der Serie 80-Story angelangt und möchte nur noch einige meiner Ansicht nach schwerwiegende Marketing- und Konzeptfehler aufzeigen:

Die Serie 80 wurde zwar als offenes System konzipiert, es wurden allerdings nie Unterlagen veröffentlicht, die es Fremdfirmen ermöglichten eigenes Zubehör zu entwickeln. Hierbei sei die außergewöhnliche Leistung unseres Clubmitglieds Klaus Kaiser erwähnt, der es trotzdem geschafft hat, mit dem Portextender eine sehr gute Sloterweiterung zu entwickeln.

Der Preis des Systems war die gesamte Zeit zu hoch.

Die Werbung ließ zu wünschen übrig.

Die Taktfrequenz des Prozessors hätte mit heutiger Technologie mit Sicherheit um den Faktor 10 bis 20 vergrößert werden können.

Eine 16- oder 32-Bit Version des Systems wäre mit Sicherheit realisierbar gewesen (Adressarithmetik ändern, ein paar neue 3- oder 4-Byte Register für den Daten- und Programmtransfer)

Da HP die Serie 80 über die Klinge springen ließ ohne ein Nachfolgemodell vorzustellen, denn die als Nachfolge deklarierte Serie 300 kann man vom Preis her einfach nicht als Nachfolge anerkennen, lohnt es sich durchaus, mal in tiefes Nachdenken zu versinken. Will sich HP vom PC-Markt zurückziehen? Die neuen Vectras decken einen ganz anderen Bereich ab, sie sind mit den vorhandenen Betriebssystemen einfach nicht in der Lage eine einfache Meßgeräte- und Anlagensteuerung zu realisieren.

Außerdem ist HP schließlich nicht die einzige Firma, die PC's an den Mann brin-

gen möchte. Dabei bietet HP bei einem hohen Preis nicht einmal die Leistung anderer Hersteller wie z.B. Compaq. Ich weis durchaus, von was ich spreche, denn dieser Text wurde auf einer Vectra geschrieben, und neben mir steht ein Compaq Portable III.

Einige meiner Kunden, die Ihre Anlagen mit Serie 80-Rechnern steuern seit es diese Computerfamilie gibt, sind zumindest stinksauer, und deren Gefühlsregungen kann ich mich ohne weiteres anschließen. Was wird nun? Am besten wird man Surflehrer auf den Malediven.

Dem größten HP-orientierten Anwenderclub in Europa (ich meine natürlich uns) wird wohl nichts anderes übrigbleiben, als mit unwohl Gefühlen zu warten.

Andre Koppel  
Schildhornstraße 23  
1000 Berlin 41

## Lineare Netzwerke

Programme: LINAP1 (833 Bytes)  
LINAP2 + PLOT2 (955 Bytes + )

2 Programme, 1 Subprogramm  
LINAP1 833 Bytes, Mathe-Modul  
LINAP2 955 Bytes, Mathe-Modul, Drucker RX80

### Analyse linearer passiver Netzwerke.

In PRISMA 84.3.25 hat P. Ehrenberg ein Netzwerkanalyse-Programm vorgestellt, mit dem man Schaltungen aus passiven Elementen mit dem HP41C berechnen konnte. Das Programm benutzte das sog. Knotenreduktionsverfahren. Es waren fast alle Arten von Filterschaltungen damit berechenbar, wenn man die Methode der Makro-Modellierung beherrschte. Hierzu war allerdings das Buch von D. Lange [1] sehr empfehlenswert. Wer hiermit gearbeitet hat, kennt die Vor- und Nachteile dieses Programm-Paketes „NET“:

- Das Verfahren ist schnell und holt das Maximum an Rechengeschwindigkeit aus dem HP41C heraus.
- Die „Kunst“ der Makro-Programmierung muß nach 4 Wochen Pause meist wieder erlernt werden.

Diesen letzten Punkt versucht unser neues HP71-Programm zu verbessern. Die Eingabe der Schaltung und deren Topologie ist im Programm LINAP (-Lineares Netzwerk Analyse Programm) fast so einfach wie bei SPICE. Als Lösungsverfahren werden die Knotengleichungen durch Aufstellen der Leitwert-Matrix G in komplexer Form benutzt. Hierbei ist das Mathe-Modul wieder sehr vorteilhaft, obwohl es im Prinzip auch ohne MM mit FOR/NEXT-Schleifen (nur langsamer!) geht.

Das Aufstellen der Knotengleichungen für lineare, passive Netzwerke kann in allen Lehrbüchern der Elektrotechnik z.B. [2] nachgelesen werden. Hierbei ist bekanntlich zwischen der definiten (N-1 Gleichungen) und der indefiniten G-Matrix (N Gleichungen, N Knoten) zu unterscheiden. Bei der definiten G-Matrix wird das Referenz-Potential (Ground) zu Null gesetzt, so daß wir einen Knoten weniger zu berechnen haben.

Bekanntlich lauten die Knotengleichungen in Matrix-Form:

$$G \cdot U = J$$

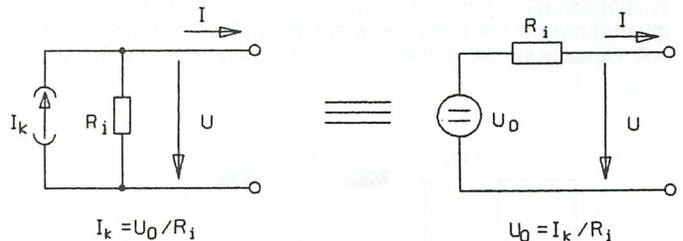
Oder ausführlich:

$$\begin{pmatrix} G_{11} & G_{12} & \dots & G_{1N} \\ G_{21} & G_{22} & \dots & G_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & G_{N2} & \dots & G_{NN} \end{pmatrix} \cdot \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{pmatrix} = \begin{pmatrix} J_1 \\ J_2 \\ \vdots \\ J_N \end{pmatrix}$$

Hierbei wird der Stromquellen-Vektor **J** als gegeben betrachtet und die Spannungen in allen Knoten sollen berechnet werden. Die Ströme ( $J_1, J_2, \dots, J_N$ ) sind die Einströmungen in jeden der N Knoten. In den meisten Fällen können wir uns jedoch auf eine einzige Stromquelle beschränken, so daß der Stromvektor die Form ( $J_1, 0, \dots, 0$ ) erhält. Das Programm ist also spezialisiert auf eine einzige Stromquelle und diese soll im Knoten 1 eingespeist werden.

In den Knotengleichungen werden somit als Generatoren Stromquellen (mit unendlich hohem Innenwiderstand) zugrundegelegt.

Will man die Schaltung jedoch mit einer Spannungsquelle mit vorgegebenem Innenwiderstand betreiben, so gilt bekanntlich folgende äquivalente Umrechnung von Urstromquelle (mit Kurzschlußstrom  $J_k$ ) in Urspannungsquelle (mit  $U_0$ , früher: EMK = elektromotorische Kraft genannt):



LINAP halte ich in 2 Versionen im Rechner gespeichert: Die einfache Version LINAP1 dient zur Berechnung von Knotenspannungen für jeweils eine Frequenz und anschließender Ausgabe auf dem einzeiligen LC-Display.

LINAP2 berechnet Frequenzgänge von einer Start-Frequenz bis zu einer Stop-Frequenz mit vorgebbarem Frequenz-Intervall. Die Ausgabe erfolgt hierbei als Printer-Plot auf einem RX80-Drucker (EPSON), wobei das Programm PLOT2 benötigt wird.

### Hinweise zur Topologie der Schaltung:

Die Beschreibung der Schaltung erfolgt in DATA-Zeilen zu Beginn des Programmes. Die Größe der Schaltung ist im Prinzip nur durch die Größe des Kernspeichers begrenzt. Vorerst sind als Schaltungselemente Ohmsche Widerstände (R), Induktivitäten (L) und Kapazitäten (C) zugelassen. Diese werden in Zeile 10...19 (ich hoffe, 10 Elemente genügen) ins Programm nach folgendem Schema eingebaut:

DATA R, Knoten(von), Knoten(nach), Wert  
DATA C, Knoten(von), Knoten(nach), Wert  
DATA L, Knoten(von), Knoten(nach), Wert

Als Werte sind nur Konstanten im geeigneten Maßsystem (z.B.: Ohm, Henry, Farad, Hertz) zugelassen.

Die Knoten im Schaltbild werden vom Benutzer lückenlos durchnummeriert. Hierbei erhält der Eingangsknoten die Nummer 1 und der Ground-Knoten die letzte (höchste) Knoten-Nummer. Die Reihenfolge der Durchnummerierung ist im Prinzip beliebig. Es empfiehlt sich, den Ausgangsknoten mit 2 zu bezeichnen, damit wir bei einer Spannungsübertragung in den Outputs immer die Bezeichnung  $U(2)/U(1)$  beibehalten können.

Grundsätzlich werden durch den SYS-Befehl alle komplexen Knotenspannungen nach jeder Frequenzeingabe neu berechnet, die Sie sich aber nicht ausdrucken lassen müssen. Der Schaltungsentwickler benötigt oft nur die Spannungsverstärkung nach Betrag

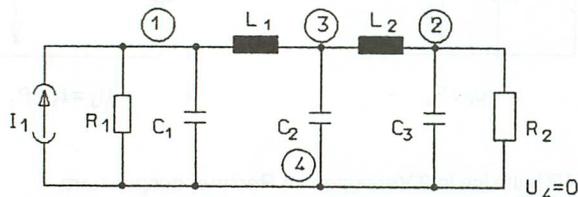
und Phase, wobei der Betrag mitunter auch in  $[dB] = 20 \cdot \text{LOGT}(\text{ABS}(U(2)/U(1)))$  verlangt wird. Einen Eingangswiderstand formuliert man als  $U(1)/J1$ . Einen Ausgangswiderstand berechnet man durch Vertauschen der Knoten 1 und 2 und Berechnung wie ein Eingangswiderstand.

**Benutzer-Hinweise:**

1. Programm eingeben. LINAP2 erfordert zusätzlich PLOT2.
2. DATA-Zeilen gegebenenfalls ändern.
3. Ausgangsgröße(n) im Programm formulieren.
4. Starten mit RUN
5. Das Programm fragt die Anzahl der Elemente der Schaltung mit  $R+L+C=$  ab. Diese ist identisch mit der Zahl DATA-Anweisungen.
6. Das Programm fragt die Anzahl der Knoten der Schaltung inklusive Ground-Knoten (Referenz-Knoten) ab.
7. Bei LINAP2 werden die Plotgrenzen in Y-Richtung vom Programm abgefragt, wobei  $Y1, Y2 = \text{Min, Max}$  entspricht.
8. Bei LINAP1 wird nach der Eingangsfrequenz abgefragt.
9. Bei LINAP2 wird der Reihe nach einzeln nach Start-Frequenz, Stop-Frequenz und Frequenz-Intervall abgefragt. Die zu plottende Funktion muß in Punkt 3. in der Variablen  $Y0$  abgelegt und als Ausdruck gegebenenfalls formuliert sein.

**Beispiel:**

Wir betrachten folgendes Tiefpass-Filter, das mit einem Spannungsgenerator von 1 Volt und Innenwiderstand 50 Ohm angeregt werden soll. Gesucht sei der Betrag des Übertragungsfaktors  $U(2) : U(1)$ .



Werte der Schaltelemente:

- |               |             |
|---------------|-------------|
| R1 = 50 Ohm   | L1 = 364 nH |
| C1 = 103.3 pF | L2 = 364 nH |
| C2 = 191.4 pF | R2 = 50 Ohm |
| C3 = 103.3 pF | J1 = .02 A  |

DATA-Zeilen der Topologie der obigen Schaltung:

- 10 DATA R, 1, 4, 50
- 11 DATA C, 1, 4, 103.3E-12
- 12 DATA L, 1, 3, 364E-9
- 13 DATA C, 3, 4, 191.4E-12
- 14 DATA L, 3, 2, 364E-9
- 15 DATA C, 2, 4, 103.E-12
- 16 DATA R, 2, 4, 50

Das Programm verlangt für obige Schaltung folgende Antworten:

- „Anzahl der  $R+L+C$ “ = 7
- „Knoten inclusive Ground“ = 4
- „JEIN“ = .02

Happy Programming!

Peter JOCHEN  
Heilbronner Straße 240  
7410 Reutlingen

**Literatur-Hinweise:**

- [+] D. Lange: „Algorithmen der Netzwerkanalyse für programmierbare Taschenrechner (HP-41C)“, Vieweg-Verlag, 1982.
- [2] D.A. Calahan: „Rechnergestützter Schaltungsentwurf“, Oldenbourg-Verlag, München 1973.

Programm: " PLOT2 "

```

10 ! PLOT2
20 ! MIT SUB-PROGR.
200 SUB PLOTA(R1,R2)
205 M=(R2+R1)/2 @ FIX 2
210 PRINT 'ORDINATE: A=';R1; ' B=';M; ' C=';R2
215 PRINT TAB(3); 'A';TAB(37); 'B';TAB(71); 'C'
220 FOR V=1 TO 69
225 IF V=2 THEN 235
230 IF V#36 THEN 245
235 PRINT TAB(V+1); '*'
240 GOTO 250
245 PRINT TAB(V+1); '+'
250 NEXT V
255 PRINT TAB(71); '*'
260 END SUB
300 SUB PLOTY(Y,R1,R2)
305 T=(R2-R1)/68
310 N=(Y-R1)/T+2
315 IF N>70 THEN N=70
320 IF N<2 THEN N=2
325 PRINT TAB(2); '+';TAB(N+1); '*'
330 END SUB
    
```

Programm: " LINAP1 "

```

5 ! LINAP1 FUER PASSIVE NETZWERKE
6 OPTION BASE 1 @ DEGREES
7 DESTROY ALL
10 DATA R,1,4,50
11 DATA C,1,4,103.3E-12
12 DATA L,1,3,364E-9
13 DATA C,3,4,191.4E-12
14 DATA L,3,2,364E-9
15 DATA C,2,4,103.3E-12
16 DATA R,2,4,50
17 DATA
20 INPUT 'JEIN=';J1
25 INPUT 'ANZ.D:R+L+C=';NO
45 REAL F,W
50 DIM T$(7)[1]
51 DIM N1(7),N2(7),V1(7)
55 INPUT 'NODES INCL.GRND=';N9
61 RESTORE 10
65 FOR K=1 TO NO
70 READ T$(K),N1(K),N2(K),V1(K)
75 NEXT K
80 COMPLEX G(N9,N9),U(N9),I(N9),A
100 'START': INPUT 'FREQ=';F @ W=2*PI*F
105 MAT G=ZERO
110 FOR K=1 TO NO
115 IF T$(K)='R' THEN A=1/(V1(K),0)
120 IF T$(K)='L' THEN A=1/(0,W*V1(K))
125 IF T$(K)='C' THEN A=(0,W*V1(K))
130 G(N1(K),N1(K))=G(N1(K),N1(K))+A
135 G(N2(K),N2(K))=G(N2(K),N2(K))+A
140 G(N1(K),N2(K))=G(N1(K),N2(K))-A
145 G(N2(K),N1(K))=G(N1(K),N2(K))
150 NEXT K
151 FOR K=1 TO N9
152 G(N9,K)=(0,0) @ G(K,N9)=(0,0)
153 NEXT K
155 MAT I=ZERO @ I(1)=(J1,0)
160 MAT U=SYS(G,I)
161 ! FOR L=1 TO N9-1
162 ! DISP 'U#';L; '=';U(L)
163 ! PAUSE
164 ! NEXT L
175 DISP '/VU/,PHI=';POLAR(U(2)/U(1))
180 PAUSE
185 GOTO 'START'
190 END

```

Programm: " LINAP2 "

```

5 ! LINAP2 FUER PASSIVE NETZWERKE MIT P PLOT2
6 OPTION BASE 1 @ DEGREES @ PWIDTH 80
7 DESTROY ALL
10 DATA R,1,4,50
11 DATA C,1,4,103.3E-12
12 DATA L,1,3,364E-9
13 DATA C,3,4,191.4E-12
14 DATA L,3,2,364E-9
15 DATA C,2,4,103.3E-12
16 DATA R,2,4,50
17 DATA
20 INPUT 'JEIN=';J1
25 INPUT 'ANZ.D:R+L+C=';NO
45 REAL F,W
50 DIM T$(7)[1]
51 DIM N1(7),N2(7),V1(7)
55 INPUT 'NODES INCL.GRND=';N9
57 DISP 'GRENZEN Y1,Y2=';
58 INPUT R1,R2 @ IF R2<=R1 THEN 58
59 CALL PLOTA(R1,R2) IN P PLOT2
60 PRINT CHR$(27);"3";CHR$(18);
65 FOR K=1 TO NO
70 READ T$(K),N1(K),N2(K),V1(K)
75 NEXT K
80 COMPLEX G(N9,N9),U(N9),I(N9),A
85 INPUT 'START-FREQ=';F1 @ W1=2*PI*F1
86 INPUT 'STOP-FREQ=';F2 @ W2=2*PI*F2
87 INPUT 'FREQ-STEP=';F3 @ W3=2*PI*F3
100 FOR W=W1 TO W2 STEP W3
105 MAT G=ZERO
110 FOR K=1 TO NO
115 IF T$(K)='R' THEN A=1/(V1(K),0)
120 IF T$(K)='L' THEN A=1/(0,W*V1(K))
125 IF T$(K)='C' THEN A=(0,W*V1(K))
130 G(N1(K),N1(K))=G(N1(K),N1(K))+A
135 G(N2(K),N2(K))=G(N2(K),N2(K))+A
140 G(N1(K),N2(K))=G(N1(K),N2(K))-A
145 G(N2(K),N1(K))=G(N1(K),N2(K))
150 NEXT K
151 FOR K=1 TO N9
152 G(N9,K)=(0,0) @ G(K,N9)=(0,0)
153 NEXT K
155 MAT I=ZERO @ I(1)=(J1,0)
160 MAT U=SYS(G,I)
161 Y0=ABS(U(2)/U(1))
162 CALL PLOTY(Y0,R1,R2) IN P PLOT2
165 NEXT W
167 PRINT CHR$(27);"2";
190 END

```

# Update

## HP-71 Benutzerhandbuch

Bei der Umdimensionierung einer Variablen kann bei der Betriebssystem-Version 2CCCC eine Veränderung des Speicherinhalts, bzw. „memory lost“ auftreten. Dies ist dann der Fall, wenn eine skalare Variable (REAL) existiert, deren Name mit dem gleichen Buchstaben beginnt.

Eine Umdimensionierung kann mit den Anweisungen

**DIM, INTEGER, REAL, SHORT, STAT, COMPLEX**  
und **COMPLEX SHORT**

erreicht werden. Um die oben genannten Probleme damit zu vermeiden, müssen bei Verwendung dieser Befehle folgende Regeln beachtet werden:

1. Zuerst **DESTROY** auf die Variable anwenden, dann erst umdimensionieren, bzw. ihren Typ ändern.
2. **DESTROY** auf alle skalaren REAL-Variablen anwenden, deren Namen mit demselben Buchstaben anfangen, dann umdimensionieren (für jeden Buchstaben des Alphabets sind 11 Variablen-Namen möglich: z.B. R, R1, ..., R9).
3. Am besten einen bestimmten Buchstabenbereich nur für skalare REAL-Variablen verwenden (z.B. A .. M), den Rest (N .. Z) für alle anderen Variablen.

## Beispiele:

Methode 1:

DIM L,L0,L3,L1,L7  
DESTROY L0 DIM L0(4)

5 skalare REAL-Variable  
Umdimensionierung nach  
DESTROY

Methode 2:

DIM R,R1\$,R2,R0(3,5),Q0,T,X  
DESTROY R,R0,R1,R2

verschiedene REAL-Variable  
R1\$ soll umdimensioniert werden,  
alle „R“-Variablen müssen  
DESTROY werden  
Umdimensionierung von R1\$

DIM R1\$[10]

Methode 3:

DIM A\$, B1,L5,L6,L7  
DIM N(3),N1(5)  
SHORT Q0,R0  
INTEGER P,Q,R  
COMPLEX R1  
DIM N(5),R0(3)

A..M für skalare REAL-Variable  
N..Z für alle anderen Variablen

in Ordnung, um Variable im Bereich  
N..Z umzudimensionieren

Quelle: Update 00071-90107 (Juli 87, englisch)

Henry Schimmer

# Telefonverzeichnis

## Tricks in einem BASIC-Programm und Beschreibung der Arbeitsweise

### 1013 Bytes

Das von mir geringfügig veränderte Programm stammt aus dem Buch „HP-71-BASIC leicht gemacht“ von W. Stroinski. Ich habe den DATA-File WORT genannt.

Es werden nur die interessantesten Raffinessen beschrieben.

Tastenfunktionen (aus dem Englischen übernommen):

- A: Eingabe von Name u. Telefonnummer (Append)
- B: „Blättern“ rückwärts (Backward)
- D: Löschen von Einträgen (Delete)
- F: „Blättern“ vorwärts (Forward)
- S: Suchen eines Eintrags (Search)
- U: Eintrag ändern (Update)
- Z: Verzeichnis löschen (Zap)
- Q: Verzeichnis wird nicht mehr gebraucht aber nicht gelöscht (Quit)

Die Startroutine:

Beim ersten Starten mit RUN versucht der Computer dem noch nicht bestehenden File WORT die Kanalnr. #1 zuzuordnen. Das ergibt ERROR und Verzweigung zu Zeile 20 und weiter nach 370, weil WORT noch nicht existiert. Die ERROR-Bedienung wird in 370 aufgehoben, die Frage „Welcher Umfang“ (=Wieviel Einträge soll das Verzeichnis maximal haben) wird gestellt und neben anderen Variablenzuordnungen der File WORT eingerichtet. Beim erneuten Starten entsteht in 30 kein ERROR mehr; das Programm läuft bis zur „Tastenschleife“ (Zeilen 40 u. 50), wo der Computer auf neue Befehle wartet.

## Befehlstastenaktivierung mit GOTO KEY\$:

In Zeile 50 entsteht ERROR, wenn keine Befehlstaste gedrückt worden ist (GOTO KEY\$ hat dann nämlich keine Zielzeile). Deshalb wird eine Schleife zwischen Zeile 50 u. 40 andauernd und in schneller Folge ausgeführt. Sobald aber eine Taste, z.B. A, gedrückt wird, wird mit GOTO KEY\$ zu der entsprechenden Zeile verzweigt und die zugehörige Routine ausgeführt. Danach läuft wieder die Tastenschleife, bis erneut eine der Befehlstasten gedrückt wird. Am Schluß jeder Programmroutine steht deshalb GOTO 50.

Eingabe eines aus Name u. Telefonnummer bestehenden Eintrags:

Nach Drücken von A wird geprüft, ob die zu Anfang gewählte und in M stehende Zahl der Einträge etwa schon erreicht worden ist; wenn ja -Anzeige „Voll“ und S\$ zunächst mit Leerstrings "" belegt, außerdem wird die Zählvariable N um eins erhöht und der Recordzähler R=N gesetzt. In 190 wird nach der Eingabeaufforderung „Name“ dieser in N\$ abgelegt; wird ohne Eingabe END LINE gedrückt, so wird die Eingabeaufforderung wiederholt, andernfalls wird zur Eingabe der Telefonnummer aufgefordert; sie wird in S\$ und P\$ aufgenommen. Beide Informationen werden anschließend, im Data-File WORT unter der Recordnr. R eingetragen. Danach geht es wieder zur Befehlsschleife 40/50.

Die Suchroutine, Taste S:

Wenn noch gar kein Eintrag vorhanden ist (N=0), Verzweigung zu 90: „kein Eintrag“ und weiter zur Tastenschleife, andernfalls Eingabeaufforderung „Suche“ und Ablegen des Suchwortes in S\$ sowie Umwandlung

aller eventueller Klein- in Großbuchstaben. Die Suche läuft mittels der Schleife FOR R=1 TO N. Damit werden der Reihe nach solange die Einträge in WORT aufgerufen, bis die eingegebene Zeichen- bzw. Buchstabengruppe mittels der Anweisung POS(.....)=1 als Teil eines grade durch die Schleife aufgerufenen Records gefunden wurde. Dann Anzeige des betreffenden Namens mit „?“ . Wird daraufhin Y=Ja gedrückt, so wird der 2. Teil des Eintrags (Tel.) angezeigt (Zeile 80). bei N=Nein, wird mit Zeile 270 nach einem weiteren Namen gesucht, der die gewünschte Zeichengruppe enthält.

Die Möglichkeiten: Aufforderung Y oder N einzugeben, oder Wiederholung dieser Aufforderung, werden durch Zeile 290 gewährleistet: Wird weder Y noch N gedrückt, so ist POS(„YN“,KEY\$)+1=1; Verzweigung zu 260. Wird Y gedrückt, so wird der o.g. Ausdruck gleich 2, denn Y steht in „YN“ an erster Stelle; das bedeutet Verzweigung nach 80. Im dritten Fall ist der Ausdruck 3; Verzweigung nach 270.

Nach diesen Ausführungen wird der Leser wohl kaum noch Schwierigkeiten haben, den Rest dieses lehrreichen Programms zu analysieren.

Walter Prankel  
Brunckstr. 21  
6710 Frankenthal

```

10 DIM N$(30),S$(30),P$(30),N,R,M
20 ON ERROR GOTO 370
30 UNSECURE WORT @ ASSIGN #1 TO WORT @ READ #1;N,M
40 DISP "Bereit" @ ON ERROR GOTO 50
50 GOTO KEY$
60 'Q': PRINT #1,0;N,M @ SECURE WORT @ DISP "ENDE!" @ END
70 'P': IF R=0 THEN 90
80 DISP P$ @ GOTO 50
90 DISP "KEIN EINTRAG!" @ GOTO 50
100 'N': IF R=0 THEN 90
110 DISP N$ @ GOTO 50
120 'D': IF R=0 THEN 90
130 IF R=N THEN 150
140 READ #1,N;N$,P$ @ PRINT #1,R;N$,P$
150 R=0 @ N=N-1 @ DISP "GELÖSCHT" @ GOTO 50
160 'A': IF N=M THEN DISP "VOLL!" @ GOTO 50
170 N$="" @ S$="" @ N=N+1 @ R=N
180 ON ERROR GOTO 190
190 LINPUT "NAME: ";N$ @ IF N$="" THEN 190
200 ON ERROR GOTO 210
210 INPUT "TEL.: ",S$;S$ @ P$=S$
220 PRINT #1,R;N$,P$ @ GOTO 40
230 'S': IF N=0 THEN 90 ELSE LINPUT "SUCHE: ";S$ @ S$=UPRC$(S$)
240 FOR R=1 TO N @ READ #1,R;N$,P$
250 IF POS(UPRC$(N$),UPRC$(S$))=0 THEN 270 ELSE DISP N$;"?"
260 ON POS("YN",KEY$)+1 GOTO 260,80,270
270 NEXT R @ R=0 @ GOTO 90
280 'Z': DISP "ALLE";N;"NAMEN?"
290 ON POS("YN",KEY$)+1 GOTO 290,300,40
300 PURGE WORT @ GOTO 370
310 'F': IF N=0 THEN 90 ELSE R=R+1 @ IF R>N THEN R=1
320 READ #1,R;N$,S$ @ GOTO 110
330 'B': IF N=0 THEN 90 ELSE R=R-1 @ IF R<1 THEN R=N
340 GOTO 320
350 'U': IF R=0 THEN 90 ELSE S$=P$ @ GOTO 180
370 OFF ERROR @ INPUT "WELCHER UMFANG? ";M @ N=0
380 CREATE DATA WORT,M+1,50 @ ASSIGN #1 TO WORT
390 PRINT #1;N,M @ GOTO 40
    
```

Telefonverzeichnis

# GAS

## Molmasse und Volumenprozent

### 304 Bytes

In letzter Zeit benötige ich öfters die Berechnung der Zusammensetzung eines Gemisches aus zwei Gasen mit gegebener Dichte. Zu diesem Zweck habe ich folgende kleine Routine geschrieben, die sehr hilfreich sein kann. Allerdings muß bekannt sein, um welche beiden Gase es sich handelt, es müssen nämlich die Molmassen dieser beiden Gase eingegeben werden. Ferner muß der Druck und die Temperatur zum Zeitpunkt gegeben sein, zu dem die Dichte ermittelt wurde. Nach Eingabe der benötigten Größen wird die Molmasse des Gasgemisches sowie dessen Zusammensetzung in Volumen-% ausgegeben.

### Beispiel:

Die Dichte eines Methan-Wasserstoff-Gemisches beträgt bei  $p = 1050$  mbar und  $t = 30$  Grad  $0,4628$  g/l.

Welche Zusammensetzung hat das Gemisch in Volumen- bzw. Mol.-%?

Ausgaben	Eingabe
1. Gas :	CH4    ENDLINE
1. Molmasse :	16      ENDLINE
2. Gas :	H2      ENDLINE
2. Molmasse :	2       ENDLINE
Dichte (g/l) :	.4628   ENDLINE
t (Grad) :	30      ENDLINE
Druck (mbar) :	1050   ENDLINE

M=11.10 CH4:65% H2:35%

→ Das Gasgemisch hat eine Molmasse von 11.10 g/mol. 65% des Gemisches sind Methan. 35% Wasserstoff.

```

10 Delay 0,0 $ Fix 2
20 Input "1.Gas : ";G1$
30 Input "1.Molmasse : ";M1
40 Input "2.Gas : ";G2$
50 Input "2.Molmasse : ";M2
60 Input "Dichte (g/l) : ";S
70 Input "t (Grad) : ";T
80 Input "Druck (mbar) : ";P
90 V0=P*273/((273+T)*1013)
100 D=S/V0 $ M=D*22.4
110 Y=(M1-M)/(M1-M2) $ X=1-Y
120 Disp "M="&Str$(Ip(M*100)/100)&" "&
    G1$&":"&Str$(X)(3,4)&"% "&G2$&":"&
    Str$(Y)(3,4)&"%"

```

Dennis Föh  
 Drangstweg 1  
 2190 Cuxhaven

## Lex-File: Findlex

### 329 Bytes, HP-71

Dieser LEX-File bringt eine große Hilfe für alle BASIC-Programmierer und -Anwender. Wie oft schon stand man vor dem Problem, eine bestimmte Stelle in einem längeren BASIC-Programm zu finden, ohne das Programm Zeile für Zeile absuchen zu müssen, oder ein gedrucktes Listing zu erstellen und das Ganze dann auf Papier zu tun. Auch ergibt sich oft die Notwendigkeit, ein Programm daraufhin zu überprüfen, ob eine bestimmte Variable bereits verwendet wurde, oder nicht.

Der vorliegende LEX-File schafft Abhilfe. Er wurde geschrieben von Jean-Jacques Moreau vom französischen SIG und wurde in deren Clubjournal JPC Nr. 25 und 31 veröffentlicht.

Die Größe des Files ist 329 Bytes. Es ist nur ein Schlüsselwort vorhanden.

Filename: FINDLEX  
 Filesize: 329 Bytes  
 ID# Hex: E1

Word	Token	Char
FIND	4B	7

Der genaue Befehl lautet FIND *string*, wobei *string* eine mit Anführungszeichen versehene Zeichenkette oder ein Stringausdruck sein muß. FIND ist nicht programmierbar, kann also nur direkt vom Tastenfeld aus eingegeben werden. FIND durchsucht den momentanen BASIC-File nach den Zeichen, die der String *a* vorgibt. Die Suche beginnt bei der nächsten nach der aktuellen Zeile. Wird der String gefunden, so wird die entsprechende BASIC-Zeile zur aktuellen gemacht und im EDIT-Modus angezeigt.

Es empfiehlt sich, da meist das ganze Programm durchsucht werden soll, vor der ersten Ausführung „g“ und „Pfeil nach oben“ auszuführen, um die erste Zeile zur aktuellen zu machen. Nach jeder Fundstelle kann man dann einfach „g“ „CMDS“ und ENDLINe ausführen, um nach dem nächsten Vorkommen zu suchen. Wird der String *a* im Programm nicht gefunden, oder war die letzte Anzeige die letzte gefundene Stelle, so wird eine Fehlermeldung ausgegeben.

### Beispiel:

Angenommen, Sie besitzen das BASIC-Programm EDTEXT aus dem HP-Text Editor-, oder aus dem FORTH/Assembler Modul.

Tippen Sie:

```

EDIT EDTEXT (Endline)
FIND „GOSUB 'N' “ (Endline)

```

Nun wird Zeile 289 angezeigt, in der sich tatsächlich die gesuchte Anweisung befindet. Um die nächste Stelle im Programm zu finden, rufen Sie den Commandstack auf mit CMDS, wo an erster Stelle immer noch der FIND-Befehl stehen müßte, und drücken Sie ENDLINe. Auch in Zeile 2996 sehen wir, daß der gesuchte Befehl vorkommt. Nach nochmaliger Ausführung wird dann die Fehlermeldung angezeigt.

Achten Sie darauf, daß die gesuchten Zeichen beim Aufruf des FIND-Befehls unbedingt unter Anführungszeichen stehen müssen (einfache oder doppelte), sonst wird versucht, den Begriff als Variable zu interpretieren.

### Beispiel:

Tippen Sie in einem neuen File (z.B. workfile) den kleinen 3-Zeiler ein:

```

10 FOR I=1 TO 13
20 DISP I;A$
30 NEXT I

```

Nun weisen Sie der Variablen A\$ den String „NEXT“ zu: A\$=„NEXT“ (Endline). Dann noch „g“, „Pfeil nach oben“ und „ON“ ausführen. Tippen Sie nun:

```
FIND 'A$' (Endline)
```

so wird Zeile 20 angezeigt, da sich „A\$“ dort befindet. Tippen Sie aber (wieder „g“, „Pfeil nach oben“ und „ON“):

```
FIND A$ (Endline)
```

so wird Zeile 30 angezeigt, da dort „NEXT“ steht, was dem Inhalt der Variablen A\$ entspricht.

### Hexdump-Listing: FINDLEX

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

FINDLEX	L	ID#E1	329 Bytes		
	0123	4567	89AB	CDEF	ck
000:	6494	E444	C454	8502	7E
001:	802E	0040	0130	4068	37
002:	1720	01EB	4B40	0000	C2
003:	F710	0020	0000	0000	B5
004:	0E30	0077	6494	E444	D2
005:	B41F	F101	0811	08E4	C3
006:	F647	0264	F657	E646	AE
007:	CFF8	DD97	309F	FFF8	16
008:	F681	F08F	13DB	08A8	DE
009:	E41B	B88F	2140	1371	52
00A:	35C2	CECE	1641	448F	99
00B:	8BB8	18FE	EB60	4018	EF
00C:	F627	705D	031F	38D3	CE
00D:	9390	8F99	9705	1120	E9
00E:	3310	E58D	A939	01F8	43
00F:	E7F2	D215	F38F	4EFF	D4
010:	05CD	137D	5137	8F13	82
011:	0018	B370	D413	1DB8	FE
012:	FD3D	E013	7135	10B8	E0
013:	FF6F	4020	D4C4	1B1C	73
014:	6F21	46C2	10A1	351B	F5
015:	098F	2146	1091	8414	C5
016:	6EA4	E5AD	1D88	1DCE	AC
017:	80D0	F6D7	5D2D	9061	F4
018:	3706	1371	6117	1DBD	0B
019:	58FF	E1B1	0713	507D	83
01A:	5562	CD11	9134	4511	D4
01B:	4A1C	114F	9620	CCD5	EC
01C:	2F11	A135	11AA	F0DA	41
01D:	137E	A81C	2011	B135	B2
01E:	8A8E	68F9	6F40	1BF8	57
01F:	5F21	4613	41F1	C6F2	F3
020:	AD01	43EA	81C1	00D9	E9
021:	067C	00B1	C3E3	B1E3	AA
022:	FF07	1358	FE0C	1007	51
023:	D58F	6179	0118	8F1C	FB
024:	6908	F898	108D	E730	C6
025:	08FC	2DE0	D78F	1300	C4
026:	1208	B360	657E	11B1	25
027:	3710	B8FD	A670	11B1	B3
028:	3549	08D8	4A80	6F9E	38
029:	F				46

Michael Fiedler  
 Friedrichstr. 17  
 6070 Langen

**Funktion:**

CARDFILE kombiniert eine speicherresidente Datenbank für Namen, Adressen und Telefonnummern mit einem Hayes-kompatiblen Modem und dessen Auto-Dial-Funktion.

**Format:**

[d:][Prad][Dateiname]

**Typ:**

Intern Extern  
\*\*\*

**Hinweise:**

CARDFILE wird normalerweise mit dem Namen einer Datenbank aufgerufen. Mit <Alt-RShift> (RShift <=> Rechte Shift-Taste) kann es dann jederzeit aktiviert werden.

Dann sind folgende Funktionen verfügbar:

- <ESC> → beendet CARDFILE
- <F1> → erstellt eine neue Karteikarte
- <F2> → editiert die angezeigte Karteikarte
- <F3> → löscht eine Karteikarte
- <F4> → sucht eine Karteikarte
- <F5> → speichert die Datenbank
- <F6> → wählt die Nummer der angezeigten Karteikarte

Mit <PgDn> und <PgUp> kann man in der Datenbank blättern.

Beim Wählen der Telefonnummer ignoriert CARDFILE alle nicht numerischen Zeichen mit Ausnahme des Kommas. Es kann eingefügt werden, um beim Wählen kleine Pausen einzufügen, was manchmal nötig ist.

**Funktion:**

CTYPE dient zum Einstellen der Cursorform. CTYPE arbeitet mit allen Grafikkarten (MDA/HGC, CGA, EGA und VGA) zusammen. Die Einstellung erfolgt entweder direkt per Aufrufparameter oder interaktiv.

**Format:**

CTYPE [/xy]

**Typ:**

Intern Extern  
\*\*\*

**Hinweise:**

Wird CTYPE ohne /xy aufgerufen, erfolgt die Einstellung im interaktiven Modus. Dort werden die zu dem jeweiligen Cursortyp zugehörigen Parameter (/xy) zusätzlich mit angezeigt.

Die Parameter und Cursortypen variieren zwischen den einzelnen Grafikkarten.

## CCDe.V.

## SNIPPER

### Funktion:

Überträgt einen beliebigen Teil des Bildschirms auf den Drucker, in eine Datei oder als Tastatureingabe zu einem Anwendungsprogramm.

### Format:

[*d*:] [*Pfad*] SNIPPER [*Zeilen, Spalten*]

### Typ:

Intern Extern

\*\*\*

### Hinweise:

SNIPPER ist ein speicherresidentes Programm, das, einmal geladen, mit <Alt-W> aufgerufen wird. Danach kann mit den Pfeiltasten ein beliebiger Teil des Bildschirms markiert werden. Dazu fährt man die obere linke Ecke an und drückt die Eingabetaste. Dasselbe muß man mit der unteren rechten Ecke wiederholen. Daraufhin erscheint in der linken Bildschirmcke ein Menü mit folgenden Optionen:

- F – dient dazu, den markierten Bildschirmbereich in eine Datei zu schreiben. Wird kein Dateiname eingegeben, nimmt SNIPPER SCREEN.CUT an.
- P – druckt den markierten Bereich auf dem Drucker aus.
- S – legt den markierten Bereich in einem internen Puffer ab. Er kann später daraus wieder abgerufen werden, indem man beim nächsten Aufruf von SNIPPER (<Alt-W>) anstatt einen Bereich zu markieren gleich F, P oder G eingibt.
- G – übergibt den markierten Bereich dem zur Zeit laufenden Anwendungsprogramm als Tastatureingabe.

ESC – verläßt SNIPPER.

Um den EGA-Modus mit 43 Zeilen zu unterstützen, muß SNIPPER gleich beim ersten Aufruf mit den entsprechenden Parametern aufgerufen werden (SNIPPER Zeilen, Spalten)

SNIPPER darf nicht aufgerufen werden, während sich der Rechner im Grafikmodus befindet (WINDOWS, WORD, CAD, etc.).

## CCDe.V.

## DIRNOTES

### Funktion:

DirNotes ist ein Utility zum Kommentieren von Inhaltsverzeichnisn. Jede einzelne Datei kann mit einem Kommentar versehen werden, der maximal 38 Zeichen lang sein darf. Die Kommentare werden in einer Datei abgespeichert und beim nächsten Laden von DirNotes wieder aktiviert.

### Format:

DIRNOTES [*d*:] [*Pfad*]

### Typ:

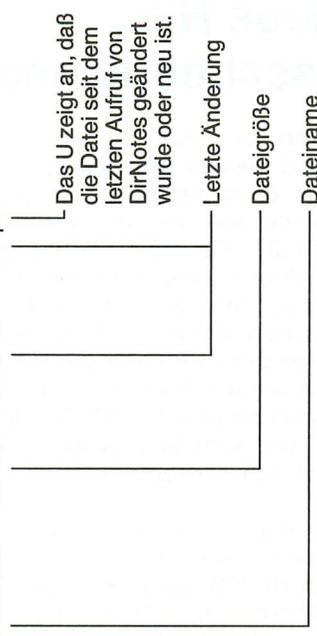
Intern Extern

\*\*\*

### Hinweise:

Die Dateinamen werden im folgenden Format angezeigt:

DIRNOTES ASM 23468 9-27-87 4:18pU



Die Kommentare werden in einer Datei abgespeichert, deren Namen vom kommentierten Verzeichnis abhängt. Die ersten 5 Zeichen sind „DIRN-“ darauf folgen die ersten 3 Zeichen des Verzeichnisnamens und „.DAT“. Beim Wurzel-Verzeichnis (/) lautet der Name „DIRN-----.DAT“.

## HP-75 Speed-Up

SPEEP UP YOUR HP75C

Rechner vorsichtig aufschrauben. An der Stirnseite 3 Schrauben entfernen. Achtung: Schrauben sind unterschiedlich lang! Sehr vorsichtig die obere Platine mit Hilfe eines Schraubendrehers anheben und zur Seite nach links oben legen. Die untere CPU-Platine durch Lösen der zwei Schrauben anheben und ausbauen. Links neben dem Kartenleser über der Stifteleiste befinden sich drei Widerstände. Parallel zu dem obersten lötet man einen zusätzlichen Widerstand von etwa 390 kOhm (ein bißchen kann man noch experimentieren). Man kann diesen Widerstand sehr schön in die Platine einlöten, denn die Löcher sind schon dafür vorgesehen!!? Man sollte das SPEEDUP umschaltbar ausführen, denn der Kartenleser steigt sehr schnell aus (verunreinigte Karten bereiten Probleme). Sollte die IL aussteigen, muß auch der Oszillator verändert werden. Dazu wird der Kondensator, der zwischen dem zweiten und dritten Chip der oberen Reihe (von links gesehen) liegt, gegen einen 82pF Kondensator ausgetauscht. Der Rechner läuft danach etwa 1,8-fach schneller. Leider verändert sich die Frequenz des Beepers, die Wait-, Delay- und Timerfunktionen bleiben dagegen unverändert. Wenn es Probleme gibt, kann ich Auskunft geben.

Happy Tuning

Ralf Rosche  
 Berner Heerweg 155 B  
 2000 Hamburg 72

## Neue HP-Taschenrechner

Gleich vier neue Taschenrechner auf einmal werden von Hewlett-Packard für März '88 vorgestellt. Als kaufmännische Rechner kommen der HP-17B und der HP-19B. Für den technisch-wissenschaftlichen Bereich werden der HP-27S und der HP-28S vorgestellt. Es hat den Anschein, daß damit die bisher gelieferten Rechner HP-18CD und HP-28CD ersetzt werden. Weiterhin bleiben im Programm der HP-11C, HP-12C, HP-15, HP-16C und auch der altbewährte HP-41CX. Doch nun im einzelnen:

Im Vergleich zum HP-18CD haben die beiden neuen Finanzrechner HP-17B und HP-19B auch Lösungen für Obligationen und Abschreibungen eingebaut. Der Arbeitsspeicher wurde auf 8 KByte angehoben. Außerdem wurden beim 19'er auch mathematische Funktionen, Währungsumrechnungen, Datenbankfunktionen und Graphikmöglichkeiten implementiert.

Der technisch-wissenschaftliche HP-27S hat nur ein 2-zeiliges Display, keine Graphik und keine UPN-Eingabe mehr. Sein Arbeitsspeicher beträgt ebenfalls 8 KByte.

Entschieden verbessert wurde dagegen der neue HP-28S. Mit 32 KByte RAM schlägt er jeden Speicherplatzmangel tot und zusätzlich wurde er gegenüber seinem Vorgänger um einen Graphikspeicher und eine entsprechende Graphikorganisation bereichert.

Obwohl die Preise noch nicht bekannt sind, darf man auf diese neuen Rechner sehr gespannt sein.

Alf-Norman Tietze

## Mitglieder- versammlung

Am Samstag, den 16. April 1988  
 um 11 Uhr,  
 im Bürgerhaus Frankfurt-Nordwest,  
 Niddacorso 5,  
 6000 Frankfurt am Main  
 (Nordweststadt)

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**

# Die Barcodes des HP41

Barcodes, was ist das ?

– 2. Teil –

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

## Programmbarcodes für den HP41

befassen, da kann man sich als Programmierer so richtig dran austoben, aber keine Angst, so kompliziert ist das Thema nun auch wieder nicht, wenn man nicht gerade mit den verschiedenen Zahlensystemen auf Kriegsfuß steht.

Man kann aber durch die Beschäftigung mit den Programmbarcodes einiges über den Aufbau der HP41 Programme erfahren, die Barcodes sind nämlich das Spiegelbild der im Rechner gespeicherten Information, deren „Interpretation“ wir in der Anzeige in Form von Programmzeilen als Text oder Zahlen sehen können.

Es gibt im Prinzip zwei Möglichkeiten der Programmbarcodeerzeugung, da es beim HP41 inzwischen zwei Möglichkeiten der Übertragung von Programmen zu anderen Geräten gibt:

- 1) Man tippt das Listing in den Rechner ein, der den Barcode drucken soll oder man druckt das Programm auf diesen Rechner aus, indem man ein HP-IL Interface (seriell oder parallel) als Drucker deklariert, an dessen Schnittstelle der „größere Rechner“ hängt und diesen Text zur Weiterverarbeitung abspeichert.

In beiden Fällen liegt die zu Barcode zu verarbeitende Programminformation als Text vor, die in Form einer Vergleichstabelle in die Reihen- und Spalteninformation umgerechnet werden muß, da dies für die Generierung des Barcodes Voraussetzung ist, ihr werdet schon sehen.

- 2) Man überträgt das in Barcode zu verwandelnde Programm, GTO .. (PACKING) vorher nicht vergessen, damit das Programm mit END aufhört, mit Hilfe eines HP-IL Konverters mit paralleler oder serieller Schnittstelle zu dem den Barcode druckenden Rechner, indem man den Befehl OUPP (OUT Programm) auf dem HP41 aufruft (eXtendet I/O-Modul).

Das im ALPHA-Register benannte Programm wird dabei als Folge von Hexadezimalcodes ausgesandt, die man wirklich lesen kann:

Der Befehl INT wird rechnerintern als Byte 0110 1000<sub>2</sub> oder 68<sub>16</sub> gespeichert, d.h. OUPP sendet die beiden Textzahlen „68“ aus, für den Befehl ATAN wird dementsprechend „5E“ gesendet.

Man muß also auch hier eine Umwandlung der Programminformation in das zu verarbeitende Byte vornehmen, was selbst dem ungeübten Programmierer keine Schwierigkeit bereiten sollte.

Die erste Methode hat einen nicht unwesentlichen Nachteil: In der Tabelle, in der alle Befehle zum Vergleichen und Speichern der Reihen- und Spalteninformation enthalten sind müssen natürlich auch alle zu erwartenden XROM-Funktionen als Klartext mit den erforderlichen beiden Nummern, der ID und der laufenden Funktionsnummer innerhalb des Moduls, enthalten sein.

Wer kann aber schon alle noch auf den Markt kommenden Module? Bei der zweiten Methode der Datenübertragung erübrigt sich dieses Problem, der Befehl OUPP schickt schon die für den Barcode erforderliche Information, damit ist man also für alle Zeiten gewappnet.

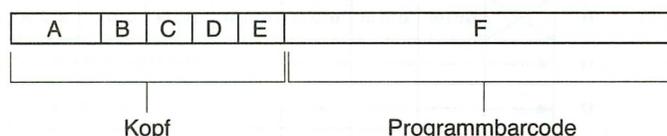
### Vorsicht:

Beim Einlesen der Barcodes oder dem Betrieb eines Programms mit Befehlen aus Modulen ist unbedingt darauf zu achten, daß auch das benötigte Modul im Rechner steckt und nicht eines mit derselben ID, da dann die Funktionen dieses Moduls aufgerufen werden, NON-EXISTENT gibts nur bei fehlendem Modul mit der geforderten ID.

Für die Modul-ID, d.h. die Identifikationsnummer stehen nur 5 Bit zur Verfügung, mehr als 32 verschiedene Nummern sind also nicht möglich. Es gibt aber inzwischen mehr als 32 Module, dementsprechend mehrere doppelt, also aufpassen!

Wer bis jetzt noch nicht abgegeben hat, darf sich jetzt freuen, es wird konkret:

### Aufbau einer HP41 Programmbarcodezeile



Kopf = 24 Bit = 3 Byte:

- A 8 Bit Prüfsumme aus B+C+D+E+F mit jeweils dazuaddiertem Überlauf
- B 4 Bit Typ-Indikator: 0001<sub>2</sub>=1 für normal und 0010<sub>2</sub>=2 für PRIVATE
- C 4 Bit fortlaufende Barcodezeilennummer
- D 4 Bit für die Anzahl der aus der vorhergehenden Barcodezeile übriggebliebenen Bytes einer Mehrbytefunktion
- E 4 Bit für die Anzahl der in dieser Barcodezeile übrigbleibenden Bytes einer Mehrbytefunktion, die nicht mehr in die Zeile paßte
- F bis zu 13 Programmbytes, wie sie im Rechner im Programmspeicher stehen müßten

Diese kurzen Informationen über den Aufbau einer Barcodezeile werden den meisten nicht viel weiter helfen, wir müssen uns so wieso noch einmal die Tabelle der HP41-Funktionen anschauen, sie sei der Vollständigkeit halber noch einmal abgebildet.

Diese Tabelle enthält einige Tippfehler:

FRAC heißt auf dem HP41 FRC,  
CL müßte CLΣ heißen,  
F?C heißt in Wirklichkeit FS?C.

Wie man am rechten Rand erkennen kann ist die Tabelle in verschiedene Gruppen eingeteilt; ein Befehl kann aus einem, aus zwei oder drei Byte und bei Text aus bis zu 15 Byte bestehen.

Der jeweilige Code des Befehls ergibt sich aus dessen Position in der Tabelle, indem man Reihe\*16+Spalte berechnet:

$$\text{AOFF} = 8 \cdot 16 + 11 = 139 = 8B_{16}$$

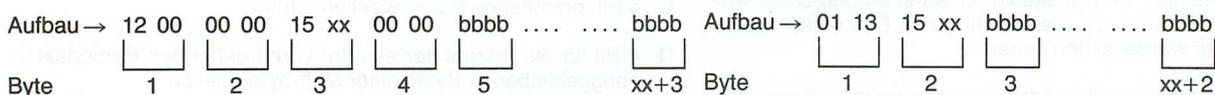
Daß in diesen Gruppen durchaus auch Ausnahmen zu finden sind bestätigt ja bekanntlich jede Regel, darum fangen wir gleich damit an:

### LBL “

Zu allererst der globale ALPHA-Label, zu finden in der Function Table in Reihe 12, Spalte 0-13. Diese Funktion besteht normalerweise aus 5 Byte, nämlich 4 Byte Vorspann und mindestens einem Buchstaben. Ich sagte deshalb normalerweise 5 Byte, da man ja auch den einen Buchstaben weglassen kann, dies kann aber über die Tastatur nicht eingegeben werden.

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

**HP41 Function Table (Quelle: Hewlett Packard)**

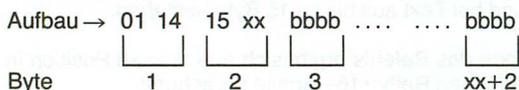


**Byte**

- 1 Position in der Funktionstabelle
- 2 Information über Position in der ALPHA-Labelkette im Speicher, kann 0 gesetzt werden
- 3 die obersten 4 Bit sind 1, die untersten 4 Bit sind die Anzahl der im Label vorkommenden Buchstaben + 1
- 4 Reserviert für Tastenzuweisung des Labels, ist bei nicht benutzter Zuweisung 0
- 5 1. Buchstabe des ALPHA-Labels, kann natürlich auch wegfallen, wenn xx=1 gesetzt wird!! (nicht über Tastatur möglich)

**XEQ“**

Der Aufbau ist dem des Labels sehr ähnlich, man benötigt nur die Verkettungsinformation und die Tastenzuweisung nicht, dementsprechend ist der Befehl ein wenig kürzer:



**Byte**

- 1 Position in der Funktionstabelle
- 2 die obersten 4 Bit sind 1, die untersten 4 Bit sind die Anzahl der in XEQ vorkommenden Buchstaben
- 3 1. Buchstabe des ALPHA-XEQs, kann natürlich auch wegfallen, wenn xx=0 gesetzt wird!! (nicht über Tastatur möglich). Damit kann ein Unterprogramm mit Label ohne Buchstaben aufgerufen werden ...

**GTO“**

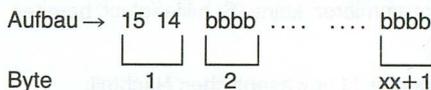
Der Sprung zu einem ALPHA-Label ist genauso aufgebaut wie der Abruf eines Unterprogramms mit ALPHA-Label:

**Byte**

- 1 Position in der Funktionstabelle
- 2 die obersten 4 Bit sind 1, die untersten 4 Bit sind die Anzahl der in GTO vorkommenden Buchstaben
- 3 1. Buchstabe des ALPHA-GTOs, kann natürlich auch wegfallen, wenn xx=0 gesetzt wird!! (nicht über Tastatur möglich). Damit kann der Label ohne Buchstaben angesprungen werden ...

**Text**

Der Aufbau von Texten ist besonders einfach, es steht dem Text lediglich das Byte mit der Reiheninformation, die obersten 4 Bit = 15, voran, gefolgt von der Anzahl der folgenden ALPHA-Zeichen:



**Byte**

- 1 die obersten 4 Bit sind 1, die untersten 4 Bit sind die Anzahl der im Text vorkommenden Buchstaben
- 2 1. Buchstabe des Textes, kann natürlich auch wegfallen, wenn xx=0 gesetzt wird!! (nicht über Tastatur möglich). Damit kann man einen Text ohne Buchstaben kreieren, d.h. man löscht lediglich das ALPHA-Register!

**Zwei-Byte-Funktionen**

Unter Zwei-Byte-Funktionen finden wir Funktionen, die irgend etwas adressieren oder einfach neben dem eigentlichen Befehl eine Adresse oder einen Label ansprechen:

Speicherbezogene Funktionen →

RCL, STO, ST+, ST-, ST\*, ST/, ISG, DSE, VIEW, ΣREG, ASTO, ARCL, TONE, X<>, LBL

Aufbau → 09 xx nn  
 Byte 1 2

Byte  
 1 09 ist die Reihe in der Funktionstabelle, xx die Spalte (bei X<> und LBL ist es 12)  
 2 zu adressierendes Register im Speicher  
 0 <= nn <= 127

Ein Byte hat bekanntlich 8 Bit, damit läßt sich nach Adam Riese der Zahlenbereich von 0 bis 255 darstellen. Setzt man das höchste Bit in Byte 2 auf 1, so hat man damit die indirekte Adressierung gewählt, d.h. statt STO 41 steht STO IND 41 in der Anzeige.

Bei TONE ergeben sich dieselben Anzeigen wie z.B. bei STO, es werden dabei jeweils verschiedene Tonarten erzeugt.

Funktionen mit einstelligem Argument →

FIX, SCI, ENG

Aufbau → 09 xx nn  
 Byte 1 2

Byte  
 1 09 ist die Reihe in der Funktionstabelle, xx die Spalte  
 2 Argument  
 0 <= nn <= 9 für höchstes Bit = 0, also direktes Argument  
 128 <= nn <= 255 erzeugt die indirekte Adressierung (höchstes Bit = 1)

XROM →

Aufbau → 1010 0fff gghh hhhh (Zahlen zur Basis 2!!)  
 Byte 1 2

fff gg 5 Bit ROM-ID  
 hhhh hhh 6 Bit laufende Funktionsnummer im Modul

In der Beschreibung eines ROM steht irgendwo, welche „XROM-Nummer“ eine bestimmte Funktion des Moduls hat, z.B. WNDSCN hat XROM 27,05; d.h. der Barcodeleser hat ROM-ID 27, die Funktion WNDSCN hat die Funktionsnummer 5. Die obersten 4 Bit des 1. Byte sind die Reihe der Funktionstabelle, nämlich 10, das höchste der unteren 4 Bit ist 0.

Flags →

SF, CF, FS?C, FC?C, FS?, FC?

Aufbau → 10 xx nn  
 Byte 1 2

Byte  
 1 10 ist die Reihe in der Funktionstabelle, xx die Spalte  
 2 zu adressierender Flag  
 0 <= nn >= 29 für höchstes Bit = 0, also direktes Argument  
 128 <= nn <= 255 erzeugt die indirekte Adressierung (höchstes Bit = 1)

Sprünge →

GTO 01 bis GTO 14

Aufbau → 11 nn 00 00  
 Byte 1 2

Byte  
 1 die obersten 4 Bit sind die Reihe in der Funktionstabelle, nn ist die Spalte

2 enthält die Sprungweite, kann 0 gesetzt werden; die Berechnung erfolgt normalerweise erst, wenn das Programm das erste Mal diesen Befehl erreicht.

GTO IND und XEQ IND

Aufbau → 10 14 fx xx  
 Byte 1 2

Byte  
 1 Position der Funktion in der Funktionstabelle  
 2 f=0 für GTO IND (f ist das höchste Bit)  
 f=1 für XEQ IND  
 xxx = Nummer des anzuspringenden Labels

Über die Tastatur des HP41 kann man Register und Sprungadressen bis maximal 99 eingeben, es gibt aber noch weitere 27 Möglichkeiten, von denen einige sogar legal eingebbar sind, nämlich z.B. lokale Label A-J und a-e; der Rest fällt unter das Sammelkapital Synthetik.

So, nun die Bedeutung der über 99 hinausgehenden Argumente:

Argument	Anzeige	Drucker
100	100	
101	101	
102	A	
103	B	
104	C	
105	D	
106	E	
107	F	
108	G	
109	H	
110	I	
111	J	
112	T	
113	Z	
114	Y	
115	X	
116	L	
117	[	
118	\	M
119	]	N
120	↑	O
121	—	P
122	T	Q
123	a	R
124	b	
125	c	
126	d	
127	e	

Drei-Byte-Funktionen

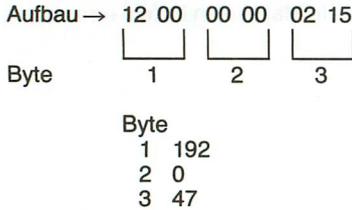
Hier gibt es eigentlich nur GTO und XEQ nach Labels mit maximal 128 Byte Entfernung im Speicher. Die Entfernung zum gewünschten Label ist in Byte 2 enthalten, dessen höchstes Bit die Sprungrichtung angibt, 0 für vorwärts und 1 für rückwärts.

Aufbau → zz 00 00 00 xx xx  
 Byte 1 2 3

Byte  
 1 die obersten 4 Bit zz sind die Reihe in der Funktionstabelle  
 2 hier wird beim ersten Auftreten im Programm die Sprungweite abgelegt  
 3 Sprungziel

Zu guter Letzt muß an jedem Programmbarcode das END hängen, damit der Barcodeleser automatisch erkennt, wann er mit der Arbeit aufhören kann. Hat man vor der Übertragung des zu druckenden Programms mit OUPP schön brav GTO . . (PACKING) gemacht, so wird dies auch übertragen und man kann das Thema vergessen.

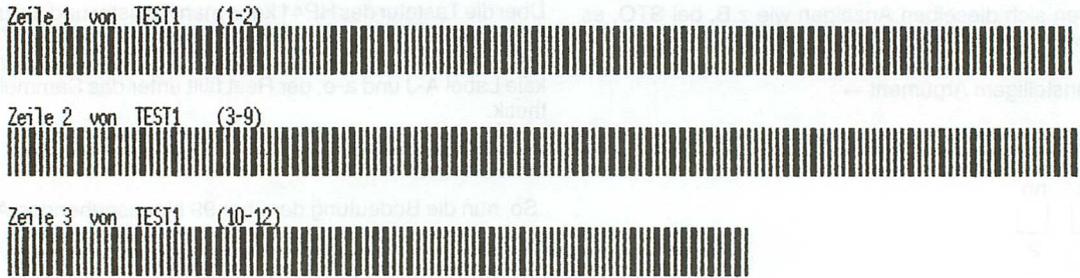
Ansonsten muß noch ein END an das Programm gehängt werden:



```

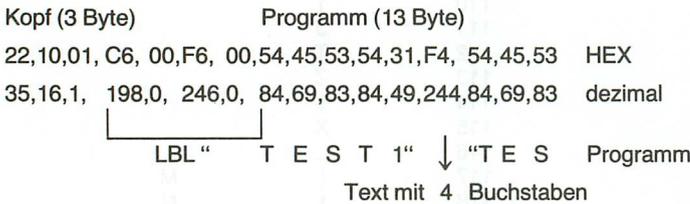
01*LBL "TEST1"      07 PRA
02 "TEST"           08 .012
03 AVIEW            09*LBL 01
04 PSE              10 ISG X
05 CLD              11 GTO 01
06 FS? 55           12 END
    
```

So, nach der vielen grauen Theorie nun endlich einmal ein praktisches Beispiel:



Das Beispielprogramm ist zwar keines von ausgesprochener Funktionalität, es kommen aber einige wichtige Dinge darin vor, die das Verständnis für das Selbermachen verbessern sollen.

Erst einmal möchte ich die erste Barcodezeile im Klartext hinschreiben, sie enthält folgende 16 Bytes (niemals mehr, höchstens weniger):



Byte 1 Prüfsumme: dies ist die Summe aller 15 verbleibenden Byte dieser Barcodezeile, wobei bei der Addition die Summe den Wert 255 wegen der zur Verfügung stehenden 8 Bit nicht überschritten werden kann.

Für den Programmierer gilt dementsprechend folgende Berechnungsformel:

$$\text{Prüfsumme} = \text{Summe aller 15 Byte der Zeile} + \text{Prüfsumme der Vorzeile (in 1. Zeile 0)}$$

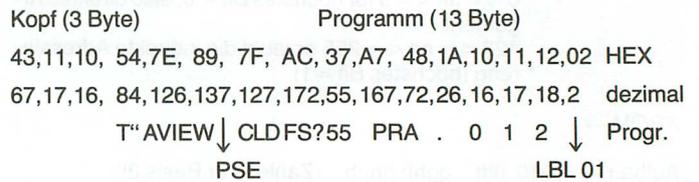
$$\text{Prüfsumme} = (\text{Prüfsumme MOD } 256) + \text{INT}(\text{Prüfsumme}/256)$$

2 Die obersten 4 Bit sind 0001<sub>2</sub>, d. h. kein PRIVATE  
Die untersten 4 Bit sind 0, das Zählen der Barcodezeilen beginnt bei Null. Ist die Zeile 15 erreicht, so beginnt man wieder bei 0

3 Die obersten 4 Bit sind 0, keine Reste von Mehrbytefunktionen aus der vorhergehenden Zeile, logisch, da gibt's keine.  
Die unteren 4 Bit sind 0001<sub>2</sub>, also 1, die Anzahl Bytes einer Mehrbytefunktion, die nicht ganz in diese Zeile passten. Damit ist der Text „TEST“ gemeint, dessen „T“, also 1 Byte, nicht mehr in diese Zeile passte.

4-16 Dies sind die ersten 13 Programmbyte unseres Testprogramms, jetzt kann jeder mal schauen, ob er die Definitionen der in dieser Zeile vorkommenden Befehle verstanden hat.  
Das erste Byte des ALPHA-Labels könnte für den Barcode auch CO<sub>16</sub> oder 192 heißen, die untersten 4 Bit spielen für den Barcode keine Rolle.

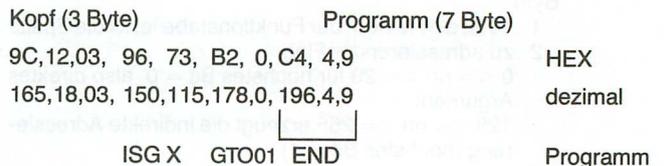
2. Barcodezeile:



- Byte
- Siehe 1. Zeile
  - Zeilennummer wurde auf 1 erhöht (die unteren 4 Bit)
  - 1 Byte aus der vorhergehenden Zeile (obere 4 Bit = 1), keine Mehrbytefunktion am Ende der Zeile abgebrochen, untere 4 Bit also 0.
  - 16 PRA ist XROM 29,08 (siehe Druckerhandbuch oder HP-IL Modulhandbuch)  
→ 1010 0111<sub>2</sub> 0100 1000<sub>2</sub>
- |      |    |   |   |
|------|----|---|---|
| XROM | 29 | , | 8 |
|------|----|---|---|

Zahlen werden genauso kodiert, wie man sie über die Tastatur eingibt, wie man leicht erkennen kann.

3. Barcodezeile:



- Byte
- Siehe 1. Zeile
  - Zeilennummer wurde auf 2 erhöht (die unteren 4 Bit)
  - Die END-Anweisung wurde vom Rechner so kodiert, die in der Definition stehende Kodierung 192,0,47 geht aber genauso.

So, das wär's für dieses Mal, das nächste Mal kommt hoffentlich der Rest zum Thema Barcodes auf dem HP414.

Es fehlen noch die Barcodes für

- 1) Direkt ausführbarer Barcode
- 2) Numerischer Datenbarcode
- 3) ALPHA Datenbarcode
- 4) Aufeinanderfolgender Datenbarcode

Zum Thema Barcodes ist im Helder mann Verlag ein Buch von Konrad Albers zum Thema „HP-41 Barcodes mit dem HP-IL-System“ erschienen, wenn mir die Zeit reicht gibt es dazu das nächste Mal eine Buchbesprechung, versprechen kann ich das jetzt noch nicht.

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

Happy Programming,

Martin Meyer (1000)  
Redaktion

## Dreiecksberechnung

409 Zeilen, 822 Bytes, 118 Regs., SIZE 026, HP-41CV, X-F/M, (ThinkJet)

Nachdem mein Programm TRI, PRISMA 6/87, unerwartet reges Interesse fand, wurde es noch einmal überarbeitet.

In Zusammenarbeit mit Herrn Dr. Hans Berghaus (CCD 1845) wurde dem vorherrschenden Wunsch nachgekommen, das Programm zu kürzen. Durch indirekte Steuerung und kompakte Routinen (hier ist Herr Berghaus spitze) konnte das Programm auf die Hälfte der ursprünglichen Länge schrumpfen, obwohl die Berechnung der Höhen in das Programm aufgenommen wurde und der Ein- und Ausgabekomfort erhalten blieb.

### Programm

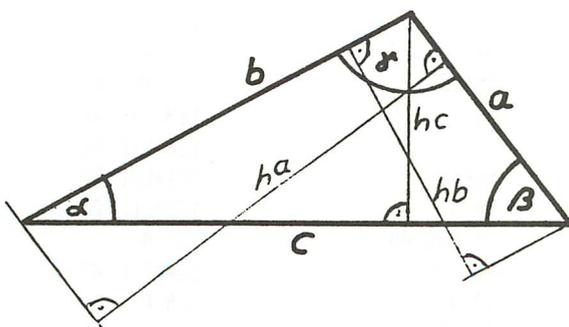
Das Programm TRI1 entspricht in der Grundform dem Programm TRI; ich werde mich deshalb auf die Programmanwendung beschränken.

Folgende Forderungen werden erfüllt:

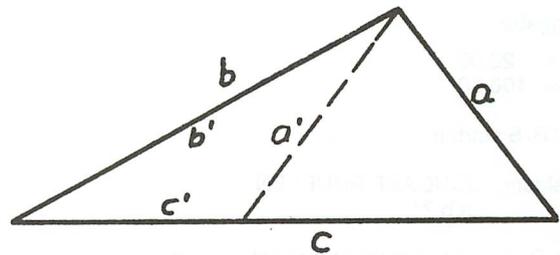
1. Berechnung der Seiten und Winkel und Ausgabe der Ergebnisse mittels R/S.
2. Berechnung der Flächen und Höhen und Ausgabe der Ergebnisse mittels LBL „FH“ (Fläche, Höhe)
3. Ausgabe aller Seiten und Winkel, auch der eingegebenen Daten mittels LBL D (Daten)
4. Eingabeüberwachung zur Trennung von Eingabe und Datenfehlern.
5. Meldung, wenn ein Dreieck nicht berechnet werden kann. s. Beispiel.
6. Dateneingabe:

Seite/Winkel	=	LBL	Taste
a	=	A	A
b	=	B	B
c	=	C	C
$\alpha$	=	F	F
$\beta$	=	G	G
$\delta$	=	H	H

### Beispiele:



- |               |        |
|---------------|--------|
| 1. Eingabe:   | Taste: |
| a = 20        | A      |
| b = 100       | B      |
| $\alpha$ = 10 | F      |



- a) Programm starten
- b) Die Meldung: „DATEN?“ fordert zur Eingabe auf.
- c) Nach der Dateneingabe mit R/S das Hauptprogramm starten.
- d) Meldung: „2 LOESUNGEN“ Es sind 2 Dreiecke möglich.
- e) Erneut mit R/S starten.
- f) Ergebnis:

c = 108,40  
R/S BE = 60,25  
R/S GA = 109,75

R/S c' = 88,56  
R/S BE' = 119,75  
R/S GA' = 50,25

R/S c = 108,40  
usw. rundum

- g) Flächen und Höhen:  
mit LBL „FH“ starten

Tasten: Shift und D

- b) Ergebnis:

FL = 941,20  
R/S Ha = 94,12  
R/S Hb = 18,82  
R/S Hc = 17,36

R/S FL' = 768,90  
R/S Ha' = 76,89  
R/S Hb' = 15,38  
R/S Hc' = 17,36

R/S FL = 941,20  
usw. rundum

i) Ausgabe aller Seiten und Winkel:

mit LBL D

Taste: D

k) Ergebnis:

a = 20,00  
 R/S b = 100,00  
 R/S c = 108,40  
 R/S AL = 10,00  
 R/S BE = 60,25  
 R/S GA = 109,75

R/S a' = 20,00  
 R/S b' = 100,00  
 R/S c' = 88,56  
 R/S AL' = 10,00  
 R/S BE' = 119,75  
 R/S GA' = 50,25

R/S a = 20,00  
 usw. rundum.

2. Eingabeüberwachung:

a) Eingabe:

a = 20,00  
 b = 100,00

b) Mit R/S starten

c) Meldung: „EINGABE PRUEFEN“  
 „a b?“

3. Ein Dreieck kann nicht berechnet werden.

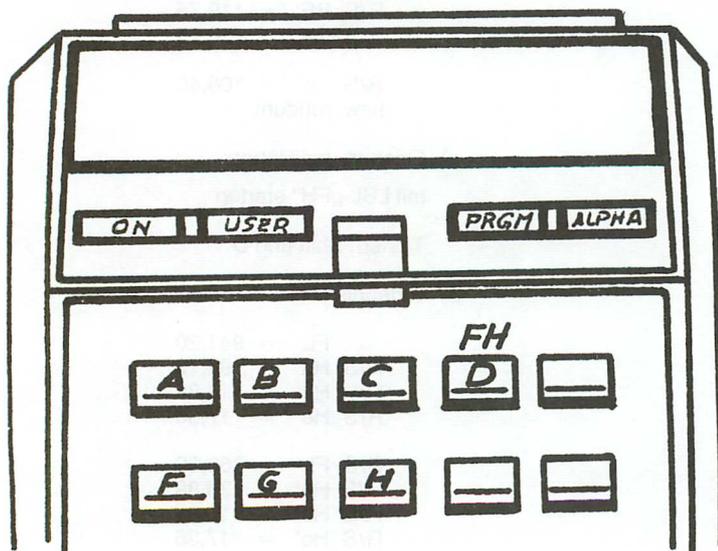
a) Eingabe:

a = 20,00  
 b = 100,00  
 = 30,00

b) Meldung: „KEINE LOESUNG:

4. Nach jeder Anzeige oder Meldung kann sofort eine neue Eingabe erfolgen. Es ist nicht notwendig das Programm TRI1 jeweils erneut aufzurufen. Das ermöglicht die zwischenzeitliche Zuweisung zusätzlicher Register, welche dann erhalten bleiben.

01*LBL "TRI1"	64*LBL G
02 CLA	65 5
03 26	66 GTO 00
04 PSIZE	67*LBL H
05 11	68 6
06 PASH	69*LBL 00
07 12	70 X<>Y
08 PASH	71 STO IND Y
09 13	72 CLA
10 PASH	73 ARCL X
11 21	74 2
12 PASH	75 RCL Z
13 22	76 Y↑X
14 PASH	77 ST- 25
15 23	78 GTO 02
16 PASH	79*LBL 03
17 14	80 CF 00
18 PASH	81 CF 09
19 "FH"	82 CF 10
20 CHS	83 126
21 PASH	84 X<> 25
22 " a"	85 STO 00
23 ASTO 14	86 X<>F
24 " b"	87 2
25 ASTO 15	88 ST/ 00
26 " c"	89 36
27 ASTO 16	90 ST+ 00
28 "AL"	91 SF 25
29 ASTO 17	92 CLD
30 "BE"	93 GTO IND 00
31 ASTO 18	94 GTO 17
32 "GA"	95*LBL 49
33 ASTO 19	96*LBL 57
34 "FL"	97*LBL 61
35 ASTO 20	98*LBL 73
36 "Ha"	99*LBL 77
37 ASTO 21	100*LBL 78
38 "Hb"	101 SF 09
39 ASTO 22	102 XEQ 09
40 "Hc"	103 GTO IND 00
41 ASTO 23	104*LBL 47
42 1.007006	105*LBL 55
43 STO 13	106*LBL 62
44 126	107*LBL 64
45 STO 25	108*LBL 71
46*LBL 19	109*LBL 86
47 "DATEN?"	110 SF 08
48*LBL 02	111 XEQ 10
49 TONE 5	112 GTO IND 00
50 PROMPT	113*LBL 49
51 GTO 03	114*LBL 55
52*LBL A	115*LBL 74
53 1	116 6
54 GTO 00	117 RCL 04
55*LBL B	118 RCL 05
56 2	119 GTO 00
57 GTO 00	120*LBL 50
58*LBL C	121*LBL 57
59 3	122*LBL 71
60 GTO 00	123 4
61*LBL F	124 RCL 05
62 4	125 RCL 06
63 GTO 00	126 GTO 00



127*LBL 47	190 RCL 01	253*LBL 05	316 GTO 08	379 RTN
128*LBL 58	191 RCL 02	254 5	317 GTO 15	380*LBL 17
129*LBL 73	192 X>Y?	255 STO 00	318*LBL 01	381 "EINGABE PRUEFEN"
130 5	193 SF 00	256 1.006	319 CLA	382 TONE 3
131 RCL 04	194 RCL 04	257 STO 24	320 ARCL IND 24	383 AVIEW
132 RCL 06	195 SIN	258*LBL 06	321 FS? 08	384 CLA
133*LBL 00	196 *	259 FC? IND 24	322 "+ "	385 1.006
134 XEQ 13	197 RCL 01	260 FS? 09	323 FC? 08	386*LBL 18
135 XEQ 14	198 /	261 GTO 01	324 "+ "	387 FS? IND X
136 LASTX	199 ASIN	262 GTO 00	325 "+ -"	388 GTO 00
137 RCL 05	200 XEQ 12	263*LBL 01	326 1 E2	389 RCL X
138 SIN	201 XEQ 14	264 RCL IND 24	327 X>Y?	390 13
139 *	202 FC? 00	265 13	328 "+ "	391 +
140 STO 02	203 GTO 16	266 ST+ 24	329 RDN	392 "+ "
141 GTO 16	204 RCL 13	267 X<>Y	330 1 E1	393 ARCL IND X
142*LBL 64	205 REGMOVE	268 XEQ 01	331 X>Y?	394 13
143*LBL 78	206 180	269 PROMPT	332 "+ "	395 -
144*LBL 85	207 RCL 05	270 13	333 RDN	396*LBL 00
145 RCL 04	208 -	271 ST- 24	334 ARCL X	397 ISG X
146 RCL 03	209 XEQ 12	272*LBL 00	335 RTN	398 GTO 18
147 P-R	210 XEQ 14	273 ISG 24	336*LBL 09	399 "+ ?"
148 RCL 02	211 SF 10	274 GTO 06	337 XEQ 10	400 TONE 6
149 -	212 XEQ 16	275 GTO 15	338*LBL 10	401 PROMPT
150 R-P	213 RCL 13	276*LBL "FH"	339 SF 14	402 GTO 19
151 STO 01	214 REGSWAP	277*LBL 07	340*LBL 11	403*LBL I
152 GTO 00	215*LBL 04	278 7	341 RCL 01	404 " a↑A, b↑B, c↑C"
153*LBL 92	216 CLA	279 STO 00	342 X<> 02	405 " AL↑F, BE↑G,"
154 RCL 02	217 FC? 25	280 20.023	343 FS? 14	406 " GA↑H"
155 RCL 03	218 " KEINE "	281 STO 24	344 X<> 03	407 " DATEN ↑D"
156 R-P	219 FS? 25	282 RCL 06	345 STO 01	408 "FL,Ha,Hb,Hc ↑-D"
157 X↑2	220 "+ 2 "	283 SIN	346 RCL 04	409 END
158 RCL 01	221 "LÖSUNG"	284 RCL 02	347 X<> 05	
159 X↑2	222 FS? 25	285 RCL 05	348 FS?C 14	
160 -	223 "+EN"	286 SIN	349 X<> 06	
161 RCL 02	224 TONE 6	287 RCL 01	350 STO 04	
162 RCL 03	225 PROMPT	288 *	351 RTN	
163 *	226 FC? 25	289 LASTX	352*LBL 12	
164 ST+ X	227 GTO 19	290 R↑	353 FC? 25	
165 /	228 GTO 16	291 *	354 GTO 04	
166 ACOS	229*LBL 15	292 LASTX	355 STO 05	
167 FC? 25	230 FC? 00	293 R↑	356 6	
168 GTO 04	231 GTO IND 00	294 *	357 RCL 04	
169 STO 04	232 RCL 13	295 R↑	358 RCL 05	
170*LBL 00	233 REGSWAP	296 STO L	359*LBL 13	
171 RCL 03	234 FC?C 08	297 RDN	360 +	
172 RCL 02	235 SF 08	298 RCL 03	361 180	
173 RCL 04	236 GTO IND 00	299 ST* L	362 X<=Y?	
174 COS	237*LBL D	300 RDN	363 CF 25	
175 *	238 SF 09	301 2	364 FC? 25	
176 -	239 GTO 05	302 ST/ L	365 GTO 04	
177 RCL 01	240*LBL 16	303 X<> T	366 -	
178 /	241 FS? 07	304 X<> L	367 CHS	
179 ACOS	242 XEQ 11	305 XEQ 01	368 STO IND Y	
180 XEQ 12	243 FS? 09	306 PROMPT	369 RTN	
181 GTO 16	244 XEQ 10	307 LASTX	370*LBL 14	
182*LBL 61	245 FS? 08	308 RDN	371 RCL 06	
183*LBL 80	246 XEQ 09	309 RDN	372 SIN	
184*LBL 86	247 FS?C 10	310 ISG 24	373 RCL 01	
185 SF 07	248 RTN	311*LBL 08	374 RCL 04	
186 XEQ 11	249 CF 08	312 XEQ 01	375 SIN	
187*LBL 62	250 CF 09	313 PROMPT	376 /	
188*LBL 77	251 CF 25	314 RDN	377 *	
189*LBL 88	252 SF 07	315 ISG 24	378 STO 03	

Adolf Radas  
Prielmayerstraße 20  
8058 Erding

# Jacobi-Verfahren

370 Zeilen, 535 Bytes, 77 Regs., SIZE 020 + 2·h<sup>2</sup>, HP41C, MM, X-F/M, TIME, CCD-ROM, X-I/O, IL, DRIVE.

Das Problem der Eigenwerte bzw. der Eigenvektoren einer quadratischen Matrix  $\underline{A}$  der Dimension  $n \times n$  läßt sich folgendermaßen formulieren:

$$(I) \quad \underline{A} * \underline{x} = \lambda * \underline{x}$$

Hierbei werden der Zahlenwert  $\lambda$  „Eigenwert von  $\underline{A}$ “ und der Vektor  $\underline{x}$  „Eigenvektor zum Eigenwert  $\lambda$ “ genannt.

Wäre nun der Vektor  $\underline{x}$  gleich dem Nullvektor  $\underline{0}$ , so stellte sich dieses Problem erst gar nicht, da Gleichung (I) für alle  $\lambda \in \mathbb{C}$  erfüllt wäre. Dieser sogenannte „Trivialfall“ wird deshalb ausgenommen.

Mit der Voraussetzung  $\underline{x} \neq \underline{0}$  lassen sich dann die folgenden Umformungen durchführen:

$$(I) \rightarrow \underline{A} * \underline{x} = \lambda * \underline{E} * \underline{x} \rightarrow (\underline{A} - \lambda * \underline{E}) * \underline{x} = \underline{0}$$

$$\rightarrow (II) \quad \det(\underline{A} - \lambda * \underline{E}) = 0 \quad \underline{E}: \text{Einheitsmatrix}$$

Um die Lösung der Gleichung (II) zu erhalten, entwickelt man die Determinante von  $(\underline{A} - \lambda * \underline{E})$ , bis man schließlich ein Polynom n-ten Grades erhält, dessen Nullstellen die Eigenwerte von  $\underline{A}$  sind.

Eine Matrix der Dimension  $n \times n$  besitzt also stets  $n$  Eigenwerte aus den komplexen oder auch reellen Zahlen, die aber untereinander nicht unbedingt verschieden sein müssen.

Die Eigenvektoren zu einem bestimmten Eigenwert ergeben sich aus dem Ansatz

$$(III) \quad (\underline{A} - \lambda_i * \underline{E}) * \underline{x}_i = \underline{0}$$

wobei für  $\lambda_i$  der vorher errechnete Eigenwert eingesetzt wird. Das hier entstehende Gleichungssystem ist in jedem Fall unterbestimmt, sodaß es zu jedem Eigenwert  $\lambda_i$  der Matrix  $\underline{A}$  eine Menge von Vektoren gibt, die die obige Gleichung erfüllen. Im Fall von untereinander gleichen Eigenwerten sind dies mehrdimensionale Lösungsräume; sind alle Eigenwerte untereinander verschieden, so entstehen zu einem einzelnen Eigenwert lediglich Vielfache ein und desselben Eigenvektors.

Zu jeder Matrix  $\underline{A}$  existiert also eine Basis von  $n$  Eigenvektoren.

Insoweit ist das Problem also eindeutig lösbar. Die numerische Auswertung birgt allerdings noch einige Schwierigkeiten in sich:

Abgesehen davon, daß es rein algebraisch unmöglich ist, ein Polynom  $n$ -ten ( $n > 4$ ) Grades zu lösen, ist die dem Rechner eigene begrenzte Stellengenauigkeit oft Ursache von verheerend ungenauen Endergebnissen. Dies liegt auch daran, daß mehrere iterative Verfahren, die jeweils mit Fehlern behaftet sind, hintereinander ausgeführt werden. Oft hat das Resultat einer derartigen Berechnung nicht einmal das Vorzeichen mit dem wahren Ergebnis gemein.

Einen anderen, sichereren Weg beschreitet das sogenannte Jacobi-Verfahren. Hier wird die Matrix  $\underline{A}$  iterativ einer Diagonalmatrix nähergebracht. Dies geschieht durch eine Reihe von Ähnlichkeitstransformationen der Form

$$(IV) \quad \underline{A}^{(i+1)} = \underline{R}^{(i)T} * \underline{A}^{(i)} * \underline{R}^{(i)}$$

sogenannten Jacobi-Drehungen. Aus diesem Grund ist das Verfahren auch nur für reell-symmetrische Matrizen geeignet. Da aber die Mehrzahl der in der Praxis auftretenden Matrizen symmetrisch sind, stellt dies keine wirkliche Einschränkung dar.

Durch eine solche, oben beschriebene Jacobi-Drehung wird jeweils ein Nicht-Diagonalelement und dessen symmetrisches Gegenüber zum Verschwinden gebracht. Versucht man nun, durch eine weitere, geeignete Drehung ein weiteres Nicht-Diagonalelement zu Null zu machen, nehmen die ersten beiden Elemente zwar wieder einen Wert  $a_{pq} = a_{qp} \neq 0$  an; es zeigt sich jedoch, daß die Matrix  $\underline{A}$  bei konsequenter Anwendung des Jacobi-Verfahrens gegen eine Diagonalmatrix der Form

$$\lim_{m \rightarrow \infty} \underline{A} = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$$

konvergiert. Die Abbruchbedingung der Iteration ist erfüllt, wenn die Wurzel der aufsummierten Quadrate der oberen Nicht-Diagonalelemente einen vorgegebenen Genauigkeitswert (z.B. 1·E-20) unterschreitet.

In diesem Endzustand brauchen dann die Eigenwerte der Matrix  $\underline{A}$  nur noch aus der Hauptdiagonalen abgelesen zu werden.

Für die Berechnung der Eigenvektorbasis existiert ein äquivalentes Verfahren, das der folgenden, etwas mehr mathematisch gehaltenen Beschreibung des Verfahrens, entnommen werden kann.

Es sei  $\underline{A}$  eine quadratische, symmetrische Matrix der Dimension  $n \times n$ .

## JACOBI-VERFAHREN

Startwerte:

$$\underline{R}^{(0)} = \underline{A} \quad \underline{V}^{(0)} = \underline{E}$$

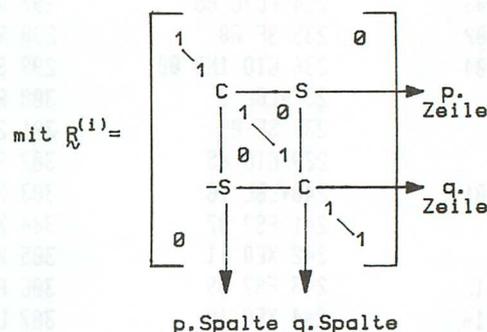
wobei  $\underline{V}^{(i)}$  die jeweils iterativ geänderte Orthonormalbasis der Eigenvektoren von  $\underline{A}$  und  $\underline{E}$  die Einheitsmatrix ist.

$$\underline{E} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$$

Für den  $i$ -ten Iterationsschritt gilt dann:

$$\underline{R}^{(i)} = \underline{R}^{(i)T} * \underline{R}^{(i-1)} * \underline{R}^{(i)} \quad \text{und}$$

$$\underline{V}^{(i)} = \underline{V}^{(i-1)} * \underline{R}^{(i)}$$



Die Parameter  $c$  und  $s$  errechnen sich aus:

$$0 = 0,5 * (a_{qq} - a_{pp}) / a_{pq} \quad (\text{falls } a_{pq} = 0 \text{ sein sollte: neues } p, q)$$

$$t = 1 / (0 + \text{sgn}(0) \sqrt{0^2 + 1}) \quad \text{mit } \text{sgn}(0) = 1 \quad (\text{HP-Standard})$$

$$c = 1 / \sqrt{1 + t^2}$$

$$s = t * c$$

für  $p =$  bis  $n-1$ , wobei  $q$  jeweils von  $p+1$  bis  $n$  läuft.



# SERIE 40

01+LBL "JACOBI"	64+LBL 15	127 LASTX	190 RCL 00	253 STO 06
02 20	65 CLX	128 X↑2	191 -	254 RCL 00
03 STO 01	66 STO IND 03	129 RCL 16	192 STO 18	255 RCL 16
04 STO 05	67 SIGN	130 +	193 RCL 00	256 +
05 1 E-20	68 ST+ 03	131 SQRT	194 STO 09	257 STO 02
06 STO 13	69 DSE 06	132 *	195+LBL 05	258 2
07 "DIM ?"	70 GTO 15	133 RCL 14	196 RCL 00	259 -
08 PROMPT	71+LBL 12	134 +	197 ST+ 17	260 STO 04
09 STO 00	72 SIGN	135 1/X	198 ST+ 18	261+LBL 09
10 STO 15	73 STO IND 04	136 ENTER↑	199 RCL IND 17	262 RCL 16
11 STO 02	74 RCL 10	137 X↑2	200 STO 08	263 ST+ 03
12 STO 03	75 ST+ 04	138 RCL 16	201 RCL 05	264 RCL 03
13 1	76 DSE 05	139 +	202 ST* IND 17	265 STO 05
14 STO 16	77 GTO 12	140 SQRT	203 RCL IND 18	266+LBL 10
15 +	78+LBL 07	141 1/X	204 RCL 10	267 RCL 04
16 STO 04	79 RCL 00	142 STO 05	205 *	268 RCL 02
17 CF 29	80 2	143 *	206 ST- IND 17	269 *
18 "a"	81 -	144 STO 10	207 RCL 05	270 RCL 01
19 ASTO 06	82 1 E3	145 RCL 11	208 ST* IND 18	271 +
20 "·"	83 /	146 RCL 07	209 RCL 08	272 RCL 02
21 ASTO 07	84 STO 07	147 -	210 RCL 10	273 -
22 "= ?"	85+LBL 02	148 RCL 00	211 *	274 RCL 05
23 ASTO 08	86 RCL 07	149 +	212 ST+ IND 18	275 +
24+LBL 00	87 1.001	150 STO 17	213 RCL 17	276 RCL IND X
25 RCL 00	88 +	151 RCL 01	214 RCL 00	277 X↑2
26 STO 02	89 STO 06	152 RCL 06	215 X↑2	278 ST+ 06
27+LBL 01	90+LBL 03	153 RCL 00	216 +	279 DSE 05
28 FIX 0	91 RCL 15	154 *	217 STO 02	280 GTO 10
29 CLA	92 STO 09	155 +	218 LASTX	281 DSE 04
30 ARCL 06	93 RCL 01	156 RCL 00	219 RCL 18	282 GTO 09
31 RCL 04	94 RCL 07	157 +	220 +	283 RCL 06
32 RCL 03	95 RCL 00	158 STO 18	221 STO 03	284 SQRT
33 -	96 *	159 RCL 00	222 RCL IND 02	285 ARCL X
34 ARCL X	97 +	160 STO 09	223 STO 08	286 AVIEW
35 ARCL 07	98 STO 11	161+LBL 04	224 RCL 05	287 RCL 13
36 RCL 04	99 RCL 06	162 RCL 16	225 ST* IND 02	288 X<=Y?
37 RCL 02	100 +	163 ST- 17	226 RCL IND 03	289 GTO 07
38 -	101 STO 12	164 ST- 18	227 RCL 10	290 BEEP
39 ARCL X	102 RCL IND 12	165 RCL IND 17	228 *	291 BEEP
40 ARCL 08	103 ABS	166 STO 08	229 ST- IND 02	292 BEEP
41 PROMPT	104 RCL 13	167 RCL 05	230 RCL 05	293+LBL 11
42 STO IND 05	105 RCL 00	168 ST* IND 17	231 ST* IND 03	294 RCL 00
43 ISG 05	106 /	169 RCL IND 18	232 RCL 08	295 STO 09
44 ADV	107 X>Y?	170 RCL 10	233 RCL 10	296+LBL 08
45 DSE 02	108 GTO 06	171 *	234 *	297 FIX 0
46 GTO 01	109 RCL 07	172 ST- IND 17	235 ST+ IND 03	298 "E"
47 DSE 03	110 ST+ 11	173 RCL 05	236 DSE 09	299 RCL 00
48 GTO 00	111 RCL 01	174 ST* IND 18	237 GTO 05	300 RCL 09
49 TONE 9	112 RCL 06	175 RCL 08	238+LBL 06	301 -
50 CLX	113 RCL 00	176 RCL 10	239 ISG 06	302 RCL 16
51 STO 19	114 *	177 *	240 GTO 03	303 +
52 RCL 01	115 +	178 ST+ IND 18	241 ISG 07	304 ARCL X
53 RCL 00	116 RCL 06	179 DSE 09	242 GTO 02	305 "t: "
54 X↑2	117 +	180 GTO 04	243 FIX 0	306 RCL 01
55 STO 06	118 RCL IND X	181 RCL 01	244 RCL 16	307 RCL 00
56 +	119 RCL IND 11	182 RCL 07	245 ST+ 19	308 RCL 09
57 STO 04	120 -	183 +	246 CLA	309 -
58 STO 03	121 RCL IND 12	184 RCL 00	247 ARCL 19	310 +
59 RCL 00	122 /	185 -	248 "t: "	311 LASTX
60 STO 05	123 2	186 STO 17	249 FIX 5	312 RCL 00
61 1	124 /	187 RCL 01	250 CLX	313 *
62 +	125 STO 14	188 RCL 06	251 STO 03	314 +
63 STO 10	126 SIGN	189 +	252 STO 05	315 FIX 5

316 RCL IND X	329 RCL 00	342 ST- 05	355 -	368 GTO 13
317 ARCL X	330 X†2	343 RCL 16	356 ARCL X	369 GTO 11
318 FIX 9	331 ST+ 05	344 ST+ 05	357 FIX 5	370 END
319 PROMPT	332 RCL 16	345+LBL 14	358 "†="	
320 DSE 09	333 ST- 05	346 FIX 0	359 RCL IND 05	
321 GTO 08	334+LBL 13	347 RCL 04	360 ARCL X	
322 RCL 00	335 RCL 00	348 RCL 03	361 FIX 9	
323 STO 02	336 STO 03	349 -	362 PRA	
324 RCL 01	337 RCL 16	350 "EV"	363 RCL 00	
325 RCL 00	338 +	351 ARCL X	364 ST+ 05	
326 X†2	339 STO 04	352 "†:"	365 DSE 03`	
327 +	340 RCL 00	353 RCL 04	366 GTO 14	
328 STO 05	341 X†2	354 RCL 02	367 DSE 02	

Michael Schilli  
Daucherstr. 2  
8900 Augsburg

## Messen mit HP-IL

67 Zeilen, 182 Bytes, HP-41C, X-I/O, IL, PRINTER, Interface HP82164

### Meßuhr und Schiebelehre am HP41 über RS 232.

Die Meßgeräte für die Längenmeßtechnik sind heute vorzugsweise digitalisiert und verfügen dadurch über Datenausgänge mit der RS232-Schnittstelle. Da die Hersteller aber kein einheitliches Datenformat verwenden, ist der Einsatz eines Software-gesteuerten RS232-Interface zum HP-IL notwendig. Das HP 82164 HP-IL/RS232-Interface ist dafür ideal geeignet. Auf dem HP41 als Controller kann der Facharbeiter in einer Meßgruppe ohne eine Programmiersprache lernen zu müssen seine meßtechnisch-mathematischen Probleme selber elegant lösen.

Nachdem die Betriebsanleitung zum HP82164-Interface für den Laien zu ausführlich ist, will ich im folgenden versuchen das Wesentliche zum Einsatz obiger Geräte zu beschreiben. Es sollen folgende Meßzeuge verwendet werden:

Eine HELIOS-Schiebelehre mit einem Interface, in dem das Datenformat von mir auf

„1 Startbit, 7 ASCII Bit, 2 Stopbit, keine Parität und 4800 Baud“ eingestellt wurde.

Drei weitere Fabrikate haben ähnliche Elektroniken.

Eine KÄFER-Multifunktionsmeßuhr MFT 30 mit dem festen Datenformat „1 Startbit, 7 ASCII Bit, 2 Stopbit, keine Parität und 600 Baud.“ Die Meßuhr kann viele Funktionen ausführen, die in der Anleitung dokumentiert sind, hier aber nicht weiter besprochen werden brauchen.

Nach dem Einschalten der HP-IL / RS232 – Schnittstelle ist ein Senden und Empfangen auf den Leitungen 2 und 3 mit 9600 Baud, 8 Datenbits, 1 Stopbit und ohne Parität sofort möglich. Die Leitungen 4 und 20 liegen auf + 8,5 Volt und zeigen damit den angeschlossenen Meßgeräten Sende- und Empfangsbereitschaft an, soweit die Geräte diese Signale überhaupt benötigen. Die von mir verwendeten Geräte brauchen sie nicht, die übrigen Anschlüsse werden hier auch nicht benötigt. Wegen der unterschiedlichen Datenformate kam ich nicht umhin, mit dem Ex I/O Modul HP82183 die notwendigen An-

passungen des Datenformates per Software vorzunehmen. Dazu genügt es, aus der HP-IL / RS232 – Anleitung die „Remote Mode Instruktionen“ = Fernsteuerbefehle auf den Seiten 37 bis 40 und das Beispiel auf Seite 90 (Programmzeile 06 bis 15) sinngemäß zu benutzen.

Die Verbindungen der Leitungen 2 und 3 zwischen Interface und Meßgeräten erfolgt so, daß man die weniger als –3 Volt führende Leitung des Meßgeräte-Interfaces mit dem Pin 3 des HP-IL Interfaces verbindet und den Pin 2 vom IL-Interface mit dem dann noch freien Pin 2 oder 3 des Meßgerätes, falls das Meßgerät vom HP41 angesteuert werden soll.

In meinem Anwendungsfall sieht es wie folgt aus:

Die von 4 Schiebelehren-Herstellern gemeinsam verwendete Sylvac-Elektronik auf der Schiebelehre gibt nur ein Low-power-Signal ab, das durch ein spezielles Interface auf den genormten RS232-Pegel gebracht werden muß. Dieses Interface hat einen Fußschalteranschluß mit dem die jeweilige Datenübergabe auf den Signalpin laut Anschlußbild des RS232-Steckers veranlaßt wird. Weiterhin kann das Datenformat in gewissen Grenzen variiert werden.

Die Meßuhr kann zwar direkt über ihre Tastatur bedient werden, es ist aber besser sie vom HP41 aus fernzusteuern, weil nur dann der Meßwert ohne Erschütterung fehlerfrei gesendet wird. Im Fernsteuerbetrieb erwartet die Meßuhr auf Pin 3 einen Ein Byte-Befehl ohne LF und CR, sie quittiert diesen im fehlerfreien Betriebsfall als gleichen EinByte-Wert auf Pin 2 auch wieder ohne LF und CR. Mit den Befehlen „OUTXB“ und „INXB“ wickelt der HP41 diesen Befehlsaustausch ab. Die Datenbytes des Meßwertes folgen dem Quittungsbyte sofort ohne ein Trennzeichen und werden mit CR abgeschlossen.

Ein 2-poliger, handbetätigter Umschalter legt den jeweils benutzten RS232-Ausgang auf den HP82164-RS232-Eingang.

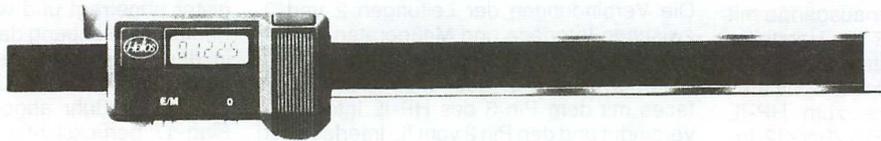
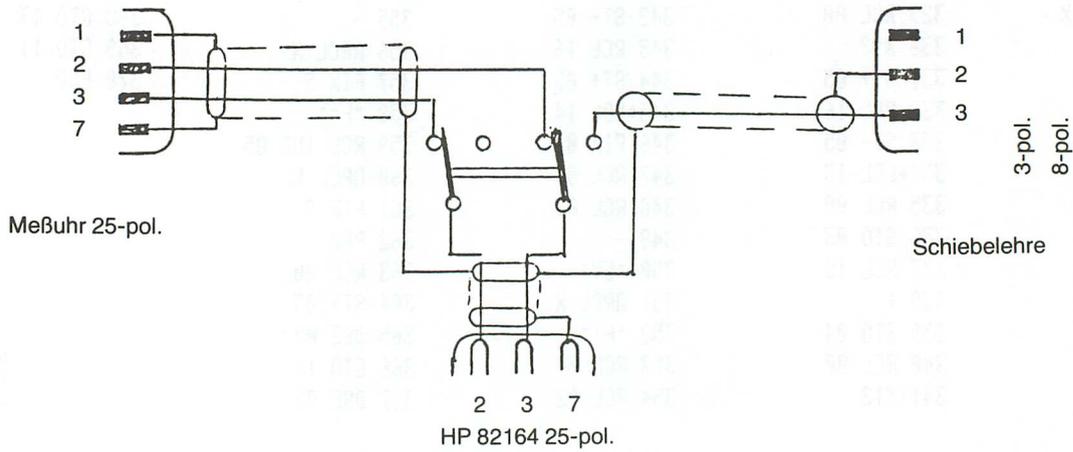
Nun zum Programm. Es erwartet auf dem IL-Platz 1 den Drucker und auf dem IL-Platz 2 das HP82164-Interface. Mit dem Label RS2 wird der Betrieb aufgenommen und das HP-Interface auf die Kommunikation mit der Meßuhr eingestellt. Da sich die beiden

Meßgeräte nur in der Baud-Rate unterscheiden, muß beim Wechsel der Meßzeuge nur die Baud-Rate und die Datenleitung umgeschaltet werden. Dazu dienen die Label „UHR“ bzw. „SCH“iebelehre. Beim Arbeiten mit der Schiebelehre wird mit dem Label „DAS“ (**D**aten **S**chiebelehre) das IL-Interface auf vorhandene Daten im Empfangsregister abgefragt und wenn diese nach der Fußtasterbetätigung da sind mit der Zeile 31 in das X-Register übernommen. Mit dem Label „DAU“ (**D**aten-**U**hr) wird ein Meßwert von der Meßuhr abgerufen. Das gesetzte Flag 17 berücksichtigt das Fehlen von LF und CR im Datenaustausch. In der Zeile 38 wird der Code – hier die Zahl 50 – für eine Meßwertübernahme bereitgestellt und mit Zeile 39 gesendet. Mit der Zeile 40 wird von der Meßuhr quittierte gleichwertige Steuerbefehl wieder aus dem IL-Empfangsregister entnommen und in der Zeile 41 mit dem gesendeten verglichen. Im Fehlerfall wird der Transferspeicher gelöscht (Zeile 43-46) und der Datenabruf wiederholt. Bei der richtigen Antwort der Meßuhr verzweigt das Programm zum Label 02 und holt den Meßwert ins X-Register. Weitere Meßwerte müssen mit dem erneuten Aufruf der Programme „DAS“ bzw. „DAU“ in den HP41 geholt werden. Da die Meßwerte-Verarbeitung individuell ist, enden die Musterprogramme mit RTN.

Da die Meßuhrfunktionen auch über die RS232-Schnittstelle eingestellt werden können, wurde der Programmteil „UHS“ (**U**hr **S**teuern) geschaffen. Mit dem Prompt von Zeile 57 kann der jeweilige Befehlscode als Dezimaläquivalent eingetippt und mit R/S abgeschlossen werden. Die Zeile 58 sendet den Befehl, die Zeile 59 holt die Quittung in den HP41 und der prüft in der Zeile 61 auf die richtige Ausführung und zeigt dieses in den Zeilen 63 bis 66 an.

Wenn zwischen den Meßwert-Übernahmen der Drucker benötigt wird, dann ist dieser mit „1 SELECT“ zu aktivieren. Alle Programmteile enthalten deshalb zu Beginn das „2 SELECT“. Die Programmteile werden vorzugsweise bestimmten Tasten im USER-Betrieb zugeordnet, was ja bekanntlich keinen zusätzlichen Speicherplatz benötigt.

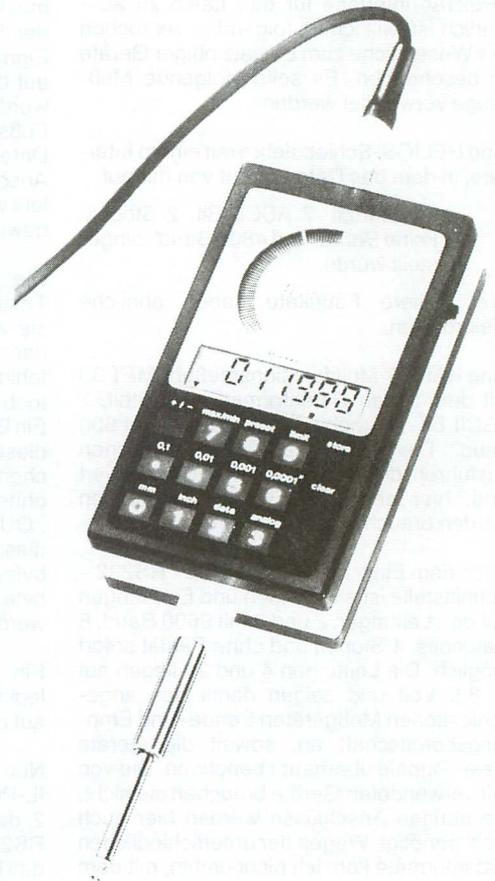
Die Synthetiker können die Zeilen 16 bis 18 als „SW1;SS1“ zusammenfassen, das Trennzeichen muß das Semikolon – Dez 59 – sein.



- |              |              |               |
|--------------|--------------|---------------|
| 01+LBL "RS2" | 31 IND       | 49 IND        |
| 02 MANIO     | 32 RTN       | 50 RTN        |
| 03 SF 10     | 33+LBL "DAU" | 51+LBL "UHS"  |
| 04+LBL "UHR" | 34 2         | 52 2          |
| 05 "SB7"     | 35 SELECT    | 53 SELECT     |
| 06 GTO 00    | 36+LBL 03    | 54 "BEFEHL ?" |
| 07+LBL "SCH" | 37 50        | 55 PROMPT     |
| 08 "SBC"     | 38 OUTXB     | 56 OUTXB      |
| 09+LBL 00    | 39 INXB      | 57 INXB       |
| 10 2         | 40 X=Y?      | 58 X=Y?       |
| 11 SELECT    | 41 GTO 02    | 59 GTO 04     |
| 12 REMOTE    | 42 REMOTE    | 60 "FEHLER"   |
| 13 OUTA      | 43 "R1"      | 61 PROMPT     |
| 14 FC?C 10   | 44 OUTA      | 62+LBL 04     |
| 15 GTO 00    | 45 NOTREM    | 63 "ERLEDIGT" |
| 16 "SW1"     | 46 GTO 03    | 64 PROMPT     |
| 17 OUTA      | 47+LBL 02    | 65 END        |
| 18 "SS1"     | 48 X<>Y      |               |
| 19 OUTA      |              |               |
| 20+LBL 00    |              |               |
| 21 NOTREM    |              |               |
| 22 RTN       |              |               |
| 23+LBL "DAS" |              |               |
| 24 2         |              |               |
| 25 SELECT    |              |               |
| 26+LBL 01    |              |               |
| 27 RDH       |              |               |
| 28 INSTAT    |              |               |
| 29 FC? 01    |              |               |
| 30 GTO 01    |              |               |

**Hinweis:**

dieses Programm wurde in letzter Minute noch korrigiert. Das Listing und die Barcodes entsprechen der aktuellsten Version. Möglicherweise sind dadurch kleine Abweichungen zur alten Programmbeschreibung entstanden, die nicht mehr geändert werden konnte.



# Tips & Utilities

## Zwei Werte zugleich anzeigen

Häufig kann es wünschenswert sein, zwei Werte gleichzeitig angezeigt zu erhalten (Koordinaten, Soll und Haben usw.). Die Routine „ZW“ („Zwei Werte“) erwartet die beiden Werte im Y- und X-Register, wobei die zuerst eingegebene Zahl (Y-Register) nachher auch zuerst erscheint.

Ist ein Drucker angeschlossen, werden die Werte mit einem Abstand gedruckt und dahinter der Wert des Registers 00 in Großdruck und ohne Komma angehängt. Das kann beim Buchhalter die Kontonummer, bei Koordinaten die Punktnummer sein. Der Anwender muß also dafür sorgen, daß Register 00 jeweils entsprechend vorher belegt ist. Selbstverständlich werden negative Werte richtig verarbeitet.

Die Zahlen werden vor Verarbeitung in dem FIX-Format gerundet, das der Benutzer vorgibt. Das ist nötig, weil der Rechner bei sehr kleinen Werten auf wissenschaftliche Schreibweise umschaltet und das in diesem Fall hier aber störend ist. 0,004 soll bei FIX 2 als 0,00 angezeigt werden. Vor dem PROMPT am Ende wird der Stack gelöscht. Somit kann der Benutzer zur weiteren Bearbeitung seiner Programme direkt Werte eingeben oder unmittelbar mit 0 starten. Bei Schritt 06 muß ein SPACE angehängt sein, sonst fehlt der Zwischenabstand zwischen den beiden angezeigten oder gedruckten Werten.

Läßt man das PROMPT am Schluß weg und baut die Routine in ein entsprechendes eigenes Programm ein, so kann man mit dem Druckerfortlaufende Verzeichnisse drucken. Einziger Schönheitsfehler: Bei Werten mit verschiedenen Stellenzahlen vor dem Komma stehen die Werte nicht korrekt untereinander.

Ist der Effekt beim Druck mit der Zahl aus Register 00 am Ende unerwünscht, so benutzt man die kleine Routine.

### ohne Nummer (Reg. 00)

```

01*LBL "ZW"
02 X<>Y
03 RND
04 CLA
05 ARCL X
06 "┌─"_____ Space!
07 RDN
08 RND
09 ARCL X
10 ACA
11 ADV
12 CLST
13 PROMPT
14 END
    
```

hier auch Großdruck  
(SF12) möglich

```

01*LBL "ZW"
02 CF 12
03 X<>Y
04 CLA
05 ARCL X
06 "┌─"_____ Space!
07 RDN
08 ARCL X
09 FC? 21
10 GTO 00
11 ACA
12 RCLFLAG
13 SF 12
14 CF 29
15 FIX 0
16 1
17 SKPCHR
18 RCL 00
19 ACX
20 RDN
21 RDN
22 STOF LAG
23 ADV
24*LBL 00
25 CLST
26 PROMPT
27 END
    
```

### Beispiele

```

25,36 59,87
-3,88 25,69
897,23 0,23
    
```

### Konstantenrechnung

Weil der HP 41 über normale Konstantenrechnung, wie sie jeder billige Kaufhausrechner kann, völlig erhaben ist, schrieb ich diese Routine. Die Konstante gibt man über XEQ „KO“ oder bei Wiederholungen mit der Taste A ein. Die individuellen Werte dann mit den Tasten B bis F. Bei B wird addiert, bei C subtrahiert, bei D multipliziert und bei E dividiert. Taste F ist die umgekehrte Division – es wird durch die Konstante dividiert. Die Konstante landet im Register 00. Nach dem erstmaligen Aufruf von B bis F kann jeweils mit R/S gestartet werden.

```

01*LBL "KO"
02*LBL A
03 STO 00
04 0
05 STOP
06*LBL B
07 RCL 00
08 +
09 STOP
10 GTO B
11*LBL C
12 CHS
13 RCL 00
14 +
15 STOP
16 GTO C
17*LBL D
18 RCL 00
19 *
20 STOP
21 GTO D
22*LBL E
23 RCL 00
24 /
25 1/X
26 STOP
27 GTO E
28*LBL F
29 RCL 00
30 /
31 STOP
32 GTO F
33 .END.
    
```

### Tachoeichung

1000 Meter müssen mit fliegendem Start exakt gestartet und gestoppt werden. Es wird nach dem Start die verstümmelte Uhrzeit und nach dem Stop die gefahrene Geschwindigkeit in km/h angezeigt. Wegen der Startgenauigkeit muß vorher das Programm mittels GTO „TA“ aufgerufen und dann mittels Taste A gestartet werden. Wegen der Start- und Stopungenauigkeiten ist es bei hohen Geschwindigkeiten ratsam, 2000 m zu messen und die dann angezeigte Geschwindigkeit zu verdoppeln.

```

01*LBL "TA"
02*LBL A
03 FIX 1
04 TIME
05 STOP
06 TIME
07 HMS-
08 CHS
09 HR
10 1/X
11 .END.
    
```

**Additionsstreifen**

Für einen gewöhnlichen Additionsstreifen ist der HP 41 nebst Drucker viel zu vornehm, zu wissenschaftlich. Mein „AD“-Programm liefert einen sauberen Streifen in Großdruck. Man beginnt immer mit Taste A – auch nach abgebrochenen Fehlversuchen. Der über die erste Zahl dabei gedruckte „\*“ zeigt den Nullstart auf dem Streifen an. Normale Additionen startet man mit R/S. Soll die zuletzt eingegebene Zahl wiederholt werden: Taste B. Negative Werte erhalten CHS. Multiplikationen gibt man so ein: 5 ENTER 6,07 Taste C. Taste D ergibt eine gedruckte Zwischensumme mit entsprechendem Zeichen. Sie wird auch laufend am Rechner angezeigt. Für die Endsumme: Taste E. Eine Tagessumme mit verstümmeltem Datum ergibt Taste F. Das setzt voraus, daß vor Beginn einmal Taste F gedrückt wird, um das Tagessummenregister zu nullen. Der Benutzer gibt sein FIX vor. Wenn in der FIX-0-Version das Komma hinter der letzten Ziffer stört: CF 29. Zwischendurch kann beliebig gerechnet und gelöscht werden – der gedruckte Streifen stimmt trotzdem, wenn die Register 00 bis 02 nicht berührt werden.

Register: 00 letzter eingegebener Wert, 01 Zwischenergebnis, 02

```

01 *LBL "AD"
02 *LBL A
03 SF 12
04 "*"
05 ACA
06 ADV
07 STO 00
08 STO 01
09 ACX
10 ADV
11 *LBL 00
12 STOP
13 *LBL 01
14 ACX
15 ADV
16 STO 00
17 *LBL 02
18 ST+ 01
19 RCL 01
20 GTO 00
21 *LBL B
22 RCL 00
23 GTO 01
24 *LBL C
25 RCLFLAG
26 CF 29
27 FIX 0
28 1
29 R↑
30 ACX
31 RDN
32 SKPCHR
33 "X"
34 ACA
35 RDN
36 STOFLAG
    
```

		*	
		25,01	
		36,56	R/S
		36,56	Taste B
		-3,66	CHS
	5	X 6,07	Taste C
Taste A	#	124,82	Zwischensumme (Taste D)
		3,00	R/S
	=	127,82	Taste E
		.	
		*	
		58,03	
		5,33	
		= 63,36	
		.	
		*	
	30,12	**	Taste F
		*	
	191,18	*	Tagessumme
		**	

```

37 RDN
38 STO 00
39 ACX
40 ADV
41 *
42 GTO 02
43 *LBL D
44 35
45 ACCHR
46 RCL 01
47 ACX
48 ADV
49 STOP
50 GTO 01
51 *LBL E
52 "="
53 ACA
54 RCL 01
55 ACX
56 ADV
57 ADV
58 ADV
59 ADV
60 ADV
61 ADV
62 ST+ 02
63 STOP
64 GTO A
65 *LBL F
66 DATE
67 PRX
68 0
69 X<> 02
70 PRX
71 ADV
72 ADV
73 ADV
74 ADV
75 ADV
76 .END.
    
```

schieben den Streifen abreifertig vor

**„Sackgasse“**

Bei greren Programmen stehen durchaus verschiedene Teile hintereinander, die nicht unbedingt mit einem RTN oder GTO ... abschlieen. Ein versehentlicher R/S-Start kann dann Programmteile starten lassen, die im Augenblick nicht erwnscht sind und deren Ablauf sogar betrchtliche Schden an den Daten anrichten kann.

Schreibt nach solchen Stops also ein GTO „SA“. Man landet in einer Sackgasse, die mit R/S nicht mehr verlassen werden kann. Ich benutze sie auch nach Durchfhrung von bestimmten Operationen und erhalte als Besttigung fr die Durchfhrung dann den „\*“ in der Anzeige.

```

01 *LBL "SA"
02 "*"
03 CLST
04 PROMPT
05 GTO "SA"
06 .END.
    
```

Jrgen Schatz  
Carl-Theodor-Strae 12  
6710 Frankenthal

**Tips & Tricks fr HP41 Benutzer**

1. Verklebte Tasten durch ein aus Versehen verschttetes Getrnk? Etwas Kontaktspray aus dem nchsten Elektronikladen hilft bestens.
2. Beim Aufstecken des Magnetkartenlesers achtet auf gute Einrastung. Vor dem Einschalten bitte nochmal wackeln. Ansonsten wartet nmlich MEMORY-LOST auf Euch!
3. Obwohl der BAT-Indikator noch keine Gefahr meldet, knnen beim Durchlaufen von Magnetkarten der Akku oder die Batterien „in die Knie gehen“. MEMORY-LOST und ein Steckenbleiben der Magnetkarte sind die Folge, evtl. auch ein Absturz des Rechners. Dummerweise kann auch der ganze erweiterte Speicher zerstrt werden. Selbst das Betreiben des Magnetkartenlesers mittels Ladegert „am Netz“ mu keine Abhilfe bringen, weil das Ladegert den hohen Strombedarf des Kartenlesers nicht bringt.

Abhilfe: Grundstzlich nur mit vollen Batterien oder frisch geladenem Akku mit Magnetkarten arbeiten und vor allem neben einer sauberen Dokumentation der Programme alles auf Magnetkarten speichern. Daten listet man per Drucker nach Schlu jeder Arbeitssitzung auf, denn nur das Aufzeichnen auf Magnetkarten kann mehrere Stunden Rechenarbeit wie oben geschildert zunichte machen.

4. Kleber für die Thermopapierstreifen? Reiner Papierkleber, wie PELIKANOL, Gutenberg-Gummierstift Nr. 70 G 72 (sog. Elefantenkleber) oder UHU-stic-Klebestift Nr. 45710. Der Streifen wird nach Jahren gleichmäßig dunkler, doch die genannten Kleber sind nicht die Urheber, denn sie waren nur punktuell aufgetragen.

Jürgen Schatz  
Carl-Theodor-Straße 12  
6710 Frankenthal

## Gesucht wird . . .

Es werden in Zukunft immer mehr Programme veröffentlicht, die Funktionen aus Erweiterungsmodulen des HP41 benutzen.

Um bei fremden Programmen nicht ständig auf unbekannte XROM-Nummern zu stoßen, die lakonische Meldung NONEXISTENT nach dem Drücken von R/S ist wohl jedem wohlbekannt, fordere ich alle mutigen Programmierer zum Schreiben eines nachfolgend beschriebenen Programms auf:

- 1) möglichst kurz,
- 2) möglichst ohne externe Module lauffähig (also synthetisch) oder nur mit dem CCD-ROM,

3) die eigentliche Funktion:

Ausgabe der von einem beliebigen Programm (eventuell auch im X-Funktionsbereich liegend) benötigten Einsteckmodule (nicht vergessen, der HP41CX hat TIME und X-Funktion Modul schon eingebaut) in kurzer, prägnanter Form, hier ist Gestaltungsfreiheit gegeben; es sollten die benötigten Module klar erkennbar sein, jeder lange Name frisst allerdings wieder viel Speicherplatz.

Martin Meyer (1000)  
Redaktion

## 20/40-stellige Gleitpunkt-Arithmetik

527 Zeilen, 851 Bytes, 120 Regs.,  
SIZE 016, HP-41CV

(Das X-F Modul wird nur für LBL „=“ benötigt)

Definition →

Um den Text nicht unnötig aufzublähen, werden die Registerbezeichnungen wie folgt abgekürzt:

- (x) steht für X-Register,
- (y) steht für Y-Register usw.

1 Subroutine-Level ist der Sprung durch GOSUB zu einem Unterprogramm, der HP41 kann bis zu 6 solcher Sprünge ausführen und sich dabei die Rückkehradresse merken, ohne auf ein RETURN zu stoßen. Bei Aufruf eines 7. GOSUB in Reihenfolge ohne dazwischenliegendes RETURN verißt der HP41 die Rückkehradresse des 1. GOSUB, d.h. er findet nicht mehr zum ersten Aufruf zurück (siehe hierzu Abschnitt 12 des Bedienungs- und Programmierhandbuchs für den HP41).

[Martin Meyer, Redaktion]

### 20-stellige Arithmetik

Die Programme „/+“ und „/\*“ verknüpfen zwei einfach genaue Operanden in (y) und (x) zu einem doppelt genauen Operanden in (x) und (y) zusammen, wobei in

- (x) die **ersten** 10 Stellen und in
- (y) die **letzten** 10 Stellen stehen.

Die Programme „-“ „+“ „\*\*“ „//“ verknüpfen zwei doppelt genaue Operanden in (z), (t) und (x), (y) zu einem doppelt genauen Ergebnis in (x) und (y).

Das Programm „SSQRT“ zieht die doppelt genaue Quadratwurzel aus dem doppelt genauen Operanden in (x) und (y), das Ergebnis steht dann wiederum in (x) und (y).

Die Programme können höhere als 20-stellige Genauigkeit liefern, da bei gesetztem Flag 00 in (y), (x) treu auf 20 Stellen gerundet wird. Nur für die Ausgaben interessiert die Vorzeichenangleichung für die Operandenteile durch „/“ in (y), (x).

Eine sauber formatierte Ausgabe des Operanden in (y), (x) liefert „=“, wobei der Operand im Stack zerstört wird!

→ Verwendung als Unterprogramm:

...; XEQ „=“; Verarbeitung der 1. Teilanzeige (dabei Belassen von SF 25, (x), (z)); PC: RTN; Verarbeitung der 2. Teilanzeige; ...

→ Verwendung durch Tastatur:

XEQ „=“ (ergibt 1. Teilanzeige); R/S (ergibt 2. Teilanzeige); ...

### Beispiele:

(3/7) \* 1234,67892468 = ?

Eingabe:

0 (Stellen 11-20 nicht vergessen!); ENTER ^; 3; ENTER ^; 0; ENTER ^; 7; XEQ „//“; 4,68E-6; ENTER ^; 12345,6789; XEQ „\*\*“;

- a) Weiterverarbeitung; ...
- b) (Zwischenergebnis betrachten:) XEQ „/“ (Anzeige: 5291,0052 je nach eingestelltem Format); x: y (Anzeige 862,86 -9); x: y; (Weiterverarbeitung)
- c) (Endergebnis:) XEQ „=“ („+5,2910052448“); R/S C,,628571430+3“)

SQRT(2) \* SQRT(2) = ?

Eingabe:

0; ENTER ^; 2; XEQ „SSQRT“  
RCL Y; RCL Y; XEQ „\*\*“;

- a) ...
- b) ...
- c) XEQ „=“ („+2,0000000000“); R/S („,000000000+0“)

„/“ verwendet den Stack und die Register M und falls Flag 00 gesetzt auch einen Subroutine-Level

„/\*“ verwendet nur den Stack und das M-Register

„/“ löscht Flag 25, verwendet den Stack und einen Subroutine-Level

„-“ „+“ „\*\*“ verwendet den Stack, die Register M und N und einen Subroutine-Level

„//“ verwendet den Stack, die Register M, N, O, P und Q, entsprechend Vorsicht mit dem Drucker und den Alarmen; einen Subroutine-Level.

„=“ löscht Flag 25, verwendet den Stack, das ALPHA-Register, zwei Subroutine-Level

ansonsten werden keine Flags oder Register verändert.

Anmerkung:

- a) „=“ ist auf Grund von Bugs in der Betriebssystem-Anzeigenformatierung so aufwendig geworden.
- b) Aus Geschwindigkeitsgründen möglichst CF 00 wählen.
- c) Die Betriebssystem-Quadratwurzel rundet nicht treu. Das kann zu Fehlern in der 18. Stelle von „SSQRT“ und zu Fehlern in der 38. Stelle von „SSSQRT“ führen.

### 40-stellige Arithmetik

Statt in (t), (z) und (y), (x) werden die Operanden in den Registern 09,08,07,06 und 04,03,02,01 erwartet; in den Registern 04,03,02,01 wird dann das Ergebnis abgelegt.

Ausnahmen:

„/+“ erwartet die Operanden in den Registern 12 und 11 und im Stack in (y) und (x).

„/\*“ erwartet die Operanden in (t), (z) und (y), (x) und legt das Ergebnis in den Registern 14,13,12,11 ab.

(Es stehen keine Formatierungsroutinen zur Verfügung)

„/+“ benötigt außer den Operandenregistern 3 Subroutine-Level

„/\*\*\*\*“ benötigt außer den Operandenregistern 3 Subroutine-Level und die Register 00 und 15

„-----“, „++++“ benötigt außer den Operandenregistern 4 Subroutine-Level und die Register 11 und 12

„\*\*\*\*“ benötigt außer den Operandenregistern 4 Subroutine-Level und die Register 00 und 11 bis 15

„//“ benötigt außer den Operandenregistern 4 Subroutine-Level und die Register 00, 05 und 10 bis 15

„SSSQRT“ benötigt außer den Operandenregistern 4 Subroutine-Level und die Register 00, 05 und 10 bis 15

Alle Programme benutzen Flag 00, verwenden den Stack und das ALPHA-Register.

Anmerkung:

- a) Natürlich ist die Erweiterung auf 80 Stellen möglich, dies dürfte aber unsinnig sein, der verwendete Algorithmus hat nämlich exponentielle Komplexität.
- b) Wer 40 Stellen so selten braucht, daß ihn der Programmspeicher reut, kann unbeschadet alle im Listing angestrichelten Zeilen löschen.

**Verwendete Formeln**

Label

**/+** In: x,y Out: s,š  
 $s := (x+y)$   
 $\hat{s} := (y-(s-x)) - ((s-(s-x)) - x)$

**/\*** In:  $u_1, u_2, v_1, v_2$  Out: t,ť  
 Mit  $u = u_1 + u_2$   
 $v = v_1 + v_2$   
 $u_1, v_1$  treu (!) gerundet  
 $t := (u_1 v_1 + (u_1 v_2 + u_2 v_1))$   
 $\hat{t} := (u_2 v_2 + ((u_1 v_2 + u_2 v_1) - (t - u_1 v_1)))$

**++** In:  $u, \hat{u}, v, \hat{v}$  Out: z,ž  
 $s, \hat{s} := u + v$   
 $z, \hat{z} := s / + (\hat{u} + \hat{v} + \hat{s})$

**\*\*** In:  $u, \hat{u}, v, \hat{v}$  Out: z,ž  
 $t, \hat{t} := u / v$   
 $z, \hat{z} := t / + ((u \hat{v} + \hat{u} v) + \hat{t})$

**//** In:  $u, \hat{u}, v, \hat{v}$  Out: z, z  
 $t, \hat{t} := \frac{u}{v} / v$   
 $z, \hat{z} := \frac{u}{\hat{v}} / + ((u - t - \hat{t} - \frac{u}{\hat{v}} \hat{v}) / v + \frac{\hat{u}}{\hat{v}})$

**SSSQRT** In:  $u, \hat{u}$  Out: z,ž  
 $t, \hat{t} := \sqrt{u} / + \sqrt{\hat{u}}$   
 $z, \hat{z} := \sqrt{u} / + [(((u - t) - \hat{t}) + \hat{u}) / \sqrt{u}] / 2$

```

129 STO 11
130 Rf
131 STO 14
132 X<>Y
133 Rf
134 XEQ 03
135 STO 15
136 X<>Y
137 STO 00
138 RCL 11
139 RCL 14
140 XEQ 03
141 STO 11
142 X<>Y
143 X<> 12
144 RCL 13
145 XEQ 03
146 RCL 12
147 RCL 11
148 XEQ 00
149 STO 11
150 X<>Y
151 STO 12
152 X<>Y
153 RCL 00
154 RCL 15
155 XEQ 00
156 ENTERf
157 X<> 11
158 X<> 15
159 RCL Z
160 X<> 12
161 X<> 00
162 X<>Y
163 XEQ 07
164 RCL 00
165 RCL 15
166 Rf
167 Rf
168 XEQ 07
169 X<> 13
170 X<>Y
171 X<> 14
172 XEQ 03
173 RCL 14
174 RCL 13
175 XEQ 00
176 STO 13
177 X<>Y
178 STO 14
179 RTN
180+LBL "/="
181+LBL 05
182 RCL Y
183 RCL Y
184 SF 25
185 *
186 FC?C 25
187 /
188 X<0?
189 GTO 00
190 RDN
191 RTN
    
```

```

01+LBL "/+"
02+LBL 01
03 RCL X
04 RCL Z
05 +
06 ENTERf
07 STO [
08 LASTX
09 -
10 ST- Z
11 LASTX
12 RDN
13 -
14 Rf
15 -
16 -
17 RCL [
18 FC? 00
19 RTN
20 XEQ 20
21 RCL Z
22 SIGN
23 X<>Y
24 LASTX
25 ABS
26 X<>Y
27 -
28 LASTX
29 +
30 *
31 X<>Y
32 RTN
33+LBL "/++"
34 SF 00
35+LBL 02
36 STO 00
37 X<>Y
38 STO 04
39 X<>Y
40 RCL 12
41 RCL 11
42 XEQ 08
43 STO 01
44 X<>Y
45 STO 02
46 X<>Y
47 RCL 12
48 RCL 11
49 XEQ 07
50 ENTERf
51 X<> 04
52 RCL Z
53 X<> 03
54 Rf
55 Rf
56 XEQ 07
57 RCL 02
58 RCL 01
59 Rf
60 X<> 03
61 Rf
62 X<> 04
63 XEQ 07
64 RCL 12
    
```

```

65 RCL 11
66 XEQ 07
67 RCL 03
68 RCL 04
69 Rf
70 Rf
71 XEQ 07
72 STO 03
73 X<>Y
74 STO 04
75 RTN
76+LBL "/**"
77+LBL 03
78 X<>Y
79 RCL d
80 ENG 4
81 X<>Y
82 RND
83 LASTX
84 X<>Y
85 -
86 LASTX
87 Rf
88 RND
89 Rf
90 STO d
91 CLX
92 LASTX
93 X<>Y
94 -
95 STO [
96 LASTX
97 X<>Y
98 Rf
99 *
100 X<>Y
101 LASTX
102 X<>Y
103 *
104 LASTX
105 X<> [
106 Rf
107 ST* [
108 *
109 +
110 ENTERf
111 Rf
112 X<>[
113 +
114 STO Z
115 LASTX
116 -
117 -
118 RCL [
119 +
120 X<>Y
121 RTN
122+LBL "/***"
123 SF 00
124+LBL 04
125 STO 12
126 Rf
127 STO 13
128 Rf
    
```

192*LBL 00	255 XEQ 01	318 XEQ 09	381 RCL 07	444 RCL Y
193 RDN	256 X<>Y	319 X<> 00	382 RCL 06	445 STO 05
194 XEQ 20	257 RCL \	320 X<>Y	383 RCL 10	446 RCL Y
195 X<Y?	258 +	321 X<> 09	384 X<> 12	447 STO 10
196 CHS	259 GTO 01	322 X<>Y	385 RCL 05	448 XEQ 04
197 +	260*LBL "----"	323 RCL 02	386 X<> 11	449 RCL 02
198 X<>Y	261 CLX	324 RCL 01	387 XEQ 07	450 RCL 01
199 LASTX	262 SIGN	325 XEQ 09	388 RCL 14	451 RCL 05
200 -	263 CHS	326 RCL 09	389 RCL 13	452 X<> 12
201 LASTX	264 ST* 01	327 RCL 08	390 XEQ 07	453 RCL 10
202 X<>Y	265 ST* 02	328 XEQ 08	391 X<> 10	454 X<> 11
203 CHS	266 ST* 03	329 RCL 14	392 X<>Y	455 XEQ 07
204 X=Y?	267 ST* 04	330 RCL 13	393 X<> 05	456 RCL 14
205 ST- Z	268*LBL "+++"	331 XEQ 08	394 RCL 04	457 RCL 13
206 X=Y?	269 SF 00	332 GTO 02	395 RCL 03	458 XEQ 07
207 CLX	270 RCL 01	333*LBL "///"	396 XEQ 09	459 RCL 04
208 CHS	271 STO 11	334*LBL 10	397 RCL 05	460 RCL 03
209 RCL Z	272 RCL 02	335 STO -	398 RCL 10	461 XEQ 08
210 RTN	273 STO 12	336 R↑	399 R↑	462 RCL 05
211*LBL 20	274 RCL 04	337 STO \	400 R↑	463 RCL 10
212 ENTER↑	275 X<> 07	338 R↑	401 XEQ 07	464 XEQ 10
213 X=0?	276 RCL 03	339 STO J	402 RCL 02	465 ENTER↑
214 RTN	277 X<> 06	340 R↑	403 RCL 01	466 CLX
215 ABS	278 XEQ 02	341 STO ↑	404 XEQ 10	467 ENTER↑
216 LOG	279 RCL 01	342 X<>Y	405 X<> 09	468 SIGN
217 INT	280 STO 11	343 R↑	406 X<>Y	469 ST+ X
218 STO T	281 RCL 02	344 /	407 X<> 08	470 XEQ 10
219 CHS	282 STO 12	345 LASTX	408 RCL 02	471 GTO 02
220 10↑X	283 RCL 09	346 XEQ 03	409 RCL 01	472*LBL "=="
221 X<>Y	284 RCL 08	347 X<>Y	410 XEQ 10	473 XEQ 05
222 *	285 RCL 07	348 RCL J	411 RCL 08	474 RCL d
223 LASTX	286 RCL 06	349 RCL Z	412 RCL 09	475 X<>Y
224 X<>Y	287 XEQ 08	350 -	413 XEQ 08	476 CLA
225 ABS	288 RCL 04	351 X<>Y	414 GTO 02	477 SCI 9
226 SIGN	289 RCL 03	352 -	415*LBL "SSQRT"	478 ARCL X
227 X<>Y	290 XEQ 08	353 RCL J	416*LBL 11	479 ABSP
228 X<> L	291 GTO 02	354 RCL -	417 X<>Y	480 ABSP
229 X<>Y	292*LBL "***"	355 /	418 STO \	481 ANUM
230 X>Y?	293*LBL 09	356 STO Z	419 X<>Y	482 "+"
231 ST- T	294 STO \	357 RCL ↑	420 STO J	483 X<0?
232 R↑	295 R↑	358 *	421 SQRT	484 "--"
233 R↑	296 *	359 -	422 ENTER↑	485 ABS
234 LASTX	297 RDN	360 RCL -	423 XEQ 03	486 FIX 9
235 RCL Z	298 *	361 /	424 X<>Y	487 ARCL X
236 PI	299 R↑	362 RCL \	425 RCL J	488 X=0?
237 X↑2	300 +	363 RCL -	426 RCL Z	489 RDN
238 INT	301 X<> \	364 /	427 -	490 X*0?
239 -	302 XEQ 03	365 +	428 X<>Y	491 /
240 10↑X	303 X<>Y	366 GTO 01	429 -	492 X<>Y
241 RTN	304 RCL \	367*LBL "///"	430 RCL \	493 STO d
242*LBL "----"	305 +	368 SF 00	431 +	494 RDN
243*LBL 07	306 GTO 01	369 RCL 07	432 RCL J	495 ABS
244 CHS	307*LBL "****"	370 RCL 06	433 SQRT	496 STO Z
245 X<>Y	308 SF 00	371 RCL 02	434 STO Z	497 E10
246 CHS	309 RCL 07	372 RCL 01	435 /	498 /
247 X<>Y	310 RCL 06	373 XEQ 10	436 2	499 X*0?
248*LBL "+++"	311 RCL 02	374 STO 05	437 /	500 /
249*LBL 00	312 RCL 01	375 X<>Y	438 GTO 01	501 ABS
250 X<>Y	313 XEQ 04	376 STO 10	439*LBL "SSSQRT"	502 ARCL I
251 R↑	314 RCL 07	377 X<>Y	440 SF 00	503 SF 25
252 +	315 RCL 06	378 RCL 02	441 RCL 02	504 PC<>RTN
253 STO \	316 RCL 04	379 RCL 01	442 RCL 01	505 FC?C 25
254 RDN	317 RCL 03	380 XEQ 04	443 XEQ 11	

506 PROMPT	515 STO d	524 ATOX
507 RCL d	516 R+	525 AVIEW
508 X<>Y	517 X#0?	526 CLST
509 FRC	518 LOG	527 .END.
510 FIX 9	519 X#0?	
511 RND	520 X)0?	
512 CLA	521 "I+"	Christian Martini
513 ARCL X	522 ARCLI	Am Reichswald 6
514 X<>Y	523 ATOX	8503 Altdorf

## Steuer 88

494 Zeilen, 912 Bytes, 131 Regs., SIZE 015, HP-41CV

### Einkommensteuer-/Lohnsteuerprogramm für 1987/1988

#### Aufruf:

XEQ „ST87“  
 (Die einzelnen Bereiche (Steuerprogramme) A bis D können innerhalb von ST87 beliebig oft und in beliebiger Reihenfolge aufgerufen werden. E sollte nur im Anschluß an einen dieser Bereiche aufgerufen werden.)

#### Bereiche:

- A Einkommensteuer aus zu versteuerndem Einkommen
- B Prozentualer Spitzensteuersatz
- C zu versteuerndes Einkommen aus Steuerbetrag
- D Lohnsteuer nach allgemeiner Tabelle
- E Kirchensteuer

#### Registerbelegung:

- R00 Lohn
- R01 ZVE (zu versteuerndes Einkommen, ungerundet)
- R02 ST (Steuerbetrag, Einkommen- bzw. Lohn-)
- R03 KIST (Kirchensteuer)
- R04 VSP (Vorsorgepauschale)
- R05 ZKF (Zahl der Kinderfreibeträge)
- R06 STKL (Steuerklasse, 1-6)
- R07 LZZ (Lohnzahlungszeitraum, 1 = Jahr, 2 = Monat, 3 = Woche, 4 = Tag)
- R08 Rechenfeld, enthält LZZ + 10
- R09 RE40, Rechenfeld, Obergrenze der Lohnsteuerstufe
- R10 RE4U, Rechenfeld, Untergrenze der Lohnsteuerstufe
- R11 %ST (prozentualer Spitzensteuersatz) bzw. TABFB (Tabellenfreibeträge der Lohnsteuerklassen)
- R12 Rechenfeld, Steuer für ZVE-Berechnung (Bereich C)
- R13 Rechenfeld, Δ ZVE
- R14 Rechenfeld, Steuer aus vorheriger ZVE-Berechnung

#### Flags:

- SF 00 STKL nicht eingegeben
- SF 01 STKL 1
- SF 02 STKL 2
- SF 03 STKL 3 und Splitting-Berechnung
- SF 04 STKL 4
- SF 05 STKL 5
- SF 06 STKL 6
- SF 07 für %ST zur Berechnung der 1. Ableitung der Steuerformel
- SF 08 für ZVE-Berechnung

Erläuterungen (s. a. Registerbelegung) zu den einzelnen Bereichen; nach jeder Eingabe (oder Unterlassung) mit R/S fortfahren:

#### Bereich A:

Eingabe von ZVE, gültiger Bereich ≥ 0, Unterlassungswert und Ersatzwert für ungültigen Bereich = 0  
 Eingabe, ob Splitting = 3, Unterlassungswert und Ersatzwert für ungültigen Bereich = 0  
 Ausgabe ST = Einkommensteuer

#### Bereich B:

Eingabe von ZVE und Splitting siehe Bereich A  
 Ausgabe %ST, Berechnung aus der 1. Ableitung der Steuerformel,

theoretischer Wert für die Steuerbelastung einer zusätzlichen DM zum ZVE; ungenau, da ST in Stufen steigt

#### Bereich C:

Eingabe ST und Splitting = 3, Unterlassungswert und Ersatzwert siehe Bereich A (bei ST = 0 wird programmbedingt ZVE = -10.000 angezeigt)  
 Ausgabe ZVE (angenähert), Berechnung iterativ über einen Anfangswert für ZVE = 10.000 und eine Schrittweite für Δ ZVE = 10.000, Reduzierung von Δ ZVE um den Faktor -0,1, wenn sich das Vorzeichen der Differenz zwischen vorgegebener ST und einer Berechnung ändert, Abbruch, falls zwei Berechnungen hintereinander das gleiche Ergebnis liefern

#### Bereich D:

Eingabe LZZ, gültiger Bereich 1-4, sonst Fehlermeldung  
 Eingabe STKL, gültiger Bereich 1-6, sonst Fehlermeldung und Rücksprung zur erneuten Eingabe ab LZZ  
 Eingabe ZKF, gültiger Bereich ≥ 0, Fehlerbehandlung wie bei STKL  
 Eingabe LOHN, gültiger Bereich usw. wie bei ZKF  
 Ausgabe ST = Lohnsteuer entsprechend LZZ gemäß allgemeiner Lohnsteuertabelle; es gibt noch eine besondere Lohnsteuertabelle für nicht rentenversicherungspflichtige Arbeitnehmer (z.B. Beamte), hierzu ist folgende Programmänderung bei der Vorsorgepauschale erforderlich:

Zeile 194 = 1 (statt 2)  
 Zeile 196 = 100.000 (statt 117.000)

#### Hinweise:

STKL 2 ohne ZKF, STKL 5 mit ZKF und STKL 6 mit ZKF sind nach den Lohnsteuerrichtlinien nicht definiert, diese Fälle werden im Programm nicht als Fehler abgefangen, es kann also ein falsches Ergebnis kommen. Steuerfreibeträge lt. Lohnsteuerkarte müssen vorab manuell abgezogen werden.

#### Bereich E:

Eingabe keine, das Programm benutzt die aus der Vorberechnung A-D in den Registern 02 (ST) und 05 (ZKF) abgespeicherten Werte, als Steuersatz sind 9% (NRW) in Zeile 296 vorgegeben für die Berechnung im Anschluß an A-C muß ZKF > 0 in das Register 05 vor Abruf von E gebracht werden  
 Ausgabe KIST gemäß ST (Einkommensteuer bzw. Lohnsteuer gemäß LZZ)

Diese Programme wurden nach dem Programmablaufplan zur maschinellen Berechnung der Lohnsteuer, veröffentlicht im Bundesgesetzblatt I 1987 Seite 571 ff., erstellt. Für 1990 sind neue Steuerformeln zu erwarten (Steuerreform), mit deren Veröffentlichung im Herbst 1989 zu rechnen ist.  
 Die Lohnsteuertabellen sind im Bundesanzeiger vom 29.09.1987 (als Anfrage) veröffentlicht worden.

#### Übersicht über die einzelnen Programmteile:

LBL / Titel	Zeilen	Bemerkungen
Hauptroutine „ST87“	001 - 014	mit Rücksprungpunkt LBL c aus A, B, C, D in Zeile 008
A Einkommensteuer	015 - 026	ruft a, G (XEQ a, XEQ G) springt nach LBL c
B Prozentuale Steuerbelastung	027 - 037	ruft a, G (XEQ a, XEQ G) springt nach LBL c
C ZVE aus Steuer	038 - 070	ruft I (XEQ I) springt nach LBL c
D Lohnsteuer	071 - 121	Eingabe ruft d (GTO d) springt nach LBL 01 ... 06 (GTO 01 ... 06)
	weiter 122 - 155	Tabellenfreibeträge gem. STKL
	weiter 156 - 186	Lohnstufen ruft 00, 10 ... 14 für LZZ-Umrechnung (XEQ 00, 10 ... 14 aus REG 08)
	weiter 187 - 214	Vorsorgepauschale ruft b (XEQ b)

LBL / Titel	Zeilen	Bemerkungen		
			127 27	190 GTO 08
	weiter 215 - 283	Steuerberechnung	128 RCL 05	191 9
		ruft b, H (XEQ b, XEQ H)	129 1242	192 *
		(Zeilen 225-252 STKL 5/6)	130 *	193 STO 04
		ruft 00, 10... 14 für LZZ s.o.	131 +	194 2
		(Zeilen 253-278 Rundungen	132 270	195 ENTER↑
		für LZZ = 2 - 4)	133 +	196 117000
		springt nach LBL c	134 GTO 05	197 FS? 03
E Kirchensteuer	284 - 308	ruft 00, 10... 14 für LZZ s.o.	135*LBL 02	198 ST+ X
		springt zurück in die	136 4482	199 *
		Hauptroutine	137 +	200 X>Y?
G Steuerformeln	309 - 314	ruft b (XEQ b)	138*LBL 03	201 RDN
H dto. Fortsetzung	315 - 424	darin 1. Ableitung bei SF 07	139 270	202 X<> 04
		enthalten	140 +	203 LASTX
I Steuerung Iteration ZVE	425 - 451	ruft G (XEQ G)	141*LBL 01	204 X>Y?
a Eingabe bei A und B	452 - 466		142 270	205 RDN
b Rundet auf volle	467 - 473		143 +	206 RCL 04
DM 54,-			144 RCL 05	207 +
d Fehlermeldung bei D	474 - 479	springt an den Anfang von D	145 2484	208 100
		(GTO D)	146 *	209 /
00, 10... 14	480 - 494	Aufteilung gemäß LZZ	147 +	210 INT
			148*LBL 05	211 XEQ b
			149 1026	212 STO 04
			150 +	213 ST- 01
			151*LBL 06	214*LBL 08
			152 18	215 .5
			153 +	216 RCL 01
			154 STO 11	217 FS? 03
			155 ST- 01	218 *
			156 CLST	219 XEQ b
			157 STO 04	220 FS? 05
			158 STO 09	221 GTO 01
			159 STO 10	222 FS? 06
			160 RCL 00	223 GTO 01
			161 100	224 XEQ H
			162 *	225 INT
			163 7200	226 FS? 03
			164 XEQ IND 08	227 ST+ X
			165 X>Y?	228 GTO 02
			166 GTO 07	229*LBL 01
			167 CLX	230 1.25
			168 1800	231 *
			169 XEQ IND 08	232 XEQ H
			170 -	233 INT
			171 5400	234 2
			172 XEQ IND 08	235 *
			173 /	236 STO 02
			174 INT	237 RCL 01
			175 5400	238 .75
			176 *	239 *
			177 1800	240 XEQ H
			178 +	241 INT
			179 100	242 2
			180 /	243 *
			181 ST+ 01	244 RCL 02
			182 STO 10	245 X<>Y
			183 53	246 -
			184 +	247 RCL 01
			185 STO 09	248 .22
			186*LBL 07	249 *
			187 FS? 05	250 INT
			188 GTO 08	251 X<=Y?
			189 FS? 06	252 RDN
01*LBL "ST87"	43 INT	85 RDN		
02 SF 27	44 X<0?	86 STO 07		
03 CF 28	45 0	87 10		
04 SF 29	46 STO 12	88 +		
05 FIX 0	47 STO 14	89 STO 08		
06 "BEREICH? A-D"	48 3	90 "STKL? 1-6"		
07 PROMPT	49 "SPLIT?J=3"	91 PROMPT		
08*LBL c	50 PROMPT	92 INT		
09 CF IND 06	51 INT	93 ENTER↑		
10 ARCL X	52 X=Y?	94 1		
11 PROMPT	53 GTO 01	95 X>Y?		
12 "KIST? E"	54 STO 06	96 GTO d		
13 PROMPT	55 2	97 CLX		
14 RTN	56 ST/ 12	98 7		
15*LBL A	57 ST/ 14	99 X<=Y?		
16 CLRG	58*LBL 01	100 GTO d		
17 CLST	59 SF IND 06	101 RDN		
18 XEQ a	60 1 E5	102 STO 06		
19 XEQ G	61 STO 01	103 SF IND 06		
20 INT	62 STO 13	104 FIX 1		
21 FS? 03	63 XEQ I	105 0		
22 ST+ X	64 FS? 03	106 ENTER↑		
23 STO 02	65 ST+ X	107 "ZKF?"		
24 FIX 2	66 STO 02	108 PROMPT		
25 "EST="	67 RCL 01	109 X<Y?		
26 GTO c	68 FIX 0	110 GTO d		
27*LBL B	69 "ZVE="	111 STO 05		
28 CLRG	70 GTO c	112 FIX 2		
29 CLST	71*LBL D	113 0		
30 XEQ a	72 CLRG	114 ENTER↑		
31 SF 07	73 CLST	115 "LOHN?"		
32 XEQ G	74 "LZZ? 1-4"	116 PROMPT		
33 STO 11	75 PROMPT	117 X<Y?		
34 CF 07	76 INT	118 GTO d		
35 FIX 2	77 ENTER↑	119 STO 00		
36 "%ST="	78 1	120 CLST		
37 GTO c	79 X>Y?	121 GTO IND 06		
38*LBL C	80 GTO d	122*LBL 04		
39 CLRG	81 CLX	123 RCL 05		
40 CLST	82 5	124 FRC		
41 "EST?"	83 X<=Y?	125 0		
42 PROMPT	84 GTO d	126 X=Y?		

Auf das Ihr viele Steuern spart !!!

253*LBL 02	316 4537	379*LBL 04	442 XEQ 02
254 XEQ IND 08	317 X>Y?	380 CLX	443 GTO I
255 RCL 08	318 GTO 01	381 79974	444*LBL 01
256 12	319 CLX	382 -	445 FC?C 08
257 -	320 18036	383 I E2	446 XEQ 02
258 X=0?	321 X>Y?	384 /	447 GTO I
259 GTO 03	322 GTO 02	385 FS? 07	448*LBL 02
260 X>0?	323 CLX	386 GTO 07	449 -.1
261 GTO 04	324 80028	387 7 E4	450 ST+ 13
262 RDN	325 X>Y?	388 %	451 RTN
263 GTO 05	326 GTO 03	389 INT	452*LBL a
264*LBL 03	327 CLX	390 49 E5	453 "ZVE?"
265 CLX	328 130032	391 +	454 PROMPT
266 10	329 X>Y?	392 %	455 X<0?
267 *	330 GTO 04	393 INT	456 0
268 INT	331 CLX	394 I E3	457 STO 01
269 10	332 56	395 /	458 3
270 /	333 FS? 07	396 26974	459 "SPLIT?J=3"
271 GTO 05	334 RTN	397 +	460 PROMPT
272*LBL 04	335 %	398 RTN	461 INT
273 CLX	336 19561	399*LBL 06	462 X*Y?
274 100	337 -	400 1360	463 0
275 *	338 RTN	401 %	464 STO 06
276 INT	339*LBL 01	402 INT	465 SF IND 06
277 100	340 0	403 64740	466 RTN
278 /	341 RTN	404 -	467*LBL b
279*LBL 05	342*LBL 02	405 %	468 54
280 STO 02	343 CLX	406 INT	469 /
281 FIX 2	344 22	407 784 E3	470 INT
282 "LST="	345 FS? 07	408 +	471 54
283 GTO c	346 RTN	409 %	472 *
284*LBL E	347 %	410 INT	473 RTN
285 100	348 1045	411 22 E5	474*LBL d
286 RCL 02	349 -	412 +	475 BEEP
287 *	350 RTN	413 I E5	476 "FEHLER"
288 RCL 05	351*LBL 03	414 /	477 AVIEW
289 30000	352 CLX	415 RTN	478 PSE
290 *	353 17982	416*LBL 07	479 GTO D
291 FC? 04	354 -	417 14 E4	480*LBL 13
292 ST+ X	355 I E2	418 %	481 7
293 XEQ IND 08	356 /	419 INT	482 *
294 INT	357 FS? 07	420 49 E5	483*LBL 14
295 -	358 GTO 06	421 +	484 30
296 9	359 340	422 I E5	485 /
297 %	360 %	423 /	486*LBL 12
298 INT	361 INT	424 RTN	487 12
299 100	362 21580	425*LBL I	488 /
300 /	363 -	426 RCL 13	489*LBL 11
301 X<0?	364 %	427 ST+ 01	490 1
302 0	365 INT	428 XEQ G	491 /
303 STO 03	366 392 E3	429 INT	492*LBL 00
304 FIX 2	367 +	430 RCL 14	493*LBL 10
305 "KIST="	368 %	431 X=Y?	494 END
306 ARCL X	369 INT	432 RTN	
307 AVIEW	370 22 E5	433 RDN	
308 RTN	371 +	434 STO 14	
309*LBL G	372 %	435 RCL 12	
310 .5	373 INT	436 X>Y?	
311 RCL 01	374 I E3	437 SF 08	
312 FS? 03	375 /	438 RCL 13	
313 *	376 2911	439 X>0?	
314 XEQ 'b	377 +	440 GTO 01	
315*LBL H	378 RTN	441 FS?C 08	

# CAT-Bug

## CAT-Bug in der Anzeige

Da der Fehler sehr unterschiedlich auftreten kann, habe ich unten ein Diagramm aufgezeichnet, das alle mir bekannten Möglichkeiten erfaßt. Als Beispiel und gewissermaßen als Anleitung zum Diagramm beschreibe ich hier den spektakulären Fall:

Irgendeinen Katalog laufen lassen, wobei aber nicht die erweiterte Katalogfunktion des CCD-Moduls verwendet werden kann (Katalognummer-Taste vor dem Loslassen der CAT-Taste drücken). Den Katalog abstoppen und SST oder BST durch **Buchstabieren** ausführen. Wird nun irgendeine Taste außer der Korrekturtaste kurz angeippt, fängt die Sache endlich an: Der Programmname oder was sonst in der Anzeige stand, steht wie im Diagramm beschrieben an verschobener Stelle. Kaum jemand wird davon schon genug haben und deshalb schlage ich vor, die fehlerhafte Meldung mit der Korrekturtaste wegzuschaffen. Falls jemand den Inhalt des X-Registers erwartet hat, wird er sich noch umgewöhnen müssen: Das Ergebnis heißt DATA ERROR. Möchte man diesen mit der Korrekturtaste überlisten, verschiebt er sich nur nach rechts, und bei weiteren Versuchen erscheint immer das gleiche Bild: Teile der Anzeige verlieren während dem Tastendruck den Kontrast, je nach Jahrgang des Rechners kann es etwas anders sein, bei meinem älteren HP-41CV sind es die unteren Hälften der 14-Segment-Ziffern, die beim richtigen Blickwinkel erblassen. Erneutes Drücken einer anderen Taste löscht die Meldung, oder der Rechner schaltet sogar ab. Für weitere Details betrachte man das Diagramm und die Tabelle.

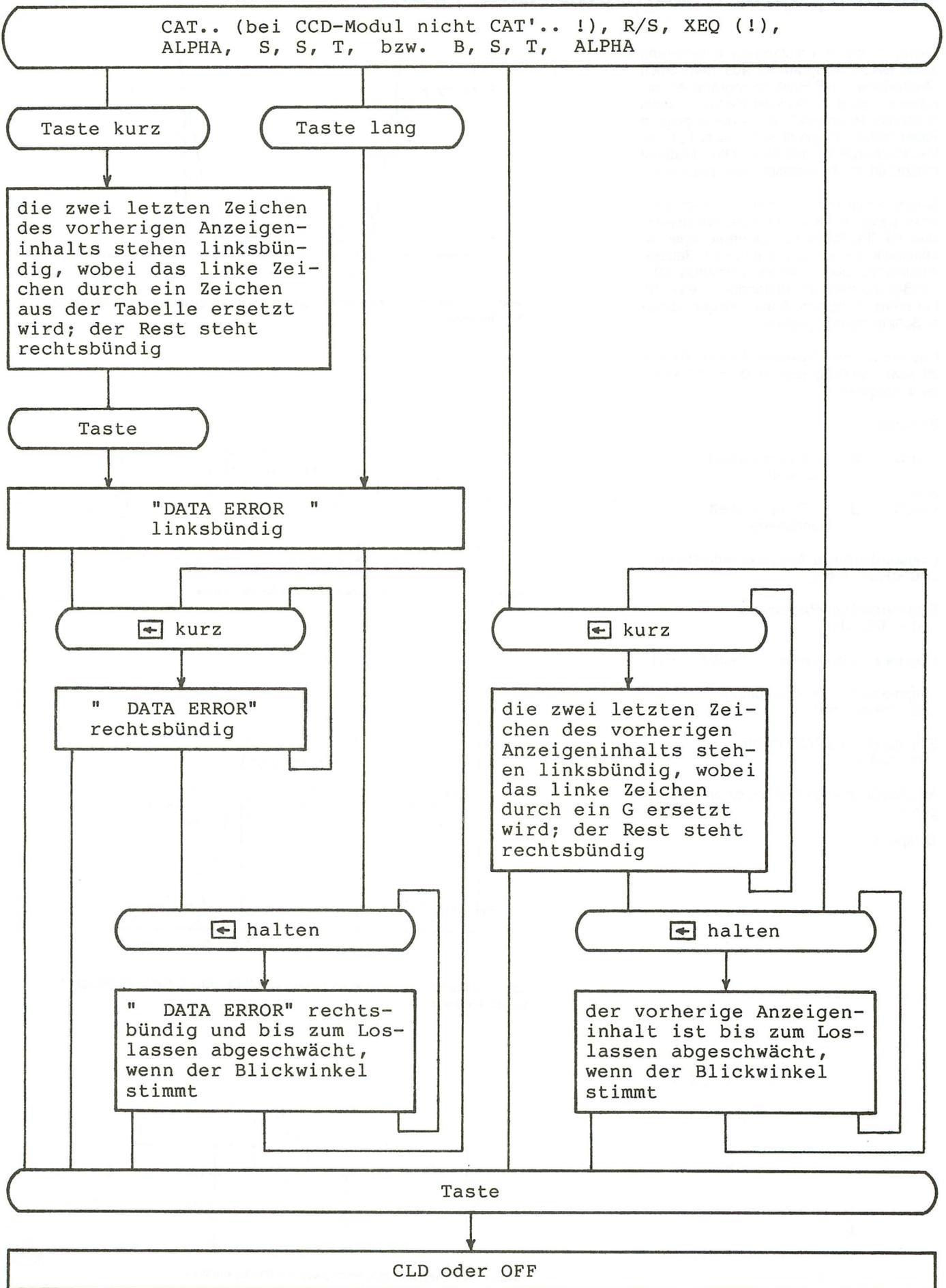
Zum Schluß habe ich noch eine Frage an ROM-Kenner: Wer könnte mir sagen, wie die Einstiegsadressen der Shifttaste heißen, mit denen man das Shiftflag programmiert setzen, löschen und vor allem wechseln kann, ohne dabei immer direkt das Register d manipulieren zu müssen?

Tabelle der tastennummerabhängigen Ersatzzeichen:

11: S	12: S	13: S	14: S	15: S
21: S	22: S	23: S	24: S	25: S
31: C.	32: C	33: C	34: C	35: C
41: C	42: C	43: C	(44: G)	
51: C	52: K	53: K	54: K	
61: C	62: K	63: K	64: K	
71: C	72: K	73: K	74: K	
81: C	82: K	83: C:	84: C	

Georg Hoffmeier (1439)  
Marienburger Str. 6  
4040 Neuss 1

Michael Brand  
Ramelgässli 74  
CH-3067 Boll



# Tschebyscheff Bandpaß- und-sperre

Dieses Programm entstand durch Portierung eines BASIC-Programms aus dem Buch „Nolte-Basic: HF-Rechenprogramme“ erschienen als Band 9 in der Reihe „Franzis Computer-Bibliothek“. Auch die gezeigten Bilder stammen aus diesem Buch. Für das freundlicherweise bereitgestellte Material möchte ich mich an dieser Stelle bedanken.

Dieses Programm brechnet die Elemente eines passiven Bandpaß- bzw. Bandsperrefilter mit Tschebyscheff-Übertragungscharakteristik. Es werden realistische Bauelementewerte bis zu einem Verhältnis 10:1 von Betriebsfrequenz zu Bandbreite erreicht. Für schmalbandigere Anwendungen ist diese Schaltung nicht geeignet.

Das Programm verwendet die Flag 01 und 29 sowie die Datenregister 00 bis 10 für interne Aufgaben.

Bedienung:

<XEQ> „TBP“ für Tschebyscheff Bandpaß

bzw.

<XEQ> „TBS“ für Tschebyscheff Bandsperre

Eingabe des Ein- u. Ausgangswiderstands „R0“ <R/S> [ohm];

Eingabe der Betriebsfrequenz „F0“ <R/S> [Hz];

Eingabe der Bandbreite „ΔF“ <R/S> [Hz];

Eingabe der Rückflußdämpfung „RD“ <R/S> [dB];

Eingabe der Anzahl Elemente „N“ <R/S>.

Anschließend werden alle Elemente ausgegeben.

Beispiele

```

XEQ "TBP"
R0 = ?
 75,0000 RUN
F0 = ?
 200+05 RUN
ΔF = ?
 60+05 RUN
RD = ?
 18,0000 RUN
N = ?
 3,0000 RUN
CP(1)=33,49E-12 F
RUN
LP(1)=18,91E-9 H
RUN
CS(2)=2,811E-12 F
RUN
LS(2)=225,3E-9 H
RUN
CP(3)=33,49E-12 F
RUN
LP(3)=18,91E-9 H
RUN
    
```

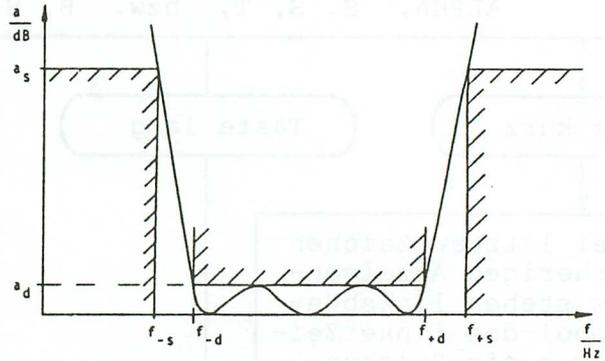


Abb. 1 Dämpfungsverlauf eines Tschebyscheff-Bandpaßfilters in Abhängigkeit von der Frequenz

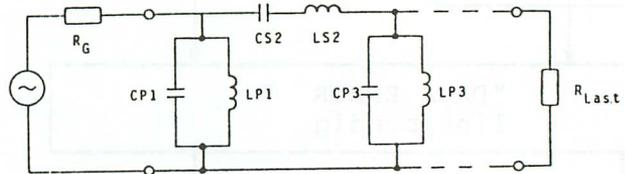


Abb. 2 Prinzipschaltbild eines passiven Bandpaßfilters

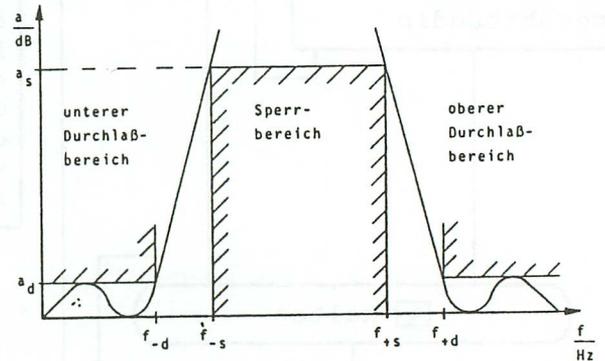


Abb. 3 Dämpfungsverlauf eines Tschebyscheff-Bandsperrefilters in Abhängigkeit von der Frequenz

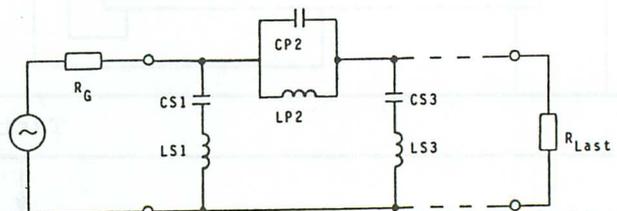


Abb. 4 Prinzipschaltbild eines passiven Bandsperrefilters

XEQ "TBS"		75*LBL 04		138 E	163 PROMPT
R0 =?		76 RCL 00		139	164 "L"
75,0000 RUN		77 RCL 06		140 PI	165 ARCL X
F0 =?		78 FS? 01		141 *	166 "f("
98+06 RUN		79 1/X		142 RCL 07	167 FIX 0
ΔF =?		80 *		143 /	168 ARCL 08
20+06 RUN		81 RCL 02		144 SIN	169 "f)="
RD =?		82 /		145 X+2	170 ENG 3
18,0000 RUN		83 RCL 01		146 RCL 04	171 ARCL Z
N =?		84 RCL 03		147 X+2	172 "f H"
3,0000 RUN		85 YfX		148 +	173 PROMPT
CS(1)=4,184E-12 F		86 /		149 /	174 RTN
RUN		87 RCL 01		150 STO 06	175*LBL 14
LS(1)=630,3E-9 H		88 RCL 03		151 GTO 04	176 "f =?"
RUN		89 YfX		152*LBL 10	177 PROMPT
CP(2)=93,70E-12 F		90 RCL 00		153 ASTO X	178 RTN
RUN		91 /		154 "C"	179*LBL 00
LP(2)=28,15E-9 H		92 RCL 06		155 ARCL X	180 CLST
RUN		93 FS? 01		156 "f("	181 CLA
CS(3)=4,184E-12 F		94 1/X		157 FIX 0	182 CF 01
RUN		95 /		158 ARCL 08	183 SF 29
LS(3)=630,3E-9 H		96 RCL 03		159 "f)="	184 DEG
RUN		97 X<0?		160 ENG 3	185 FIX 4
		98 "P"		161 ARCL Y	186 END
		99 X<0?		162 "f F"	
		100 GTO 00			
01*LBL "TBP"	38 E	101 "S"			
02 SF 01	39 +	102 RDN			
03 -1	40 1/X	103 X<>Y			
04 GTO 00	41 ST+ X	104 Rf			
05*LBL "TBS"	42 E	105*LBL 00			
06 CF 01	43 -	106 XEQ 10			
07 E	44 CHS	107 -1			
08*LBL 00	45 1/X	108 ST* 03			
09 STO 03	46 STO 04	109 ST- 08			
10 RAD	47 "N"	110 RCL 07			
11 CF 29	48 XEQ 14	111 RCL 08			
12 "R0"	49 STO 07	112 X>Y?			
13 XEQ 14	50 2	113 GTO 00			
14 STO 01	51 *	114 RCL 05			
15 "F0"	52 1/X	115 STO 09			
16 XEQ 14	53 RCL 04	116 RCL 06			
17 ST+ X	54 X<>Y	117 STO 10			
18 PI	55 YfX	118 Rf			
19 *	56 RCL X	119 Rf			
20 X+2	57 1/X	120 ST+ X			
21 STO 02	58 -	121 E			
22 "ΔF"	59 2	122 -			
23 XEQ 14	60 /	123 X<>Y			
24 ST+ X	61 STO 04	124 /			
25 PI	62 PI	125 PI			
26 *	63 2	126 *			
27 STO 00	64 /	127 2			
28 "RD"	65 RCL 07	128 /			
29 XEQ 14	66 /	129 SIN			
30 -10	67 SIN	130 STO 05			
31 /	68 STO 05	131 RCL 09			
32 10fX	69 X<>Y	132 *			
33 E	70 /	133 4			
34 -	71 ST+ X	134 *			
35 CHS	72 STO 06	135 RCL 10			
36 1/X	73 E	136 /			
37 SQRT	74 STO 08	137 RCL 08			

Werner Meschede  
Sorpestraße 4  
5788 Siedlinghausen

## Wer kann mir weiter helfen?

Wenn in HP41 das Time-Modul ist und man die Befehle „PRP“ und „LIST“ ausführt, dann druckt der IL-Drucker immer als 1. Zeile Datum und Uhrzeit.

Da ich meine Programme, um sie leichter lesbar zu machen, mit „LIST----“ in Blöcke zerlege, stört die jedesmal dazwischen geschobene Uhrzeit/Datum-Zeile.

Wer kennt eine programmierbare Möglichkeit zur Unterstützung der „störenden“ Zeile?

Erhard Ristow  
Hersbruckerstraße 188  
8500 Nürnberg 30

**SE (Trzeciak & Föh) 1575**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 4) CCD-Barcodes



Zeile 2 ( 4- 8) CCD-Barcodes



Zeile 3 ( 8- 14) CCD-Barcodes



Zeile 4 ( 14- 16) CCD-Barcodes



Zeile 5 ( 16- 26) CCD-Barcodes



Zeile 6 ( 26- 33) CCD-Barcodes



Zeile 7 ( 33- 38) CCD-Barcodes

**FLHD (Kamp) 1576**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 4) CCD-Barcodes



Zeile 2 ( 4- 10) CCD-Barcodes



Zeile 3 ( 10- 19) CCD-Barcodes



Zeile 4 ( 19- 25) CCD-Barcodes



Zeile 5 ( 25- 34) CCD-Barcodes



Zeile 6 ( 34- 42) CCD-Barcodes



Zeile 7 ( 42- 50) CCD-Barcodes



Zeile 8 ( 50- 55) CCD-Barcodes

**TTP (Meschede) 1577**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 4) CCD-Barcodes



Zeile 2 ( 4- 10) CCD-Barcodes



Zeile 3 ( 10- 15) CCD-Barcodes



Zeile 4 ( 15- 22) CCD-Barcodes



Zeile 5 ( 22- 30) CCD-Barcodes



Zeile 6 ( 30- 35) CCD-Barcodes



Zeile 7 ( 35- 43) CCD-Barcodes



Zeile 8 ( 43- 51) CCD-Barcodes



Zeile 9 ( 51- 56) CCD-Barcodes



Zeile 10 ( 56- 67) CCD-Barcodes



Zeile 11 ( 67- 73) CCD-Barcodes



Zeile 12 ( 73- 85) CCD-Barcodes



Zeile 13 ( 85- 97) CCD-Barcodes



Zeile 14 ( 97- 107) CCD-Barcodes



Zeile 15 ( 107- 116) CCD-Barcodes



Zeile 16 ( 116- 125) CCD-Barcodes



Zeile 17 ( 125- 137) CCD-Barcodes



Zeile 18 ( 137- 149) CCD-Barcodes



Zeile 19 ( 149- 161) CCD-Barcodes



Zeile 20 ( 161- 173) CCD-Barcodes



Zeile 21 ( 173- 178) CCD-Barcodes



Zeile 22 ( 178- 185) CCD-Barcodes



Zeile 23 ( 185- 194) CCD-Barcodes



Zeile 24 ( 194- 207) CCD-Barcodes



Zeile 25 ( 207- 220) CCD-Barcodes



Zeile 26 ( 220- 230) CCD-Barcodes



Zeile 27 ( 230- 238) CCD-Barcodes



Zeile 28 ( 238- 239) CCD-Barcodes

**YW (Gerads) 1578**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 2) CCD-Barcodes



Zeile 2 ( 2- 9) CCD-Barcodes



Zeile 3 ( 9- 15) CCD-Barcodes



Zeile 4 ( 15- 21) CCD-Barcodes



Zeile 5 ( 21- 22) CCD-Barcodes

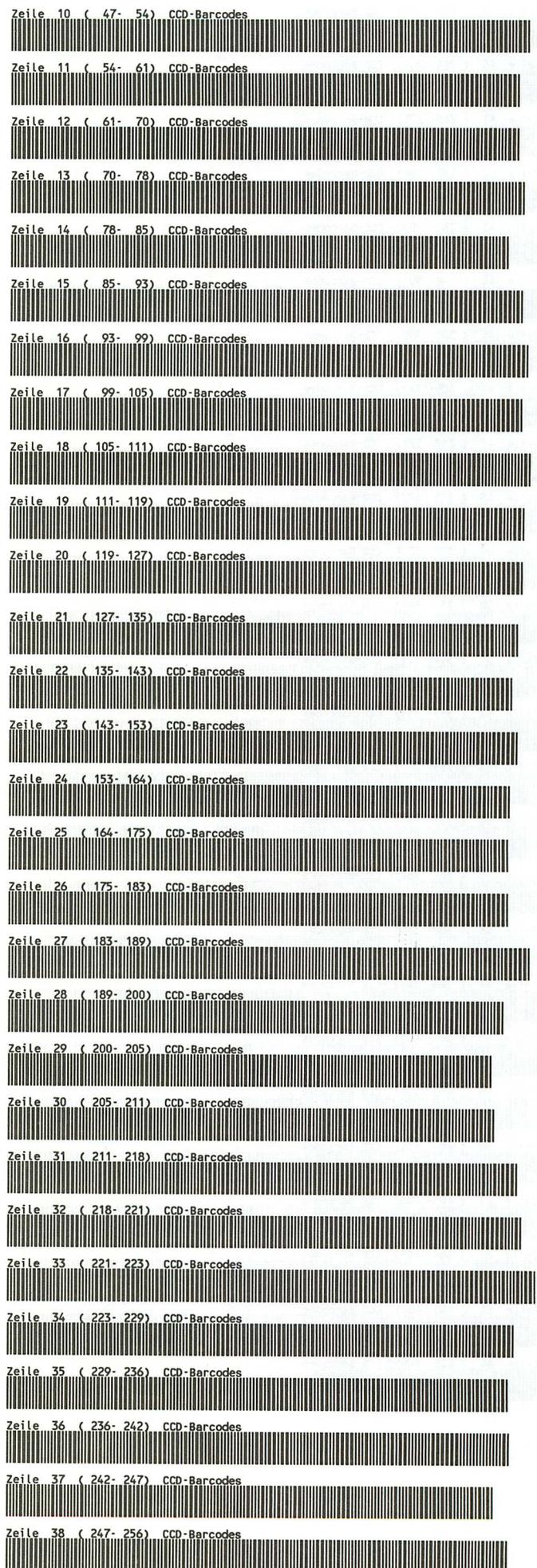
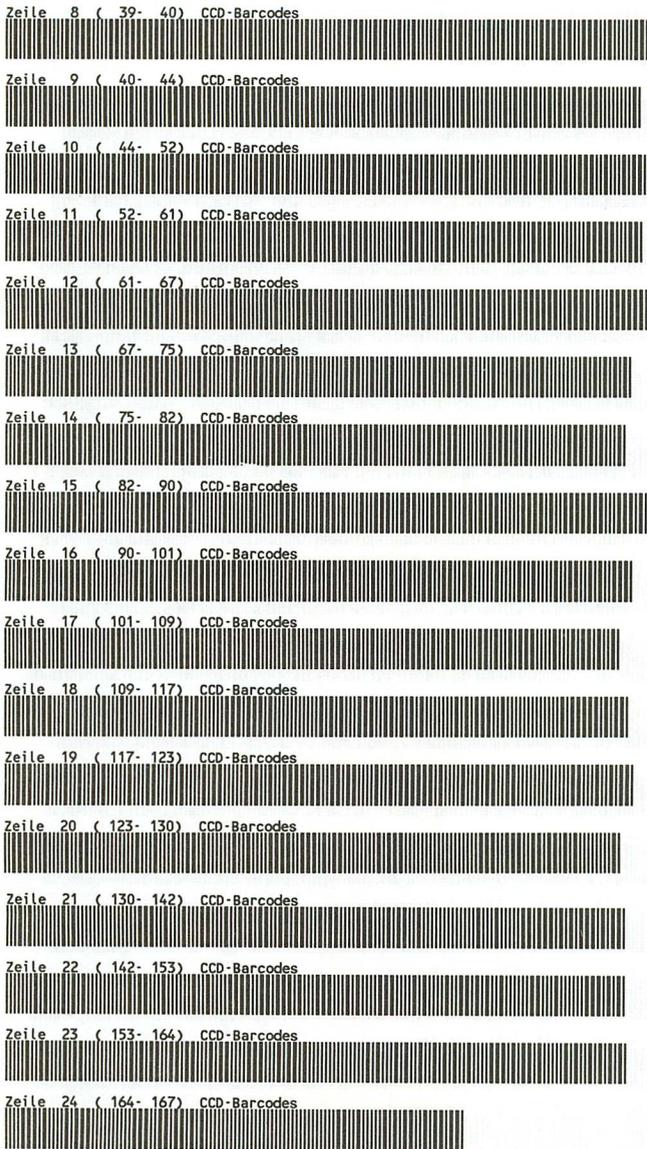


Zeile 6 ( 22- 33) CCD-Barcodes



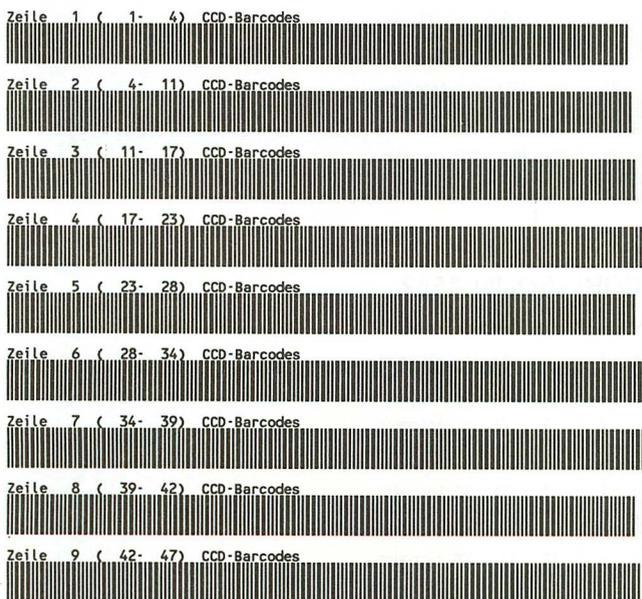
Zeile 7 ( 33- 39) CCD-Barcodes





**TRI1 (Radas) 1580**

Computerclub Deutschland e.V.



# BARCODES

Zeile 39 ( 256- 261) CCD-Barcodes



Zeile 40 ( 261- 268) CCD-Barcodes



Zeile 41 ( 268- 275) CCD-Barcodes



Zeile 42 ( 275- 280) CCD-Barcodes



Zeile 43 ( 280- 290) CCD-Barcodes



Zeile 44 ( 290- 301) CCD-Barcodes



Zeile 45 ( 301- 309) CCD-Barcodes



Zeile 46 ( 309- 317) CCD-Barcodes



Zeile 47 ( 317- 323) CCD-Barcodes



Zeile 48 ( 323- 328) CCD-Barcodes



Zeile 49 ( 328- 337) CCD-Barcodes



Zeile 50 ( 337- 344) CCD-Barcodes



Zeile 51 ( 344- 353) CCD-Barcodes



Zeile 52 ( 353- 363) CCD-Barcodes



Zeile 53 ( 363- 372) CCD-Barcodes



Zeile 54 ( 372- 381) CCD-Barcodes



Zeile 55 ( 381- 382) CCD-Barcodes



Zeile 56 ( 382- 388) CCD-Barcodes



Zeile 57 ( 388- 394) CCD-Barcodes



Zeile 58 ( 394- 400) CCD-Barcodes



Zeile 59 ( 400- 404) CCD-Barcodes



Zeile 60 ( 404- 405) CCD-Barcodes



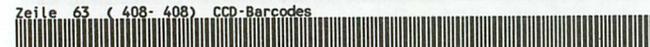
Zeile 61 ( 405- 406) CCD-Barcodes



Zeile 62 ( 406- 408) CCD-Barcodes



Zeile 63 ( 408- 408) CCD-Barcodes



Zeile 64 ( 408- 409) CCD-Barcodes



## TBP (Meschede) 1581

Computerclub Deutschland e.V.

Zeile 1 ( 1- 4) CCD-Barcodes



Zeile 2 ( 4- 10) CCD-Barcodes



Zeile 3 ( 10- 16) CCD-Barcodes



Zeile 4 ( 16- 23) CCD-Barcodes



Zeile 5 ( 23- 30) CCD-Barcodes



Zeile 6 ( 30- 41) CCD-Barcodes



Zeile 7 ( 41- 50) CCD-Barcodes



Zeile 8 ( 50- 62) CCD-Barcodes



Zeile 9 ( 62- 74) CCD-Barcodes



Zeile 10 ( 74- 86) CCD-Barcodes



Zeile 11 ( 86- 98) CCD-Barcodes



Zeile 12 ( 98- 106) CCD-Barcodes



Zeile 13 ( 106- 115) CCD-Barcodes



Zeile 14 ( 115- 127) CCD-Barcodes



Zeile 15 ( 127- 140) CCD-Barcodes



Zeile 16 ( 140- 152) CCD-Barcodes



Zeile 17 ( 152- 158) CCD-Barcodes



Zeile 18 ( 158- 163) CCD-Barcodes



Zeile 19 ( 163- 169) CCD-Barcodes



Zeile 20 ( 169- 175) CCD-Barcodes



Zeile 21 ( 175- 183) CCD-Barcodes



Zeile 22 ( 183- 186) CCD-Barcodes



## JACOBI (Schilli) 1582

Computerclub Deutschland e.V.

Zeile 1 ( 1- 3) CCD-Barcodes



Zeile 2 ( 3- 7) CCD-Barcodes

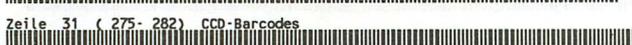


Zeile 3 ( 7- 18) CCD-Barcodes



Zeile 4 ( 18- 23) CCD-Barcodes



Zeile 5 ( 23- 34) CCD-Barcodes  
  
 Zeile 6 ( 34- 42) CCD-Barcodes  
  
 Zeile 7 ( 42- 49) CCD-Barcodes  
  
 Zeile 8 ( 49- 61) CCD-Barcodes  
  
 Zeile 9 ( 61- 70) CCD-Barcodes  
  
 Zeile 10 ( 70- 77) CCD-Barcodes  
  
 Zeile 11 ( 77- 87) CCD-Barcodes  
  
 Zeile 12 ( 87- 97) CCD-Barcodes  
  
 Zeile 13 ( 97- 108) CCD-Barcodes  
  
 Zeile 14 ( 108- 119) CCD-Barcodes  
  
 Zeile 15 ( 119- 129) CCD-Barcodes  
  
 Zeile 16 ( 129- 141) CCD-Barcodes  
  
 Zeile 17 ( 141- 153) CCD-Barcodes  
  
 Zeile 18 ( 153- 163) CCD-Barcodes  
  
 Zeile 19 ( 163- 172) CCD-Barcodes  
  
 Zeile 20 ( 172- 180) CCD-Barcodes  
  
 Zeile 21 ( 180- 192) CCD-Barcodes  
  
 Zeile 22 ( 192- 201) CCD-Barcodes  
  
 Zeile 23 ( 201- 210) CCD-Barcodes  
  
 Zeile 24 ( 210- 220) CCD-Barcodes  
  
 Zeile 25 ( 220- 229) CCD-Barcodes  
  
 Zeile 26 ( 229- 238) CCD-Barcodes  
  
 Zeile 27 ( 238- 245) CCD-Barcodes  
  
 Zeile 28 ( 245- 252) CCD-Barcodes  
  
 Zeile 29 ( 252- 263) CCD-Barcodes  
  
 Zeile 30 ( 263- 275) CCD-Barcodes  
  
 Zeile 31 ( 275- 282) CCD-Barcodes  
  
 Zeile 32 ( 282- 293) CCD-Barcodes  
  
 Zeile 33 ( 293- 303) CCD-Barcodes  


Zeile 34 ( 303- 312) CCD-Barcodes  
  
 Zeile 35 ( 312- 320) CCD-Barcodes  
  
 Zeile 36 ( 320- 331) CCD-Barcodes  
  
 Zeile 37 ( 331- 341) CCD-Barcodes  
  
 Zeile 38 ( 341- 350) CCD-Barcodes  
  
 Zeile 39 ( 350- 357) CCD-Barcodes  
  
 Zeile 40 ( 357- 363) CCD-Barcodes  
  
 Zeile 41 ( 363- 370) CCD-Barcodes  
  
 Zeile 42 ( 370- 370) CCD-Barcodes  


**DIVPLUS (Martini) 1583**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 6) CCD-Barcodes  
  
 Zeile 2 ( 6- 17) CCD-Barcodes  
  
 Zeile 3 ( 17- 25) CCD-Barcodes  
  
 Zeile 4 ( 25- 33) CCD-Barcodes  
  
 Zeile 5 ( 33- 42) CCD-Barcodes  
  
 Zeile 6 ( 42- 52) CCD-Barcodes  
  
 Zeile 7 ( 52- 60) CCD-Barcodes  
  
 Zeile 8 ( 60- 68) CCD-Barcodes  
  
 Zeile 9 ( 68- 76) CCD-Barcodes  
  
 Zeile 10 ( 76- 85) CCD-Barcodes  
  
 Zeile 11 ( 85- 96) CCD-Barcodes  
  
 Zeile 12 ( 96- 107) CCD-Barcodes  
  
 Zeile 13 ( 107- 118) CCD-Barcodes  
  
 Zeile 14 ( 118- 123) CCD-Barcodes  
  
 Zeile 15 ( 123- 134) CCD-Barcodes  
  
 Zeile 16 ( 134- 144) CCD-Barcodes  
  
 Zeile 17 ( 144- 153) CCD-Barcodes  
  
 Zeile 18 ( 153- 160) CCD-Barcodes  


# BARCODES

Zeile 19 ( 160- 168) CCD-Barcodes



Zeile 20 ( 168- 175) CCD-Barcodes



Zeile 21 ( 175- 182) CCD-Barcodes



Zeile 22 ( 182- 191) CCD-Barcodes



Zeile 23 ( 191- 202) CCD-Barcodes



Zeile 24 ( 202- 212) CCD-Barcodes



Zeile 25 ( 212- 224) CCD-Barcodes



Zeile 26 ( 224- 235) CCD-Barcodes



Zeile 27 ( 235- 242) CCD-Barcodes



Zeile 28 ( 242- 250) CCD-Barcodes



Zeile 29 ( 250- 259) CCD-Barcodes



Zeile 30 ( 259- 264) CCD-Barcodes



Zeile 31 ( 264- 268) CCD-Barcodes



Zeile 32 ( 268- 277) CCD-Barcodes



Zeile 33 ( 277- 287) CCD-Barcodes



Zeile 34 ( 287- 292) CCD-Barcodes



Zeile 35 ( 292- 301) CCD-Barcodes



Zeile 36 ( 301- 307) CCD-Barcodes



Zeile 37 ( 307- 313) CCD-Barcodes



Zeile 38 ( 313- 322) CCD-Barcodes



Zeile 39 ( 322- 331) CCD-Barcodes



Zeile 40 ( 331- 335) CCD-Barcodes



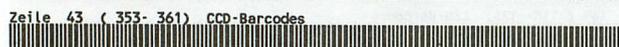
Zeile 41 ( 335- 344) CCD-Barcodes



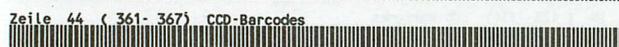
Zeile 42 ( 344- 353) CCD-Barcodes



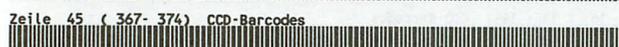
Zeile 43 ( 353- 361) CCD-Barcodes



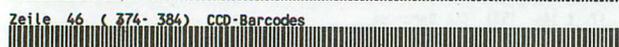
Zeile 44 ( 361- 367) CCD-Barcodes



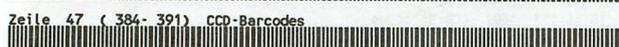
Zeile 45 ( 367- 374) CCD-Barcodes



Zeile 46 ( 374- 384) CCD-Barcodes



Zeile 47 ( 384- 391) CCD-Barcodes



Zeile 48 ( 391- 401) CCD-Barcodes



Zeile 49 ( 401- 408) CCD-Barcodes



Zeile 50 ( 408- 415) CCD-Barcodes



Zeile 51 ( 415- 419) CCD-Barcodes



Zeile 52 ( 419- 427) CCD-Barcodes



Zeile 53 ( 427- 437) CCD-Barcodes



Zeile 54 ( 437- 439) CCD-Barcodes



Zeile 55 ( 439- 447) CCD-Barcodes



Zeile 56 ( 447- 455) CCD-Barcodes



Zeile 57 ( 455- 463) CCD-Barcodes



Zeile 58 ( 463- 471) CCD-Barcodes



Zeile 59 ( 471- 474) CCD-Barcodes



Zeile 60 ( 474- 482) CCD-Barcodes



Zeile 61 ( 482- 491) CCD-Barcodes



Zeile 62 ( 491- 500) CCD-Barcodes



Zeile 63 ( 500- 508) CCD-Barcodes



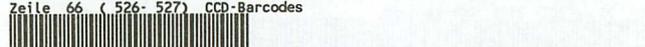
Zeile 64 ( 508- 518) CCD-Barcodes



Zeile 65 ( 518- 526) CCD-Barcodes



Zeile 66 ( 526- 527) CCD-Barcodes



## RS2 (Ristow) 1584

Computerclub Deutschland e.V.

Zeile 1 ( 1- 4) CCD-Barcodes



Zeile 2 ( 4- 7) CCD-Barcodes



Zeile 3 ( 7- 11) CCD-Barcodes



Zeile 4 ( 11- 17) CCD-Barcodes



Zeile 5 ( 17- 23) CCD-Barcodes



Zeile 6 ( 23- 29) CCD-Barcodes



Zeile 7 ( 29- 33) CCD-Barcodes



Zeile 8 ( 33- 41) CCD-Barcodes



Zeile 9 ( 41- 48) CCD-Barcodes



Zeile 10 ( 48- 53) CCD-Barcodes



Zeile 11 ( 53- 57) CCD-Barcodes



Zeile 12 ( 57- 62) CCD-Barcodes



Zeile 13 ( 62- 65) CCD-Barcodes



**AD (Schatz) 1585**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 5) CCD-Barcodes



Zeile 2 ( 5- 15) CCD-Barcodes



Zeile 3 ( 15- 24) CCD-Barcodes



Zeile 4 ( 24- 32) CCD-Barcodes



Zeile 5 ( 32- 40) CCD-Barcodes



Zeile 6 ( 40- 48) CCD-Barcodes



Zeile 7 ( 48- 56) CCD-Barcodes



Zeile 8 ( 56- 65) CCD-Barcodes



Zeile 9 ( 65- 74) CCD-Barcodes



Zeile 10 ( 74- 76) CCD-Barcodes



**ZW (Schatz) 1586**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 6) CCD-Barcodes



Zeile 2 ( 6- 12) CCD-Barcodes



Zeile 3 ( 12- 20) CCD-Barcodes



Zeile 4 ( 20- 28) CCD-Barcodes



Zeile 5 ( 28- 36) CCD-Barcodes



Zeile 6 ( 36- 41) CCD-Barcodes



**TA (Schatz) 1588**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 5) CCD-Barcodes



Zeile 2 ( 5- 11) CCD-Barcodes



**KO (Schatz) 1587**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 6) CCD-Barcodes



Zeile 2 ( 6- 16) CCD-Barcodes



Zeile 3 ( 16- 23) CCD-Barcodes



Zeile 4 ( 23- 32) CCD-Barcodes



Zeile 5 ( 32- 33) CCD-Barcodes



**ST87 (Martini) 1589**

Computerclub Deutschland e.V.

Zeile 1 ( 1- 4) CCD-Barcodes



Zeile 2 ( 4- 6) CCD-Barcodes



Zeile 3 ( 6- 12) CCD-Barcodes



Zeile 4 ( 12- 18) CCD-Barcodes



Zeile 5 ( 18- 24) CCD-Barcodes



Zeile 6 ( 24- 30) CCD-Barcodes



Zeile 7 ( 30- 36) CCD-Barcodes



Zeile 8 ( 36- 41) CCD-Barcodes



Zeile 9 ( 41- 49) CCD-Barcodes



Zeile 10 ( 49- 54) CCD-Barcodes



Zeile 11 ( 54- 62) CCD-Barcodes



Zeile 12 ( 62- 69) CCD-Barcodes



Zeile 13 ( 69- 74) CCD-Barcodes



Zeile 14 ( 74- 80) CCD-Barcodes



Zeile 15 ( 80- 89) CCD-Barcodes



Zeile 16 ( 89- 93) CCD-Barcodes



Zeile 17 ( 93- 102) CCD-Barcodes



Zeile 18 ( 102- 109) CCD-Barcodes



Zeile 19 ( 109- 115) CCD-Barcodes



Zeile 20 ( 115- 124) CCD-Barcodes



Zeile 21 ( 124- 132) CCD-Barcodes



Postvertriebsstück  
Gebühr bezahlt

**D 2856 F**

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

**CCD**

ISSN 0176-8735

**PRISMA**

**1988 Heft Nr. 1**

Zeile 22 ( 132- 139) CCD-Barcodes



Zeile 23 ( 139- 146) CCD-Barcodes



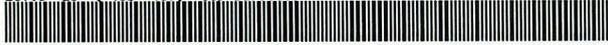
Zeile 24 ( 146- 155) CCD-Barcodes



Zeile 25 ( 155- 163) CCD-Barcodes



Zeile 26 ( 163- 169) CCD-Barcodes



Zeile 27 ( 169- 175) CCD-Barcodes



Zeile 28 ( 175- 182) CCD-Barcodes



Zeile 29 ( 182- 190) CCD-Barcodes



Zeile 30 ( 190- 197) CCD-Barcodes



Zeile 31 ( 197- 208) CCD-Barcodes



Zeile 32 ( 208- 215) CCD-Barcodes



Zeile 33 ( 215- 222) CCD-Barcodes



Zeile 34 ( 222- 229) CCD-Barcodes



Zeile 35 ( 229- 237) CCD-Barcodes



Zeile 36 ( 237- 246) CCD-Barcodes



Zeile 37 ( 246- 256) CCD-Barcodes



Zeile 38 ( 256- 265) CCD-Barcodes



Zeile 39 ( 265- 274) CCD-Barcodes



Zeile 40 ( 274- 282) CCD-Barcodes



Zeile 41 ( 282- 287) CCD-Barcodes



Zeile 42 ( 287- 293) CCD-Barcodes



Zeile 43 ( 293- 304) CCD-Barcodes



Zeile 44 ( 304- 309) CCD-Barcodes



Zeile 45 ( 309- 316) CCD-Barcodes



Zeile 46 ( 316- 322) CCD-Barcodes



Zeile 47 ( 322- 328) CCD-Barcodes



Zeile 48 ( 328- 334) CCD-Barcodes



Zeile 49 ( 334- 343) CCD-Barcodes



Zeile 50 ( 343- 351) CCD-Barcodes



Zeile 51 ( 351- 357) CCD-Barcodes



Zeile 52 ( 357- 363) CCD-Barcodes



Zeile 53 ( 363- 370) CCD-Barcodes



Zeile 54 ( 370- 377) CCD-Barcodes



Zeile 55 ( 377- 384) CCD-Barcodes



Zeile 56 ( 384- 390) CCD-Barcodes



Zeile 57 ( 390- 397) CCD-Barcodes



Zeile 58 ( 397- 403) CCD-Barcodes



Zeile 59 ( 403- 411) CCD-Barcodes



Zeile 60 ( 411- 417) CCD-Barcodes



Zeile 61 ( 417- 425) CCD-Barcodes



Zeile 62 ( 425- 434) CCD-Barcodes



Zeile 63 ( 434- 442) CCD-Barcodes



Zeile 64 ( 442- 448) CCD-Barcodes



Zeile 65 ( 448- 453) CCD-Barcodes



Zeile 66 ( 453- 459) CCD-Barcodes



Zeile 67 ( 459- 468) CCD-Barcodes



Zeile 68 ( 468- 476) CCD-Barcodes



Zeile 69 ( 476- 483) CCD-Barcodes



Zeile 70 ( 483- 494) CCD-Barcodes



Zeile 71 ( 494- 494) CCD-Barcodes

