

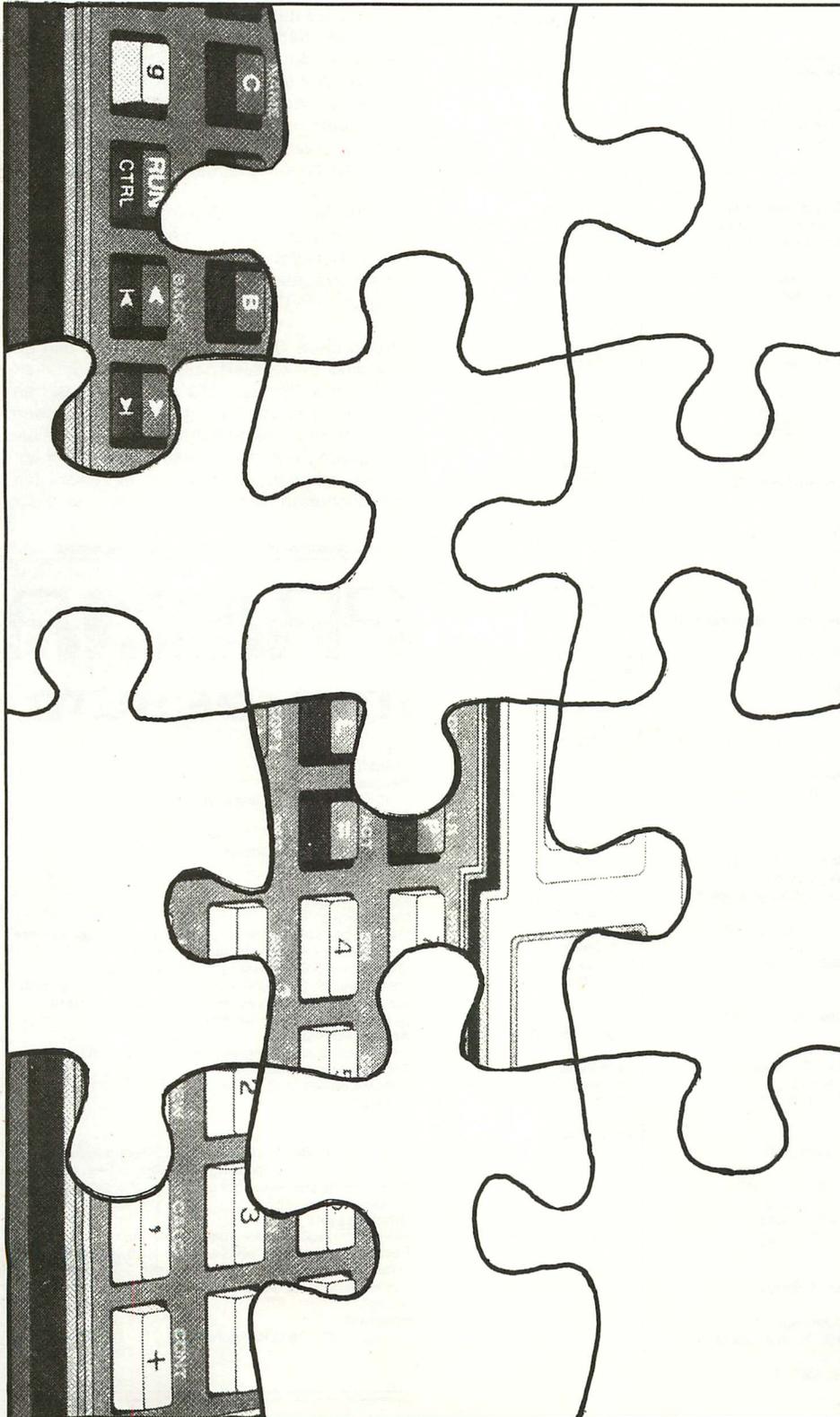
PRISMA

Computerclub Deutschland e.V.-Postfach 1924-7500 Karlsruhe

Prisma Herstellung-Limburger Str. 15-6242 Kronberg 2

Januar 1984 Nr.1

D 2856 E



Das "Kind im Manne" soll dieses neue Spiel ansprechen. Nach und nach werden wir dieses Puzzle vervollständigen. So habe ich die ewigen Sorgen um Titelbilder los und Ihr habt ein tolles Spielzeug.

Clubnachrichten
HP-75 Service
Ellipse
Tachometer
Sortieren
PRNT
CAT 2
Schachuhr
Memory
PSL
Neues LB
Streik
Neue QTH-Kenner
HP-IL
CCD-ROM
2. geodätische Hauptaufgabe (HP 15C)

Clubnachrichten

Video-Interface

Hiermit fordere ich alle CCDler, die einen HP-41 und das Video-Interface besitzen auf, sich mit mir zu einer Projektgruppe Video zusammenzuschließen.

Ziel der Gruppe soll es sein, Möglichkeiten und Grenzen des V.I.s zu ergründen; Software- und Hardwareentwicklungen rund ums V.I. zu fördern und Tips und Tricks auszutauschen. Ich möchte diese Gruppe organisieren, da es unfair wäre, die kostbaren PRISMA-Seiten für etwas in Anspruch zu nehmen, was (leider!) nur eine relativ geringe Zahl von CCDlern interessiert. Jeder, der glaubt, interessante Programme zu haben, die auf das V.I. zugeschnitten sind, soll sich mit einer Liste seiner Programme mit jeweils einer sehr kurzen Beschreibung des Zwecks an mich wenden. Wer Tips und Tricks loswerden möchte, der soll sie mir mit Beispiel zuschicken.

Jeder, der selbst Programme und/oder Tips eingesendet hat, kann gegen DM 1,20 in Briefmarken eine Liste der angebotenen Programme sowie die eingesendeten Tips und Tricks anfordern. Die Programme können dann von den jeweiligen Autoren selbst angefordert werden. Aber natürlich kann das auch anders organisiert werden. Ich warte auf Eure Briefe

Carlo Didier (1294)
127 Rue de Longuy
L 4831 Rodange

END.

Clubbibliothek

Nach recht langer Zeit habe ich endlich wieder einmal Gelegenheit, mich bei euch zu melden. Vorab ein persönliches Anliegen: Da ich umgezogen bin und ab sofort nicht mehr bei meinen Eltern wohne, wäre ich dankbar, wenn ich meine PRISMA zu meiner neuen Adresse gesandt bekommen würde (NEUE ADRESSE - SIEHE OBEN)! Vielen Dank im Voraus! So, nun aber "in medias res", zur Clubbibliothek:

(1) Entsprechend meinem Umzug hat sich auch die Adresse der Clubbibliothek geändert. Bitte so schnell als möglich in PRISMA vermerken.

(2) Wegen gleichigem Umzug, hoher Frequenzierung und interner Umorganisationsmaßnahmen der Clubbibliothek muß ich die Leistungen der Bibliothek bis zum 10.12.83 einstellen. Danach steht die Bibliothek den Clubmitgliedern wieder in vollem Umfang zur Verfügung. Erläuterung: Einige Bücher sind - bedingt durch das häufige Ausleihen - in einem dermaßen schlechten Zustand, daß es nicht mehr zumutbar erscheint, diese Bücher weiterhin zu verleihen, ohne den Totalverlust befürchten zu müssen. Ich werde diese Bücher "restaurieren" bzw. die Originale "Stillegen" und nur noch Kopien davon verleihen. Einen besonderen Engpass hat es bei dem Buch von W.C. Wickes "Synthetic Programming on the HP-41C" gegeben, so daß Wartezeiten von einigen Monaten entstanden sind. Um dies in Zukunft auffangen bzw. soweit als möglich verhindern zu können, lege ich folgendes fest:

(3) Die Ausleihdauer von Büchern beträgt maximal 10 Tage. Ausnahmen sind nur bei per-

Wichtige Clubadressen:

1. Vorsitzender Gruppenleiter München

Wolfgang Baites (155)
Guardinstr. 143
8000 München 70
☎ 089/7001540

2. Vorsitzender Artikelauswahl, Satz, Druck und Versand „PRISMA“

Rolf Hansmann (500)
Limburger Straße 15
6242 Kronberg II

Gruppenleiter Stuttgart Kassenwart

Günther Lehrke (239)
Wiesenstraße 85
7024 Filderstadt 1

1. Beisitzer Gruppenleiter Frankfurt Projektleiter „Synthetik“ Adressenverwaltung

Andreas Marktscheffel (69)
Nachtigallenweg 8
6246 Glashütten

2. Beisitzer

Wilfried Kötz ()
Im Aehlemaar 20
5060 Bergisch Gladbach

Nachsendedienst für prisma

Detlev Bock (2)
Petrikirchstraße 36
3400 Göttingen
Tel. 0551/33221

Magnetkarten-Bibliothek HP-75

Michael Hartmann (380)
Wendelsgrund 13
6690 St. Wendel

Magnetkarten-Bibliothek HP-41

Ralf Mulch
Hermannstädter Weg 17
6100 Darmstadt

Sammelbestellungen Magnetkarten Löbu 3 „Intelligente Spiele“

Ulrich Davertzhofen (3)
Osulfweg 23
4600 Dortmund 70

Projektleiter „Physik“

Olaf Gursch (120)
An der Schneise 36 A
1000 Berlin 27
Tel. 030/4310627

Clubbibliothek

Frank Allensen (117)
Uhlandsstraße 9
6365 Rosbach 3
Tel. 06007/1409

Löbu 4 „Baustatik“

Walter Pieperhoff
Walb.Osth.Wallstraße 10
4770 Soest
Tel. 02921/2895

Projektleiter „E-Technik“

Volker Wagner (1280)
Großhausberg 02-10-06
7743 Furtwangen

Projektleiter „Mathe“

Andreas Wolpers (349)
Parkstraße 11
7500 Karlsruhe

Gruppenleiter Köln Neuauflage prisma 80/81

Uwe Hommelsheim (251)
Unnauer Weg 50
5000 Köln 71

Projektleiter „Plotten“ HP-75 Service

Michael Hartmann (380)
Wendelsgrund 13
6690 St. Wendel

Osborne-Service

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

Gruppenleiter Göttingen

Wolfgang Ehrhardt
Im Großen Feld 11
3400 Göttingen

Gruppenleiter Berlin

Jörg Warmuth (79)
Wartburgstraße 17
1000 Berlin 62

Löbu 6, 7, 8 „Vermessungstechnik“

Wolfram Winter (568)
Eupenerstraße 5
5600 Wuppertal 1

Hardwareentwicklungen

Ulrich Jansen (283)
Mühlenstraße 16
5140 Erkelenz

HP 85 Service

Ingolf Seelemann
Bergstraße 22
4250 Bottrop

Projektleitung Statik

Daniel Sterner
Am Kellersbusch 8
6072 Dreieich 4

Tel.: 06133/84788

sönlicher Absprache mit mir und nur in Ausnahmefällen möglich. Wird mehrmals in eklatanter Weise gegen diese Regelung verstoßen, behalte ich mir vor, eine Sperrliste anzulegen und entsprechenden Personen keine Bücher mehr auszuleihen.

(4) Die Ausleihmodalitäten bleiben im großen und ganzen bestehen. Wer Ausleihwünsche hat, schreibt mir einen Brief mit:

- Mitgliedsnummer
- Genauer und vollständiger Adresse
- Rückporto für eine Büchersendung (1,80 DM) + (0,50 DM für die Versandtasche + eventuelle Restauration/Neukauf von Büchern (alles natürlich in Briefmarken!!))

Ich werde dann in der Reihenfolge des Eingangs die Bücher in einer Versandtasche an den Ausleiher verschicken. Pflügeliche Behandlung von Büchern und Versandtasche sind unabdingbare Voraussetzungen für den Ausleihservice. Bei Beschädigung/Verschmutzung der Bücher haftet der jeweilige Entleiher.

(5) Eine Ausleihe an Vereinsfremde ist aus ersichtlichen Gründen nicht möglich.

(6) Bücherlisten werden von mir nicht mehr an Einzelpersonen verschickt, sondern soweit möglich in PRISMA veröffentlicht.

Eine andere Sache, die ich im Rahmen der Clubbibliothek initiiert hatte, war die Bücherecke. Darin sollten Bücher zum Thema Taschenrechner/Computer vorgestellt, besprochen, kritisiert oder gelobt werden. Leider ist diese Einrichtung durch die Passivität unserer Mitglieder langsam aber sicher eingeschlafen. Ich möchte an dieser Stelle noch einmal dar-

PRISMA Impressum

Titel:
PRISMA
Herausgeber:
CCD - Computerclub Deutschland e.V.

Sitz:
Limburger Str. 15, 6242 Kronberg II

Verantwortlicher Redakteur:

Rolf Hansmann

Herstellung:

Rolf Hansmann

Manuskripte:

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

Anzeigenpreise:

Es gilt unsere Anzeigenpreisliste 2 vom Juni 1982.

Erscheinungsweise:

PRISMA erscheint 10 mal im Jahr.

Auflage:

2000

Bezug:

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

Urheberrecht:

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

Anschrift:

Postfach 1924, 7500 Karlsruhe.

END.

auf aufmerksam machen, daß es diese Einrichtung noch gibt und gleichzeitig alle PRISMA-Leser auffordern, sich daran zu beteiligen. Die "Beteiligung" kann auf verschiedene Arten erfolgen:

- Fertige Berichte (Rezensionen über Bücher)
- Vorschläge für Buchbesprechungen
- Erfahrungsberichte über Bücher
- oder in jeder anderen adäquaten Form.

Für eine rege Beteiligung von Seiten der Clubmitglieder wäre ich dankbar. Vielleicht könnte der eine oder andere auch noch ein Buch/eine Kopie der Clubbibliothek spenden!? Damit verbleibe ich mit den freundlichsten Grüßen und HP

Frank Altensen
Werner-Heisenberg-Weg 39/20
WB 17/233
8014 Neubiberg

.END.

HP 41 CX Handbücher

In Kürze können HP 41 CX Besitzer ihre Handbücher kostenlos über den Fachhandel beziehen. Bitte wartet das nächste Prisma ab. Dort werden die genauen Modalitäten veröffentlicht. Die Handbücher werden auf jeden Fall erst nach der Veröffentlichung im nächsten Prisma bei den Händlern eintreffen. Alle Anfragen, egal bei welcher Stelle, sind also sinnlos! Bereichert die Post nicht unnötig durch Telefonate die nichts bringen.

.END.

Wie kommt man an anderer Leute Geld?

In "Chip" 1/84 las ich folgende Anzeige: „HP 41 Wollen Sie das PRIVATE selbständig knacken? Tel. (0731) 42535“
Interessehalber rief ich an, um zu erkunden, was es damit auf sich hat und der Spaß kostet. Es meldete sich ein Herr Henning, der mir sagte, daß er ein Programm (eine Magnetkarte) habe, die den PRIVATE-Schutz aufhebe. Für diese Karte verlangte er 100,- DM!!
Auf mein Fragen erklärte er, daß es durchaus Leute gebe, die für solchen Schwachsinn soviel Geld ausgeben.
Ich kann sein Verfahren nicht beurteilen, da ich es nicht kenne. Allzugroß scheinen seine Kenntnisse auf dem Gebiet jedoch nicht zu sein: Er sagte mir, daß es Unterschiede beim PRIVATE des C und des CV gebe, was falsch ist.

Eduard Wulff

.END.

Ortsgruppe Karlsruhe

Am 11.1.84 fand mit Sekt und 20 anwesenden Mitgliedern die Gründungsversammlung der Ortsgruppe Karlsruhe statt. Die Ortsgruppe Karlsruhe will alle Mitglieder im Großraum Karlsruhe betreuen. Auch Baden-Badener sind herzlich willkommen. Wir treffen uns am 2. Mittwoch jedes Monats um 19.00 Uhr im Gruppenraum 2 des Jugend- und Begegnungszentrums Karlsruhe am Kronenplatz 1.
Ansprechpartner: Heinrich Lang, Stephanienstr. 78, 7500 Karlsruhe, ☎ 0721/21344

Ulrich Schwaderlap

.END.

Clubbörse

Suche gebrauchten HP-82166 Konverter. Angebote an U. Schwaderlap, An den Berken 34, 5840 Schwerte 6 oder an Gerhard Kockerbeck, Studtstr. 23, 4600 Dortmund 1, ☎ 0231/147471

Programme einschl. Formblätter für Nachweise zur neuen Wärmeschutzverordnung u. DIN A108 Info Rückp. 1,30 DM
M. Steinhauer, Zentmarkweg 10, 6000 Frankfurt 90

Verkaufe wegen Systemwechsel 1HP-IL-Modul 220,-, 1 HP-IL-Drucker + 6 Rollen Papier für 900,-, 1 HP-IL-Kassettenlaufwerk + 3 Kassetten für 900,-, 1 HP-Battery-Pack für 30,-. Alle Teile wie neu, wenig gebraucht, original verpackt.
Gerhard Borimann, Mühlackerstr. 5, 7132 Illingen, ☎ 07042/21262

Kartenleser inc. ca. 100 Magnetkarten DM 350,-
Volker Lang (130), ☎ 07138-8710 (abends)

Verkaufe: IL-Laufwerk (HP82161, 2 Mon. alt). IL- und Time-Modul (beide neuwertig), VB 1450,- DM
Rainer Maier (1708), ☎ 0511/320069 ab 19h

Verkaufe Doppelmodul X-Function/Time für 270,- DM.
P. Behrend, ☎ 0421/505117

Suche f. Eigenbedarf alle PPC J. u. HP-Key-Notes zu kopieren. Bitte per PK melden, auch teilweise, danke.
T. Flöck, Scheler Str. 59, 5650 Solingen 19

Suche billigst: unbeschriebene Magnetkarten (nur Original HP)
Gerhard Siewert, Mozartstr. 14, 6450 Hanau 1

Verkaufe HP41CV, suche IL-Laufwerk
Holger Langer, Bismarckstr. 11, 4980 Bünde, ☎ 05223/2343

HP 41C + Time + Quad zus. 650,-; HP 41 CV + XFM + Time zus. 750,-; HP 34C + Ladeg. + Zubeh. 250,-; S-Memorys 25,-, Games 50,-, Time-M. 100,-
Josef Hayda, ☎ 02238/54785

Verkaufe HP-41 Plotter-Rom: DM 140,-; Fachliteratur (Chip 82/83, PE...), Fachbücher... Port Extender CT 4100: neuwertig DM 350. Suche Softwareauschmöglichkeiten auf allen Gebieten.
H.W. Greiner, ☎ 0851/35722 (abends)

Suche HP 41-Barcodelesestift
Heinz Kling, Gresgen 36, 7863 Zell im Wiesental, ☎ 07625/7231 (nach 18.00 Uhr)

HP-41C + X-Function- + Time-Modul + Quad-Ram für ca. 900 DM zu verkaufen!
Knebel (363), ☎ 07231/67658

Besitze KONTRON PSI 80DS2, an KI und CAE im Elektronik-Lab interessiert. Suche Kontakt zu Gleichgesinnten m. gl. App.
D. Klage, 221 c/o GFS, Pf. 2, NL 1755 Petten

HP-41C (getuned: 60 % schneller) + QRAM + TIME-Modul + XF-Modul komplett für 700,- DM zu verkaufen.
Gregor Westermann, Am Weidenbach 23, 5000 Köln 1, ☎ 0221/326404 (ab 19 Uhr)

Verkaufe: 1 Doppel-Memory-Modul, aus 2 MM herstellen lassen und 8 Std. benutzt sowie 1 Einfach-Memory-Modul 82106A, kompl. DM 150,-
Wolfgang Fuchs, Silcherstr. 7, 7145 Markgröningen

Verkaufe HP 41 C (8 Mon. alt), 1 Memory-Modul für DM 330,- sowie 1 Time-Modul, 1 XF-Modul (5 Mon. alt) für je DM 100,- und 1 IL-Modul (unbenutzt) für DM 150,-
Siegfried Heinrich, ☎ 07253/21554 ab 14 Uhr

Verkaufe HP 41 CV getunt + Akku: 600,- DM; IL-Modul + IL-Printer (2 Mon. alt) + Papier: 1050,-; XF-, XM-, Timemodul: a 120,-; Cardreader 300,-; Wand: 220,-; PPC-Rom: 300,-; Games- u. Standard-Modul: a 50,-; 350 Magnetkarten mit Superprogrammen (auch synthetisch): 200,-; 2000 Seiten Literatur (HP-41): 120,-
Hansi Greiner, ☎ 0851/35722

Reste-Verk.: Koffer f. 41-System, CT-Port-Ext., 19 Digicass, 120 Magnetk. (unben.), Akku+ Ladeg., Kabel, Papier, Statmodul.
Brick, ☎ 0421/341857

.END.

Korrekturen zum CCD Artikel von Werner Dworak

Prisma 83.10.28

Bedingt durch die neue Möglichkeit, Texte aus allen Rechnern mit V 24 Schnittstelle direkt in die Fotosatzmaschine zu überspielen, sind bei den ersten Versuchen kleine Textstellen verschwunden. Hier die fehlenden Teile des Artikels. Der erste Absatz gehört auf die Seite 83.10.28 in die erste Spalte ungefähr in der Mitte.

Der zweite Absatz gehört auf die Seite 83.10.29, zweite Spalte ungefähr nach einem Drittel.

Und die "größeren" Rechner waren immer noch recht teuer und wenig leistungsfähig. So war es folgerichtig, daß ich in den Hewlett-Packard-Anwenderclub-Oliver-Rietschel eintrat, als ich davon eine Anzeige sah. Schon bald wurde mir bewußt, daß Olivers deutlich sein Taschengeld durch den HPAC aufbesserte, aber mir war ein funktionierender Club lieber als einer, der sang- und klanglos eingeht wie der MICAC von Heinrich Schnepf aus Köln.

Einige "neue" Mitglieder meinten recht direkt, die OSBORNE Gruppe solle sich abspalten und einen eigenen Verein aufmachen, während von den "alten" Mitgliedern - so weit ich weiß - alle fest dem CCD die Treue halten, auch wenn sie mit dem HP41 gar nichts mehr machen.

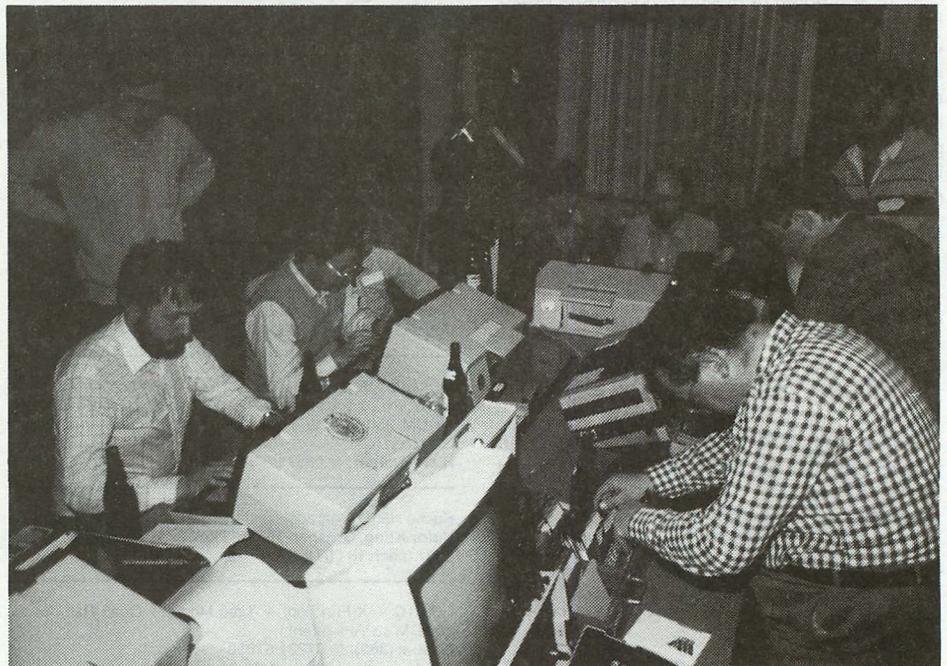
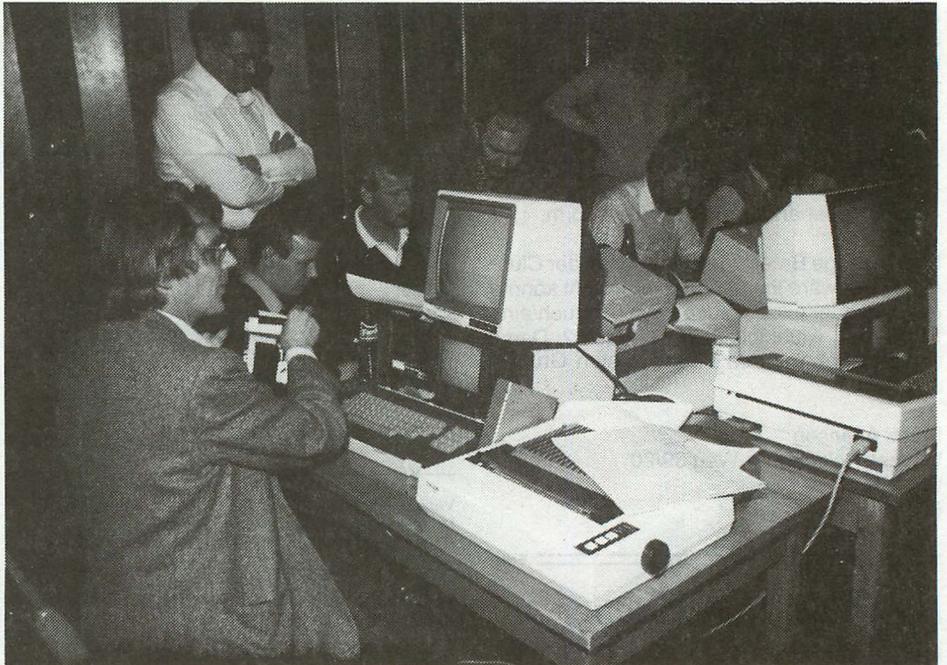
Sollte sich ein eigenständiger OSBORNE-Club auf tun, würde ich sicherlich dort Mitglied werden und auch aktiv mitarbeiten, aber auf keinen Fall in den Vorstand gehen. Dem CCD würde ich allerdings stets verbunden bleiben.

.END.

DAS OSBORNE - USER TREFFEN 1983

Für alle Osborne Besitzer im CCD war das Wochenende vom 11.11 bis 13.11 etwas ganz besonderes. Es sollte kein Faschingsanfang sein, sondern eine Zusammenkunft aller AKTIVEN Osborne - User zum Zwecke des Erfahrungsaustauschs. Die Vorbereitungen dafür begannen schon Monate vorher und unsere Telefonkosten schnellten rapide in die Höhe, denn so ein Treffen mußte sorgfältig geplant werden. Ralf Kern besorgte für uns ein gutes Hotel, und Wolfgang Fritz konnte einen hervorragenden Raum bei der Studentenvereinigung Germania besorgen (das Bier dabei nicht zu vergessen). Viele bildeten dann Fahrgemeinschaften und wir trafen uns am Abend des 11.11 in Karlsruhe. Es war schon ein tolles Gefühl, Menschen das erste mal zu sehen, die man schon lange vom Telefon her kannte. Hier nun einige Daten: Wir waren 14 Personen, 6 Osborne 1, 2 Osborne Executive, 2 Drucker, 2 Monitore, ca 400 Disketten und unzählige Fla-

schen Bier!! Es gab soviel zu erzählen und so viel Erfahrungen auszutauschen, daß der erste Abend bis zum Samstag Morgen um 2.30 Uhr dauerte! Am nächsten Tag (Samstag) wurde unser Treffen durch unsere CCD Mitgliederversammlung unterbrochen die von 10.00 Uhr bis 18.00 Uhr dauerte, so daß wir nach einer kräftigen Pizza wieder zur Tat schritten. Einige Mitglieder hatten sich mit Softwareproblemen beschäftigt, für die ein anderer bereits eine Lösung hatte oder es wurden Hardwarelösungen ausgetauscht. An dieser Stelle sollten einige Projekte erwähnt werden: Werner entwickelte ein Programm zum selektiven lesen und schreiben von Daten an jede beliebige Stelle der Diskette; Peter Ochsen schrieb ein perfektes Datenübertragungsprogramm, das sowohl Ascii als auch Hex Dateien überträgt; Ralph Kern entwickelte ein Programm, welches das nachträgliche patchen von Ascii Dateien mit mathematischen Sonderzeichen erlaubt und viele andere nützliche Routinen tauchten auf den Disketten der Osborne Freaks auf. Da wir sowohl einen ITOH als auch einen EPSON Drucker zur Verfügung hatten, konnte jeder die Leistungsfähigkeit der beiden Geräte vergleichen. Insbesondere war interessant, das wir drei Generationen von Osborne Geräten auf den Tischen zu stehen hatten: vom alten 1E bis zum neuen Executive, vom ROM 1.2 bis zum ROM 1.44C, vom blauen Monitor bis zum orange Monitor reichte die Palette. Für Programmierer die in vielen Anwendungen sowohl einen deutschen, als auch einen englischen Zeichensatz gebrauchen können (wegen der eckigen Klammer!!), wurde von Werner ein spezielles Eprom entwickelt. Wir mußten auch an dem zweiten Abend wieder feststellen, welch riesig große Softwarepalette ein CPM Computer bieten kann und das der OSBORNE immer wieder seine Leistungsfähigkeit und Portabilität unter Beweis stellte. Es ist immer noch das preiswerteste und mit der besten Standardsoftware ausgestattete System das auf dem Markt ist. Dieser Abend dauerte dann auch wieder bis um 3.30 Uhr! Am Sonntagmorgen trafen wir uns noch einmal im Hotel um den verblieben Rest der Probleme zu diskutieren. Da Werner als Leiter der Gruppe langsam zu viel Aufgaben hatte, haben wir uns die Aufgaben geteilt: Werner (607) als Leiter der Gruppe stellt weiterhin die Infos (BILD) zusammen und verschickt sie an alle Osborne - User



Dietmar (1778) kümmert sich um die neu zu uns stoßenden Osborne (BILD) User und versorgt sie mit den letzten Infos

Reinhard (888) macht die Osborne Mitgliederverwaltung und (BILD) druckt die Adressauf-

kleber für den Versand der Disketten; außerdem kümmert er sich um die Werbung für die Osborne Gruppe

Für alle, die sich zu uns gesellen wollen, hier unsere Kontaktadressen:

Werner Dworak, Allewind 51,7900 Ulm (07304.3274)
 Dietmar Pfortner, Postfach 174,4133 Neukirchen-Vluyn (02845.1922)
 Reinhard Loessner, Kuerkoppel 3,2300 Kiel 1 (0431.332605)

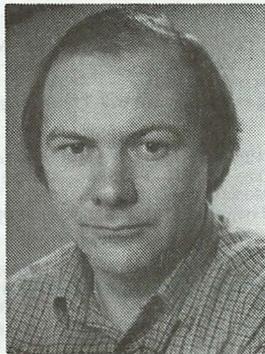
Diese Treffen war ein voller Erfolg für alle Beteiligten und an dieser Stelle sollte man allen denjenigen Danken, die weder Mühe noch Zeit gescheut haben, um den passenden Rahmen für dieses Wochenende zu schaffen. Zum Schluß noch einmal Informationen für diejenigen die uns noch nicht kennen: wir sind zur Zeit 60 Osborne-User und sind stark im wachsen. Wir schicken Software und Infos direkt an Werner, der die Beiträge zusammenstellt und monatlich eine Diskette mit 180 kb heißer Info an jeden Osborne-User verschickt. Wir wollen uns einmal im Jahr zum Erfahrungsaustausch treffen und planen das nächste Treffen auf der Hannover Messe. Wer noch Fragen hat, soll sich bei einem der drei oben erwähnten melden.
 Reinhard (888)



Werner



Reinhard



Dietmar

HP-75 Service

MATH-PAC 75-15015 - Ein Erfahrungsbericht

Seit ca. zwei Monaten bin ich nun Besitzer des Math-Pac für den HP 75C und möchte nun meine Meinung zu diesem Produkt auch an andere weitergeben. Dazu fällt mir als HP-41 Besitzer natürlich sofort ein, daß dieses Modul wesentlich besser geworden ist als das Mathe-Modul für den 41er. Sämtliche Befehle sind nämlich als BASIC-Worte implementiert, die bis auf zwei Ausnahmen (Nullstellensuche und Integration) auch über das Tastenfeld aufgerufen werden können. Und da alles in Assembler programmiert ist, ergibt sich ein erheblicher Geschwindigkeitsvorteil gegenüber BASIC-Programmen.

Aber nun zu den eingebauten Funktionen: Als erstes gibt es ein paar zusätzliche Funktionen, z.B. Hyperbelfunktionen und deren Umkehrfunktionen, Logarithmus zur Basis 2 und zu beliebigen Basen, Funktionen zum Runden bzw. Abschneiden von Stellen vor oder nach dem Komma sowie eine kombinierte Fakultät/Gammafunktion (wie beim HP-15C).

Als zweites sind zwei Funktionen zur Basisumwandlung enthalten. Es kann zwischen dem Dezimalsystem und Dual/Oktal/Hexadezimalsystem umgewandelt werden, allerdings müssen die Argumente Integerzahlen =0 sein, und Zahlen zur Basis 2, 8 und 16 werden als Strings behandelt. Man kann also in anderen Zahlensystemen nicht rechnen.

Als drittes gibt es jede Menge Matrizenbefehle. Sie beginnen fast alle mit dem Schlüsselwort MAT, das ich im folgenden weglasse. Für die Ein/Ausgabe gibt es da zunächst einmal die Befehle INPUT, DISP und PRINT, um ohne lästige FOR-Schleifen Matrizen in einer einzigen Zelle ein- und auszugeben. Bei DISP und PRINT kann mit der USING-Anweisung auch formatierte Ausgabe erfolgen. Bei MAT INPUT ist es leider im Gegensatz zum normalen INPUT-Befehl nicht möglich, einen Eingabeaufforderungsstring anzugeben, und auch die Korrektur bereits eingegebener Werte ist mit diesem Befehl nicht zu bewerkstelligen. Des weiteren gibt es Befehle, um allen Elementen einer Matrix einen konstanten Wert, z.B. 0, zuzuweisen, oder komplette Matrizen durch einfache Zuweisungen zu kopieren. Und dann gibt es noch ein REDIM. Hiermit kann man eine Matrix neu dimensionieren, allerdings mit folgenden Einschränkungen: Die Matrix muß vorher in einem DIM-Statement dimensioniert worden sein, die Anzahl der nach REDIM in der Matrix enthaltenen Elemente darf die der nach dem DIM-Statement nötigen Anzahl nicht überschreiten. Evt. wegfallende Elemente bleiben im RAM gespeichert, können aber auch durch ein zweites REDIM wieder zugänglich gemacht werden. Außerdem gibt es Befehle zur Addition, Subtraktion, Multiplikation, Skalarmultiplikation, Inversion, Transposition und einen Befehl zur Berechnung des Kreuzproduktes zweier R^3 Vektoren, zur Berechnung von Spalten- und Zeilensummen, der Determinante und diverser Matrixnormen, zur LU-Zerlegung einer Matrix und zum Lösen von linearen Gleichungssystemen. Zum INV Befehl (Invertierung einer Matrix) sollte hier noch angemerkt werden, daß der Rechner auch singuläre Matrizen invertiert (das geht natürlich nur mit Schummeln), aber dies verhindert immerhin einen unbeabsichtig-

ten Programmabbruch. Das Vorliegen eines solchen Falles kann mit DETL überprüft werden, Ergebnis von DETL ist die Determinante der zuletzt bearbeiteten Matrix. Allerdings kann man mit INV auch Matrizen invertieren, die beinahe singulär sind. Die erhöhte Genauigkeit, die bei fast allen Berechnungen des Math-Pac stattfindet (ca. 16 Stellen), kann einem aber auch manchmal einen Streich spielen. So kommt es z.B. bisweilen vor, daß als Ergebnis einer Determinantenberechnung statt 0 Ergebnisse wie 1.62...E-12 herauskommen. Weiter bietet das Mathe-Modul die Möglichkeit, Matrizen mit zwei Elementen als komplexe Zahlen zu betrachten und mit einer Viel-

zahl von Befehlen zu verknüpfen. Es läßt sich allerdings, wie bei Matrizen, immer nur eine Operation auf einmal ausführen. Zeilen wie $MAT A=A+INV(B)*LUFAC(T(B)-SYS(A,B))$ frißt der Rechner nicht. Das muß erst fein säuberlich zerlegt werden. Zu den für komplexe Zahlen verfügbaren Operationen gehören neben Trivialitäten wie Addition und Multiplikation auch Winkel- und Hyperbelfunktionen, Logarithmen, Wurzeln und Exponentiation sowie Umwandlung von Polar- in kartesische Darstellung und umgekehrt. Außerdem sind viele Matrizenfunktionen auch für komplexe Matrizen vorhanden, u.a. Multiplikation, Transposition, Inverse und Determinante, Lösen von li-

Als erstes Programm 'CHI':

```

25/1
10 DEF FNX2(X2,V)      ! Integral der chi^2 Verteilung von 0 bis x2 bei v Freiheitsgraden
20 Z0,Z1=1 Z2=2      ! Formeln siehe Handbook of mathematical functions
30 Z1=Z1*X2/(V+Z2)    IF Z0+Z0+Z1 THEN Z0=Z0+Z1 Z2=Z2+2 GOTO 30
40 Z0=Z0/EXP(X2/2)*(X2/2)^(V/2)/FACT(V/2)
50 FNX2=1-Z0
60 END DEF
70 X,V=1
80 INPUT "x^2= ",STR$(X);X
90 INPUT "v = ",STR$(V);V
100 DISP FNX2(X,V); GOTO 80
    
```

Nun das Programm 'NPV':

```

26/2
10 OPTION BASE 1
20 REAL C(100)      ! Cash-Flow-Beträge
30 INTEGER N(100)   ! Cash-Flow-Anzahl
40 REAL S,Q         ! Hilfsvariable
50 INTEGER I,N,P    ! Anzahl der Cash-Flow-Gruppen
60 DEF FNE1(I)      ! Eingabe Beträge
70 DISP "Cash-Flow-Betrag";I; INPUT "=" ,STR$(C(I));S
80 FNE1=S
90 END DEF
100 DEF FNE2(I)     ! Eingabe Anzahl
110 INPUT "Anzahl der Cash-Flows = ",STR$(N(I));S
120 FNE2=S
130 END DEF
140 DEF FNN(Q)      ! Nettokapitalwert
150 IF Q=1 THEN S=DOT(C,N) GOTO 210 ! Berechnungsschleife bespringen
160 P,S=0          ! Summe und Periodenzahl initialisieren
170 FOR I=1 TO N   ! Schleife durch alle Cash-Flow-Gruppen
180 S=S+C(I)*(1-Q^(-N(I)))/(Q-1)/Q^(P-1) ! i-ten Cash-Flow diskontieren
190 P=P+N(I)      ! Perioden summieren
200 NEXT I
210 FNN=ROUND(S,2) ! Auf ganze Pfennige runden
220 END DEF
230 PUT CHR$(204)  ! Numerisches Tastenfeld einschalten
240 INPUT "Anzahl der Cash-Flow-Gruppen ? ";N
250 IF N<100 THEN DISP "Höchstens 100!" GOTO 240 ! Neue Eingabe
260 IF N > 1 THEN DISP "Mindestens 1!" GOTO 240 ! Neue Eingabe
270 MAT C=ZER(N)   ! Felder initialisieren und neu dimensionieren
280 FOR I=1 TO N   ! Anfang der Eingabeschleife
290 C(I)=FNE1(I)   ! Eingabe des i-ten Betrages
300 N(I)=FNE2(I)  ! Eingabe der entsprechenden Anzahl
310 NEXT I
320 DISP "Änderungen (Ja(1)/Nein(2)) ";
330 K$=KEY$ IF NOT POS("12",K$) THEN 330 ! Eingabe überprüfen
340 DISP K$       ! Eingabe anzeigen
350 IF K$="2" THEN 410 ! keine Änderung; Sprung zur Programmwahl
360 INPUT "Welche Cash-Flow-Gruppe ? ";I
370 IF I > N THEN DISP "Falsche Eingabe; höchstens ";N GOTO 360 ! Neue Eingabe
380 IF I < 1 THEN DISP "Falsche Eingabe; mindestens 1" GOTO 360 ! Neue Eingabe
390 C(I)=FNE1(I) N(I)=FNE2(I) ! Änderung der gewählten Gruppe
400 GOTO 320      ! Neue Änderungsabfrage
410 DISP " IRR(1), NPV(2) oder Ende(3) ";
420 K=KEY$ IF NOT POS("123",K$) THEN 420 ! Eingabeüberprüfung
430 DISP K$      ! Eingabe anzeigen
440 ON POS("123",K$) GOTO 510,460,450 ! Auswahl IRR, NPV, Ende
450 PUT CHR$(140) ENDE ! Numerisches Tastenfeld ausschalten
460 INPUT "Diskontzinssatz ? ";Q ! Nettokapitalwert
470 DISP TAB(10);"Bitte warten"
480 Q=FNN(1+Q/100)
490 DISP "NPV= "; ! Anzeige Interner Zinsfuß
500 GOTO 560
510 DISP TAB(10);"Bitte warten"
520 Q=FNROOT(1,1.2,FNN(Q)) ! Berechnung der Nullstelle der NPV-Funkt.
530 IF FNN(Q)≠0 THEN DISP "Keine Lösung gefunden! (RTN)" GOTO 570 ! Ergebnis prüfen
540 Q=(Q-1)*100 ! Zinssatz berechnen
550 DISP "IRR= ";
560 DISP USING "m8dz.2d,k" ; Q;" (RTN)" ! Abfrage Änderungen/zum Menü
570 IF NUM(KEY$)≠13 THEN 570 ELSE 320
    
```

nearen Gleichungssystemen etc. Die Matrizenfunktionen arbeiten übrigens mit Feldern beliebiger Größe (wie zu erwarten war).

Nach den Matrizenfunktionen nun noch ein paar weitere Funktionen für den täglichen Gebrauch:

PROOT berechnet alle (auch komplexe) Nullstellen von Polynomen, über R.Laut Handbuch kann man Polynome bis 360 (mit zusätzlichem RAM bis 560) Grades damit lösen. In diesen Größenordnungen muß man allerdings etwas Geduld aufbringen. Ich hab's mal mit einem Polynom 264. Grades probiert und bin auf eine Rechenzeit von 9h gekommen. Eine etwas praxisnähere Angabe: ein Polynom 10. Grades dauert ca. 20 Sekunden.

Die Funktion FNROOT liefert zu einer benutzerdefinierten Funktion und zwei nicht notwendigerweise verschiedenen Anfangswerten (fast immer) eine Nullstelle. Falls keine Nullstelle gefunden wurde, kann der zurückgegebene Wert auch ein lokales Minimum, aus einem Bereich, in dem die Funktion konstant ist, oder Infimum sein, falls sich der Rechner beim Suchen "verlaufen" hat. Auch eine Integralfunktion fehlt nicht, sie benutzt offenbar den gleichen Algorithmus wie der HP-15C bzw. das PPC-ROM, nur werden beim HP-75 maximal 65535 Funktionswerte berechnet und mit 16 Stellen summiert.

Um die Ergebnisse von FNROOT und INTEGRAL zu überprüfen, stehen diverse Kontrollvariablen zur Verfügung.

Zum Schluß gibt es noch eine Funktion zur finiten Fourier-Transformation. Zur Anwendung dieser Funktion muß die Anzahl der Stützstellen eine Zweierpotenz sein.

Auf weitere Funktionen möchte ich hier nicht eingehen, da mir das notwendige Wissen fehlt.

Ein paar Nachteile hat das Ganze natürlich auch:

Einige Matrizenfunktionen belegen während ihrer Ausführung größere Mengen Speicherplatz (mir hat's bis jetzt aber immer gereicht), und die Zeit, bis der Rechner eine eingegebene BASIC-Zeile verdaut hat, wächst deutlich. Der TRANSFORM-Befehl braucht z.B. bis zu 60 % mehr Zeit. Dafür laufen dann aber viele Programme etwas (deutlich) schneller.

Insgesamt bietet das Math-Pac über 80 neue BASIC-Befehle und zehn zusätzliche Fehlermeldungen.

Noch eine Bemerkung zum Handbuch: Im Gegensatz zur 41er Serie findet hier ein etwas stabiler wirkendes Ringbuch Verwendung. Das hat zwar drei Ringe, aber das Rechnerhandbuch paßt trotzdem nicht rein...

Das Handbuch ist meines Wissens bisher nur in Englisch erhältlich, umfaßt ca. 140 Seiten und ist als Mathe-Buch natürlich nicht zu gebrauchen.

Auf die etwas komplexeren Algorithmen wird kurz eingegangen, eine ausführliche Beschreibung jedes Algorithmus würde sich auf den Preis des Moduls wohl recht negativ auswirken. Man kann dem Handbuch jedenfalls grob entnehmen, was der Rechner so macht, und das dürfte den meisten wohl reichen. Außerdem will man ja auch noch ein bißchen probieren!! Oder?

Für den mobilen Einsatz liegt wie bei der 41er Modulen noch eine Kurzanleitung bei, in der die Syntax der einzelnen Befehle und die Fehlermeldungen erklärt sind.

Persönliches Abschlußurteil: Mir gefällt's!!!!

Einige Laufzeitangaben (gerundete Werte, Matrizen mit Zufallszahlen besetzt):

Funktion	n	10	12	14	16	18	20
LUFAC2.7	4.5	7.0	10.3	14.2	19.5		
INV	7.3	12.4	19.6	29.0	40.4	55.4	
SYS	4.9	7.7	11.0	15.6	21.3	27.8	
DET	2.7	4.6	6.9	10.2	14.5	19.4	
*	8.0	13.7	21.8	32.8	46.6	64.0	
+	0.4			1.1	1.6		
PROOT	21.5	32.7	37.5	54.3	63.6	74.2	

Die Zeiten beziehen sich auf Operationen mit (n,n) Matrizen, bei + bzw. * werden natürlich zwei Matrizen verknüpft. Ansonsten kann man nur mit INTEGRAL erzielen, diese lassen sich aber durch Vorgabe der Genauigkeiten regulieren.

Nun möchte ich Euch noch zwei Programme vorstellen, in denen Funktionen des Mathe-Moduls verwertet werden und die als praktische Ergänzung des Artikels über das Mathe-Modul von Anreas Wolpers dienen sollen. Die Programme sind so gehalten, daß sie sich in ihrem Listing selbst erklären.

Text-Files

1. Aufbau normaler Textfiles

1.1 Der Header

Der Header besteht wie bei BASIC-Files aus 28 Bytes. Auch der Inhalt des Headers eines Text-Files und des Headers eines BASIC-Files unterscheiden sich nur unwesentlich. Der Unterschied liegt lediglich im Bereich 3 (siehe unter I.BASIC-Files). Hier ändert sich das 5. Byte vom Wert 254 zum Wert 190 und das 6. Byte ändert seinen Wert von 66 zu 84, was als ASCII-Zeichen das Zeichen T für Text-File zur Folge hat.

Gehen wir also gleich auf den eigentlichen Hauptteil des Text-Files ein.

1.2 Aufbau einer Textzeile:

Jede Textzeile beginnt mit einer Zeilennum-

mer. Die Codierung erfolgt wie bei BASIC-Files gemäß dem BCD-Code in umgekehrter Reihenfolge. So bedeutet HEX 10 00 umgedreht die Zeile 0010, also Zeile 10. Diesen 2 Bytes folgt die Längenangabe des folgenden Textes, und zwar Anzahl der Zeichen von Byte 4 bis zum letzten Byte einer Zeile. Da kein eigenes Byte vorgesehen ist zum Bezeichnen des Endes einer Zeile wie bei BASIC-Files das Byte mit dem Hex-Wert OE, gibt dieses 3. Byte allein Auskunft über die Länge des Textes und somit auch über die Länge der Zeile. Als Beispiel betrachte man nun in Abb. 3 den Beginn der Zeile 10 mit folgenden Bytes in Hexform 10 00 OF, und in Dezform 16 00 15. Die 15 gibt nun die Anzahl der Zeichen an. Das 4. Byte ist das 1. Zeichen, das Byte mit dem Hex-Wert 29 das 15. Byte. Mit diesem Zeichen endet auch die Zeile in dem unten in Abb. 3 aufgeführten Ausdruck des Textes. Danach beginnt die nächste Zeile. Dieser Aufbau der Zeilen ist durchgehend gleich.

1.3 Zusammenfügen der Textzeilen und des Headers zur Gesamtheit:

Header und Hauptteil (=Text) werden durch 10 Nullbytes getrennt. Diese Nullbytes stehen für die 5 Pointer, von denen jeder 2 Bytes beansprucht (1.Byte: Anzahl der zu überspringenden Zeichen; 2. Byte: Übertrag, falls Byte 1 255 überschreitet). Diese Pointer spielen hier jedoch eine untergeordnete Rolle (wie bei BASIC-Files). Nach diesen Pointern kommt der eigentliche Text, in dem die Textzeilen direkt aufeinanderfolgen, da sie ja durch das 3. Byte jeder Textzeile voneinander getrennt sind. Die letzte Textzeile hat bei allen Text-Files den gleichen Aufbau. Sie zeigt dem Rechner, daß hier das Ende des Textfiles erreicht ist. Diese Ende-Marke besteht aus 5 Bytes mit folgenden Hex-Werten: 99 A9 02 8A OE. Was danach kommt, ist für den Rechner uninteressant, da er diesen Teil beim Rechnen nicht mehr berücksichtigt.

Spur 1 Record 118

```

210 136 20 1 190 04 239 206 104 157 67 72 73 32 32 32 32 0 0 0 0 0 0 0 0 0 0 0 0 0 0 16 0 15 32
02 00 14 01 0E 54 EF CE 60 90 43 40 49 20 20 20 20 00 00 00 00 00 00 00 00 00 00 00 10 00 0F 20
      T
60 69 70 32 70 70 00 50 40 00 50 44 86 41 32 0 15 32 90 40 44 90 49 61 49 32 64 32 90 50 61 50
44 45 46 20 46 4E 58 32 20 58 32 2C 56 29 20 00 0F 20 5A 30 2C 5A 31 30 31 20 40 20 5A 30 20 5A
D E F F N X 2 ( X 2 , V ) Z 0 , Z 1 = 1 @ Z 2 = 2
40 0 64 32 90 49 61 90 49 42 00 50 47 40 86 43 90 50 41 32 64 32 73 70 32 90 40 35 90 40 43 90
30 00 40 20 5A 31 30 5A 31 2A 58 32 2F 20 56 20 5A 32 29 20 40 20 49 46 20 5A 30 23 5A 30 20 5A
0 @ Z 1 = 2 1 * X 2 / ( V + Z 2 ) @ I F Z 0 # Z 0 + Z
49 32 84 72 69 78 32 90 48 61 90 48 43 90 49 32 64 32 90 50 61 90 50 43 50 32 64 32 71 79 04 79
31 20 54 40 45 4E 20 5A 30 30 5A 30 20 5A 31 20 40 20 5A 32 30 5A 32 20 32 20 40 20 47 4F 54 4F
1 T H E N Z 0 = Z 0 + Z 1 @ Z 2 = Z 2 + 2 @ G O T O
32 51 48 64 0 39 32 90 48 61 90 48 47 69 00 00 40 00 50 47 50 41 42 40 00 50 47 50 41 94 40 06
20 33 30 40 00 27 20 5A 30 30 5A 30 2F 45 50 50 20 50 32 2F 32 29 2A 20 58 32 2F 32 29 5E 20 54
3 0 @ Z 0 = Z 0 / X P ( X 2 / 2 ) * ( X 2 / 2 ) ^ ( V
47 50 41 47 70 65 67 84 40 86 47 50 41 00 0 10 32 70 70 00 50 61 49 45 90 40 96 0 0 32 69 70
2F 32 29 2F 46 41 43 54 20 56 2F 32 29 50 00 0A 20 46 4E 58 32 30 31 20 5A 30 60 00 00 20 43 4E
/ 2 ) / F A C T ( V / 2 ) P F N X 2 = 1 - Z 0 ` E N
68 32 60 69 70 112 0 6 32 00 44 86 61 49 120 0 24 32 73 70 00 05 04 32 34 120 94 50 61 32 34 44
40 20 44 45 46 70 00 06 20 58 2C 56 31 00 00 18 20 49 4E 50 55 54 20 22 70 5E 32 30 20 22 2C
D D E F P X , V = 0 I N P U T " x ^ 2 = " , S T R # ( X ) ;
83 84 02 36 40 00 41 59 00 144 0 24 32 73 70 00 05 04 32 34 118 32 32 61 32 34 44 83 04 02 36 40
53 54 32 24 20 50 29 30 50 90 00 10 20 49 4E 50 55 54 20 22 76 20 20 30 20 22 2C 53 54 52 24 20
S T R # ( X ) ; X I N P U T " v = " , S T R # ( V ) ;

```

```

DEF FN X2(X2,V)
Z0,Z1=1 @ Z2=2
Z1=Z1*X2/(V+Z2) @ IF Z0#Z0+Z1 THEN Z0=Z0+Z1 @ Z2=Z2+2 @ GOTO 30
Z0=Z0/EXP(X2/2)*(X2/2)^(V/2)/FACT(V/2)
FN X2=1-Z0
END DEF
X,V=1
INPUT "x^2= ",STR#(X);X
INPUT "v = ",STR#(V);V
DISP FN X2(X,V); @ GOTO 00

```

Aufbau von keys-Textfiles

2.1 Der Header

Der Header eines Keys-Files unterscheidet sich vom Header eines normalen Text-Files überhaupt nicht. Deshalb gehen wir auch gleich über zum nächsten Punkt.

2.2 Aufbau einer Tastenzuordnung:

Beispiel für eine typische Tastenzuordnung:
:1;|

Diese bedeutet, daß der Taste, deren Zeichen den Dezimalwert 1 hat, das Zeichen | zugeordnet wird. Wie ist nun dies im Rechner codiert. Zuerst stehen in den beiden ersten Bytes im BCD-Code ähnlich der Codierung für die Zeilennummer der Dezimalwert des Zeichens, dessen Taste neu belegt werden soll. Danach kommt im 3. Byte die Länge dieser Zuordnung. Mit dem 4. Byte beginnt die Neuebelegung der Taste. So würde für das oben gewählte Beispiel die Codierung wie folgt aussehen:

01 00 02 59 124. 01 00 geben die Taste 1 an, d.h. daß das Zeichen ersetzt wird. 02 gibt die Länge der Zuordnung an (2 Zeichen: ;|). 59 ist der Dezimalwert für das Zeichen ;. 124 ist der Dezimalwert für die neue Tastenzuordnung, hier das Zeichen |. Danach kommt die nächste Tastenzuordnung.

2.3 Die Zusammenfassung der Einzelteile erfolgt wie in 1.3, wobei jetzt die Textzeile durch eine Tastenzuordnung gemäß 2.2 ersetzt wird. Auch hier wird das Ende des Files durch die Bytekombination 99 A9 02 BA OE angezeit.

Magnetkartenbibliothek HP-75C

Liste aller in der Magnetkartenbibliothek erhältlichen Programme:

Codierung: Programmnummer/Anzahl der Karten

1/1	prisma 83.3.09	PLOTTEN
2/1		GGT
3/1		Tangentenverfahren
4/1		Nim-Spiel
5/1		Integration
6/1		Fußball
7/2	prisma 83.4.07	Hangman
8/1		Quersumme
9/1		Einserspiel
10/1	prisma 83.5.06	BADE
11/1	prisma 83.5.07	BABA
12/1		Textversuch
13/3	prisma 83.6.07	HP-41 Simulation
14/2	prisma 83.6.08	MASTERM
15/1		TEWR
16/3	prisma 83.7.08	Einser
17/3		Arithmetik
18/2		Matrixinversion
19/3	prisma 83.8.05	HP-41 Simulation verbesserte Version
20/3	prisma 83.8.06	Lottoschein
21/1	prisma 83.9.04	Plotter als Printer (beide Programme)
22/1		Term
23/2	prisma 83.9.05	PRDIR
24/2	prisma 83.9.06	PRREC

Solch eine Liste werde ich zweimal im Jahr veröffentlichen. Ich hoffe, daß bei Programmbestellungen nun die Anzahl der Magnetkarten, die Ihr beigefügt habt, stimmt. Denkt bitte an den selbstadressierten und frankierten Rückumschlag. Vielen Dank!!!!!!!

Michael Hartmann (380)
Wendelsgrund 13
6690 St. Wendel

I/O Utilitis

Als wichtigste schriftliche Erweiterung der Handbücher des HP 75 muß man das Lösungsbuch I/O-Utilitis bezeichnen. Ich möchte nun eine kleine Einführung in dieses Lösungsbuch geben. Danach möchte ich in den nächsten Ausgaben von PRISMA großen Wert auf diesen Bereich legen. Auch in diesem Heft werde ich schon ein Programm vorstellen. Und zwar werde ich bei diesen Programmen den Ablauf der HP-IL-Befehle genau erläutern. Als Voraussetzung betrachte ich die Einführung von Wolfgang Baltes in die HP-IL-Funktionsweise.

Nun zu dem Lösungsbuch I/O-Utilitis:

Zu dem Lösungsbuch gehört die LEX-Magnetkarte HPILCMD5 (HPIL-Commands). Durch diese Magnetkarte erhält der HP 75 die Möglichkeit der Kommunikation mit irgendeiner Peripherieeinheit innerhalb der HPIL-Schleife (ob jetzt Talker oder Listener). Mit dieser Magnetkarte ist es nun möglich, den HP 75 so zu programmieren, daß er HP-IL-Kommandos an irgendeine Einheit sendet und Daten von dieser abrufen. Nun komme ich zu den Befehlen:

1. Das Senden von Informationen zu einer HP-IL-Einheit:

Der SENDIO-Befehl wird gebraucht, um Kommando und Daten an eine HP-IL-Einheit zu senden. SENDIO kann sowohl direkt vom Tastenfeld als auch in einem BASIC-Programm aufgerufen werden.

Syntax: SENDIO ':Code der Einheit', 'Liste der Kommandos', 'Liste der Daten': Code der Einheit: Der Name, der der Einheit durch den ASSIGN IO-Befehl gegeben wurde.

'Liste der Kommandos': Eine Liste von HP-IL-Kommandos, durch Kommata getrennt. Die Kommandos werden durch bestimmte Abkürzungen vertreten (siehe Bericht von Wolfgang Baltes)

'Liste der Daten': Ein Charakter-String, der als Daten übermittelt wird.

Ist der Code der Einheit nicht bekannt, genügt der String ''.

Mit Hilfe des SENDIO-Kommandos wird hauptsächlich eine Einheit aktiviert und bestimmt zum Listener oder Talker.

Genauere Beschreibungen werden bei den Programmen folgen, da dies hier zu weit führen würde.

2. Der SEND?-Befehl

Der SEND?-Befehl benötigt keine Parameter. Nach seiner Ausführung erhält man eine ganze Zahl. Ihr Wert ist die Position in der Datenliste des Zeichens, das infolge einer Fehlermeldung nicht vollständig mit dem letzten SENDIO-Befehl übermittelt wurde. Normalerweise erhält man nach Ausführung des SEND?-Befehls den Wert 0.

3. Das Empfangen von Informationen von der IL-Schleife

Der ENTIO\$-Befehl wird benötigt, um Daten von einer Einheit zu empfangen. Man erhält ein Zeichen mit einem bestimmten Wert.

Syntax: ENTIO\$(':Code der Einheit', 'Liste der Kommandos')

Die angesprochene Einheit muß im Talker-

Modus sein. um antworten zu können. Mit Hilfe dieses Kommandos kann man z.B. die Einheiten einer Schleife genau einordnen, da verschiedene Einheiten verschiedene Identifikationscodes aussenden. Auch hier kann, wenn der Code der Einheit nicht bekannt ist, folgendermaßen abgekürzt werden: ENTIO\$('', 'Liste der Kommandos')

Nun wollen wir gleich ins Praktische einsteigen. Betrachten wir zunächst folgendes Programm: Listing 3

Dieses Programm dient dazu, die Schleife, in der beliebige Einheiten zusammengeschaltet sind, zu initialisieren.

Zum Aufbau des Programms:

Zuerst wird die Schleife durch das Kommando 'ASSIGN IO ':zz'' in einen gelöschten Zustand gebracht, d.h. die Einheiten in der Schleife tragen keinen Namen mehr und können in diesem Zustand nicht mehr angesprochen werden.

Mit dem Befehl 'SENDIO', 'aa, aad1', '' werden die Einheiten aktiviert und in einen beliebigen Zustand versetzt, sowie wird der Schleife mitgeteilt, daß die nun folgende Adressierung bei der Einheit 1 beginnt (Einheit 1 ist die Einheit, die direkt mit dem OUT-Ausgang des HPIL-Steckers des Rechners verbunden ist. Mit dem Kommando 'B\$=ENTIO\$('', 'ad' '&STR\$(I)&', 'sai')' wird die Einheit I (zu Beginn ist I=1) veranlaßt, als Talker zu arbeiten und die Identifikation als Zeichen zu senden (auf diese Identifikation gehe ich in der Anmerkung ein). Anhand des Zeichenwertes dieses Identifikationszeichens, der für jede Einheit spezifisch ist, kann man erkennen, um welche Einheit es sich handelt. Indem man nun jeder Einheit einen Code zuweist und diese Codes zu einer Zeichenkette zusammenfaßt, in der nun alle Codes aller Einheiten der Schleife vorhanden sind, und diese dann mit dem ASSIGN IO-Befehl an den Rechner weitergibt, hat man die Schleife automatisch adressiert.

Dieses Programm benutzt man am besten bei jedem Einschalten des Rechners. So muß man sich nicht mehr darum kümmern, wenn man Einheiten austauscht, daß die Schleife wie gewünscht funktioniert.

Anmerkung:

Die Werte der Identifikationszeichen (D.h. der numerische Wert des Zeichens, das eine Einheit als Identifikation aussendet) sind für jede Einheit verschieden. Deshalb hier eine Liste der gebräuchlichsten Einheiten mit den Werten und Zeichen:

Einheit	Zeichen	Numerischer Wert des Zeichens
HP 82161A	o	16
HP 82162A	Leerzeichen	32
HP 82163B	0	48
HP 82905B	!	33
HP 2671G	#	35
HP 82166A	@	64
HP 82169A	C	67
HP 7470A	,	96

Dipl.-Ing. Wolfgang Niese (CCD-Nr. 557) # Knappschaftshöhe 5 # 4300 ESSEN 14

```

AUTOIO      B 424 Bytes 27.04.83/03.05.83 # HP-75C
100 ! ** Peripherie ansprechen **
110 INTEGER A,B,I
120 DIM B$(2),X$(3),Y$(12),Z$(11)
130 ASSIGN IO ':zz'
140 SENDIO '', 'aa, aad1', ''
150 Y$='' @ I=1
160 B$=ENTIO$('', 'ad' '&STR$(I)&', 'sai') @ B=POS(Y$, 'tv')
170 IF B$='' THEN GOTO 240
180 IF I=1 THEN Z$='' ELSE Z$=', '
190 IF NUM(B$)=48 THEN X$='tv'
200 IF NUM(B$)=16 THEN X$='ca'
210 IF NUM(B$)=33 THEN X$='pr'
220 Y$=Y$&Z$&X$
230 I=I+1 @ GOTO 160
240 ASSIGN IO Y$
250 A=POS(Y$, 'pr')
260 IF A>0 THEN PRINTER IS 'pr'
270 IF B>0 THEN DISPLAY IS 'tv'
    
```

Programme und Berichte

Umfangsberechnung einer Ellipse 1157

Das Programm berechnet mit Hilfe der Reihenentwicklung (Gl.1) den Umfang einer Ellipse. Es ist besonders für große numerische Exzentrizitäten gedacht (größer als ca. 0,995), bei denen die üblichen Näherungsformeln größere Abweichungen gegenüber der korrekten Lösung liefern.

Ausführung

- 1 XEQ"EL" a?
2. große Halbachse eingeben a R/S b?
3. kleine Halbachse eingeben b R/S
4. Anzeige num. Exzentrizität (PSE) 0,... ANZ?
5. Anzahl Reihenglieder n-1 R/S
6. Ausgabe U=....

Peter Radloff, Schlesierweg 23, 3250 Hameln

Da die Angabe der Anzahl der Reihenglieder zwar verhindert, daß die Rechenzeit unerträgliche Werte annimmt, andererseits aber auch Auswirkungen auf die Genauigkeit hat, hier als Ergänzung ein Programm, das die Reihenentwicklung abbricht, wenn der letzte Summand im gewählten Anzeigeformat (FIX) nicht mehr darstellbar ist. Dabei muß jedoch u.U. mit erheblichen Rechenzeiten gerechnet werden. Die Ausführung ist wie bei obenstehendem Programm, bei 5 wird hier aber die Reihenentwicklung mit VIEW angezeigt, und dies Programm heißt "ELLIPSE".

$$Gl.1 \quad U = 2a\pi \left[1 - \left(\frac{a}{2}\right)^2 \varepsilon^2 - \left(\frac{1.3}{2.4}\right)^2 \varepsilon^4 \frac{1}{3} - \left(\frac{1.3.5}{2.4.6}\right)^2 \varepsilon^6 \frac{1}{5} - \dots \right]$$

```

01*LBL "EL"      37 *
02 CLRG         38 STO 02
03 "a?"        39 *
04 PROMPT      40 RCL 04
05 "b?"        41 /
06 PROMPT      42 ST+ 07
07 X<Y?        43 DSE 06
08 X<>Y        44 GTO 01
09 STO 01      45 RCL 07
10 /           46 CHS
11 ACOS        47 1
12 SIN         48 +
13 PSE         49 2
14 X↑2         50 *
15 STO 00      51 RCL 01
16 "ANZ?"      52 *
17 PROMPT      53 PI
18 STO 06      54 *
19 -           55 "U="
20 STO 04      56 ARCL X
21 1           57 AVIEW
22 STO 02      58 END
23 STO 03
24*LBL 01
25 2
26 ST+ 04
27 ST+ 05
28 RCL 04
29 RCL 05
30 /
31 RCL 03      LBL*EL
32 *           END
33 STO 03      82 BYTES
34 X↑2         LBL*ELLIPSE
35 RCL 02      END
36 RCL 00      90 BYTES
    
```

```

01*LBL "ELL"    26 RCL [
IPSE"          27 2
02 "a ?"       28 +
03 PROMPT      29 /
04 "b ?"       30 X↑2
05 PROMPT      31 *
06 X<Y?        32 RCL [
07 X<>Y        33 X↑2
08 STO J       34 1
09 /           35 -
10 ACOS        36 *
11 SIN         37 ST+ J
12 STO \       38 VIEW J
13 PSE         39 2
14 CLX        40 ST+ [
15 STO [       41 X<>Y
16 RCL J       42 RND
17 PI          43 X≠0?
18 *           44 GTO 01
19 ST+ X       45 RCL J
20 STO J       46 "U="
21 STO L       47 ARCL X
22 CF 21       48 FS? 55
23*LBL 01      49 SF 21
24 LASTX      50 AVIEW
25 RCL \       51 END
    
```

Andreas Wolpers (349)

.END.

HP-41 CX als Tachometer

Zur Überprüfung der Tachometeranzeige muß man an den 500m-Markierungen eine beliebige Taste drücken, der HP-41 CX zeigt dann die Durchschnittsgeschwindigkeit der letzten 500 bzw. 1000 Meter an. Das Programm benutzt die "GETKEYX"-Anweisung des HP-41 CX, bei der die Zahl im X-Register bestimmt, wieviele Sekunden auf eine Antwort gewartet wird. Wenn $X \geq 0$, reagiert der Rechner auf das Loslassen einer Taste; wenn $X \leq 0$ (wie hier), reagiert er auf das Drücken einer Taste. Das Programm benutzt die Stack-Register und das Alpha-Register, keine Speicherregister.

```

PRP "TACHO"    1158
12:21 28.11
01LBL "TACHO"
02 FIX 0
03 .5
04 TIME
05 HR
06 "KEY AT 500 M"
07 AVIEW

OLBL 00
09 -90
10 GETKEYX
11 X<> T
12 TIME
13 HR
14 R^
15 RCL Y
16 RDN
17 -
18 /
19 " "
20 ARCL X
21 " "
22 2
    
```

```

23 *
24 ARCL X
25 AVIEW
26 .5
27 R^
28 GTO 00
29 .END.
    
```

```

PRP "TACHO"
12:22 28.11
01LBL "TACHO"
FIX 0 .5 TIME HR
"KEY AT 500 M" AVIEW
    
```

```

OLBL 00
-90 GETKEYX X<> T
TIME HR R^ RCL Y RDN
- / " " ARCL X
" 2 * ARCL X
AVIEW .5 R^ GTO 00
.END.
    
```

Toni Lerchenfeld (1209)

.END.

HP-41 CX: numerisches und alphabetisches Sortieren mit einem Programm 1159

Die neuen Vergleichsfunktionen des HP-41 CX bieten die Möglichkeit, sowohl numerische

```

01*LBL "SX"
02*LBL 00
03 RCL X
04 RCL IND      X

05*LBL 01
06 X>NN?
07 X<> IND      Y

08 ISG Y
09 GTO 01
10 STO IND      Z

11 RCL Z
12 ISG X
13 GTO 00
14 CLST
15 TONE 8
16 END

LBL* SX
END
34 BYTES
    
```

als auch Alphadaten zu vergleichen. Deshalb kann man mit dem Programm "SX" nicht nur numerische Daten, sondern auch Alphadaten sortieren. Bevor das Programm "SX" gestartet werden kann, muß eine Kontrollzahl bbb,eee in der Art wie für ISG, PRREGX, WDTAX ... in das X-Register eingegeben werden. Kontrollzahl bbb,eee bedeutet, daß bbb das erste und eee das letzte Register ist, das in die Sortierung miteinbezogen wird. Dann Programm starten. Daß der Sortiervorgang abgeschlossen ist, wird durch einen Ton angezeigt. Es wird in aufsteigender Reihenfolge sortiert; wenn man Zeile 06 in X < NN? ändert, wird in absteigender Reihenfolge sortiert. Das Programm selbst benutzt nur die Stack-Register, LASTX und Alpha-Register bleiben unverändert.

Gerhard Kruse (1004)

.END.

PRNT 1160

Das vorliegende Programm entstand auf Anregung von Gerhard Siewert (52). Mit diesem Programm ist es möglich, 2-Byte-Barcodes (Paper-Wand-Befehle) auf dem alten, nicht modifizierten Drucker auszugeben. Dies geschieht in 2 Durchgängen, da der Druckbuffer die maximal erforderlichen 56 Bytes nicht zwischenspeichern kann.

Das Programm erwartet die Anzahl der zu druckenden Barcodes in Speicher 00, die dezimalen Werte der Bytes in der Form ppp,qqq in den Speichern 03-nn. (ppp=1.Byte, qqq=2.Byte) Eingaberoutinen siehe beigefügte Programme WND und PXR. Nach dem Start des Programms wird mit ALPHA? abgefragt, ob die Barcodes kommentiert werden sollen. Ein J oder eine andere Alpha-Eingabe werden als ja interpretiert, ein R/S ohne vorherige Eingabe als nein.

Im Folgenden werden einige Hilfszeilen gedruckt, die zur Positionierung des Papiers für den zweiten Durchgang verwendet werden. Danach folgen die 2. Bytes der Barcodes (die Ausgabe erfolgt von hinten nach vorne), gegebenenfalls mit der Aufforderung ALPHA, die gewünschten Kommentare einzugeben. Nach der Aufforderung PAPIER R/S wird das Papier zurückgezogen. Es ist darauf zu achten, daß sich das Papier nicht seitlich verschiebt. Die oberste gestrichelte Linie muß mit der oberen Kante der Plexiglasabdeckung abschließen. Mit manuellen ADV's wird die zweite Linie vorgeschoben. Hier erfolgt eine Kontrolle und letzte Korrektur, bevor das Programm mit R/S erneut gestartet wird. Die fehlenden Pfeile (S. Abb 1 u. 2) werden gedruckt, das Programm stoppt mit XEQ 09 R/S. An den Pfeilen (Abb.3) erkennt man, ob das Papier richtig positioniert wurde. Ist man damit zufrieden, erfolgt nach R/S der Ausdruck der fehlenden 1. Bytes. Andernfalls muß das Papier erneut zurückgezogen werden. Ein erneuter Ausdruck der zweiten Pfeilspalte erfolgt mit XEQ 09. Dieser Vorgang kann mehrmals wiederholt werden.

Die Positionierung des Papiers erfordert einige Übung, die erzeugten Barcodes können sich jedoch sehen (lesen) lassen.
HP (happy printing)

- 01*LBL "PRNT" 06 "ALPHA?"
- 02*LBL 01 07 AON
- 03 CF 08 08 TONE 9
- 04 SF 21 09 STOP
- 05 CF 23 10 AOFF

```

11 FS? 23            75 "      <-"
12 SF 08            76 PRA
13 SF 12            77 PRA
14 "INTENSITY 5"    78 PRA
15 PRA              79 PRA
16 "> ..... "      80 PRA
17 PRA              81 "XEQ 09    R/S"
18 ADV              82 TONE 9
19 ADV              83 PROMPT
20 ADV              84 ADV
21 ADV              85 RCL 00
22 ADV              86 FRC
23 ADV              87 3
24 PRA              88 +
25 ADV              89 STO 00
26 ADV              90 GTO 05
27 "....->X-"      91*LBL 12
28 PRA              92 FC? 09
29 PRA              93 GTO 14
30 "      ->"       94 2
31 PRA              95 ENTER↑
32 PRA              96 1 E-3
33 PRA              97 GTO 06
34 PRA              98*LBL 14
35 PRA              99 21
36 ADV              100 RCL IND 00
37 SF 09            101 +
38 RCL 00           102 INT
39 2                103 SKPCOL
40 +                104 RTN
41 1 E3            105*LBL 13
42 /                106 0
43 3                107 ENTER↑
44 +                108 1 E-3
45 STO 00          109 GTO 06
46*LBL 05          110*LBL 11
47 XEQ 12          111 FC? 09
48 CLX             112 ADV
49 X<>F            113 FC? 09
50 RCL IND 00      114 RTN
51 INT             115 "ALPHA"
52 1 E3            116 CF 12
53 /                117 AON
54 X<> IND 00      118 TONE 7
55 FRC             119 STOP
56 1 E3            120 AOFF
57 *                121 PRA
58 7 E-3           122 SF 12
59 XEQ 06          123 RTN
60 X<> 2            124*LBL 06
61 ST- IND 00     125 X<>Y
62 FC? 09          126 X<>F
63 XEQ 13          127 0
64 PRBUF          128 ENTER↑
65 FS? 08          129 127
66 XEQ 11          130*LBL 02
67 ISG 00          131 ACCOL
68 GTO 05          132 FS? IND
69 FC?C 09        133 ACCOL
70 GTO 15          134 X<>Y
71 " PAPIER"      135 ACCOL
72 TONE 5          136 X<>Y
73 PROMPT         137 FS? IND
74*LBL 09          138 DSE Z

```

```

139 X<> X            147 ADV
140 ISG T            148 ADV
141 GTO 02           149 BEEP
142 RTN              150 END
143*LBL 15
144 ADV              LBL*PRNT
145 ADV              END
146 ADV

```

R00= ANZAHL BARCODE
SCHLEIFENZAehler
R01-R02= UNUSED
R03-R<3+N>= PPP,QQQ
PPP: 1.BYTE QQQ: 2.BYTE

INHALT NACH 1.DURCHGANG:
R03-R<3+N>= R,PPP
INHALT NACH 2.DURCHGANG:
R03-R<3+N>= S,00R
S: ANZAHL 1SER IM 1.BYTE
R: ANZAHL 1SER IM 2.BYTE

STACK WAEREND ACCOL-

ROUTINE <LBL06> :
T: SCHLEIFENZAehler
Z: ZAEHLT 1SER/BYTE
Y: 0
X: 127

30.09.1983 - 11:43:24
(E) b r (B P)

327 BYTES

1. Durchgang

INTENSITY 5
>

>

```

- - - - -><-
- - - - -><-
->
->
->
->
->
->

```

```

■■■■■■■■
BEEP
■■■■■■■■
X=0?
■■■■■■■■
PI
■■■■■■■■
LASTX

```

Abb. 1

59*LBL 10
ST+ IND Z R+ R+ E1
X<>Y X=Y? GTO 04 94
X<>Y X<=Y? GTO 05 189
X<>Y X<=Y? GTO 06

136*LBL B
8,013005 STO 18 "MEM"
ASTO 19 "I-O" ASTO 20
CF 27 "SPIELERANZ?"
BEEP 5 PROMPT X>Y?
GTO B SF 27 STO 07

TONE 2 TONE 0 XEQ 14
RCL 18 REGSWAP
232*LBL 14
RCL 10 SEEKPTA 27
ACCHR 37 ACCHR RCL 09
47 - X<0? SF 02
LASTX FS?C 02 32
ACCHR RCL 08 ACCHR
GETREC OUTA CLA RTN

328*LBL "R"
RCL 09 61 X<=Y? RTN
SF 17 E "EC" ST+ 09
OUTA GTO "R"

75*LBL 04
32 GTO 07

152*LBL 12
STO IND X DSE X GTO 12
CLX X<>F E STO 06
XEQ "RE" TONE IND 06
PROMPT

254*LBL 11
RCL 08 RCL 09 +
RCL 13 RCL 14 + X=Y?
SF 01 X=Y? RTN BEEP
E-3 ST+ IND 06 CLA
ARCL 19 XEQ 11 ARCL 20
XEQ 11 ARCL 20 XEQ 14
RCL 18 REGSWAP ARCL 19
XEQ 11 ARCL 20 XEQ 11
ARCL 20 XEQ 14 DSE 06
"♦" RTN

339*LBL "L"
RCL 09 33 X>Y? RTN 8
DSE 09 ACCHR GTO "L"

78*LBL 05
32 + GTO 07

82*LBL 06
65 +

163*LBL "N"
CLA ARCL 19 XEQ 11
FC? 01 XEQ D 27 ACCHR
62 ACCHR TONE IND 06
PROMPT

286*LBL 11
RCL 10 SEEKPTA GETREC
RCL 11 AROT ATOX 32
XTOA RCL 11 CHS E -
AROT DELREC INSREC
CLA RTN

348*LBL "0"
RCL 08 33 X>Y? RTN
SF 17 E "EA" ST- 08
OUTA GTO "0"

85*LBL 07
XTOA ALENG 15 RTN

90*LBL 03
196 RCL IND X 9821 *
.211327 + FRC
STO IND Y E3 * INT
900 X<>Y X>Y? GTO 03
RCL Z MOD X=0? GTO 03
RTN

175*LBL 11
RCL 08 32 - STO 10
47 RCL 09 - X<=0?
SF 03 CHS 15 MOD
STO 11 RDN ST+ 10 E
FS?C 03 ST+ 10 RCL 10
SEEKPTA GETREC RCL 11
AROT 32 ATOX SF 17
CLA XTOA "HED" OUTA
"WEITER" X=Y? RTN
STO 12 27 ACCHR 60
ACCHR FS?C 01 GTO 11
SF 01 RCL 18 REGMOVE
RTN

304*LBL D
ISG 06 "+" RCL 07
RCL 06 X>Y? E STO 06
"E<XNSPIELER " FIX 3
ARCL IND 06 "I-EZNY"
SF 17 OUTA -4 AROT
ASHF ATOX ATOX 38
STO 08 46 STO 09 RTN

359*LBL "U"
RCL 08 44 X<=Y? RTN
SF 17 E "EB" ST+ 08
OUTA GTO "U"

111*LBL 11
4 STO 00

114*LBL 09
XEQ 03 E

117*LBL 08
E RCL Z X=Y? 196
DSE X E RCL IND Y -
X<0? GTO 08 E XEQ 10
X>Y? GTO 09 APPREC
CLA DSE 00 GTO 09

220*LBL 11
RCL 12 RCL 17 X=Y?
GTO 11 CLA ARCL 20

370*LBL "RE"
CLA ARCL 20 CLX
SEEKPTA FIX 0 1.013
STO 00 "EE" SF 17
OUTA

381*LBL 00
GETREC SF 17 OUTA
GETREC SF 17 OUTA
RCL 00 10 MOD INT
VIEM X ISG 00 GTO 00
CF 01 RCL 06 1 -
X=0? RCL 07 STO 06
XEQ D PROMPT END

MICHAEL BUHLER
943
7500 KARLSRUHE
WALDSTR. 8

Programm "PSL" 1166

In der PRISMA-Ausgabe vom August/Sep-
tember 1983 erscholl der laute Ruf (mit 10
Ausrufezeichen!), daß, bitteschön, von Mit-
gliedern mit vierstelligen Nummern ebenfalls
Programme erwartet würden.

Um meinen Teil beizutragen, durchmusterte
ich also meine Programme. Die meisten sind
berufsbezogen (hauptsächlich Statistik) und
ohne besondere programmiertechnische
Tricks und bieten damit wenig für andere,
hauptsächlich am Programmieren interessier-
te Mitglieder.

Im gleichen Heft (83.7.31) wurde von Ihnen
berichtet, es lägen "viele Anfragen" zur Steue-
rung Ihrer Modelleisenbahn vor. Daraus
schließe ich, daß eine größere Anzahl von
Clubmitgliedern auch am Thema "Eisenbahn"
interessiert ist.

Ich habe nun ein Programm zur Berechnung
der indizierten Leistung von Dampflokomoti-
ven (in Psi) für eine allgemeine Verwendung
erweitert. Alle Eingaben werden abgefragt
(soweit das möglich war), damit auch Inter-
essierte ohne Kenntnisse der Thermodynamik
oder des Maschinenbaus, was wohl für die

meisten "Modelleisenbahner" gilt, dieses Pro-
gramm benutzen können. Alles weitere findet
sich bei der Programmbeschreibung. Ich hof-
fe, daß das Programm genügend Interessen-
ten anspricht.

Kurzbeschreibung zum Programm "PSL" zur Berechnung der indizierten Leistung von Dampflokomotiven

Das Programm errechnet die größte zu errei-
chende und über einen längeren Zeitraum auf-
recht erhaltene "indizierte" Leistung "moder-
ner" normalspuriger Dampflokomotiven. Die
Einschränkung "modern" bedeutet hier etwa
von der Jahrhundertwende bis zum Ende der
Dampflokzeit. Desweiteren gilt es nur für "nor-
malgebaute" Lokomotiven mit Kolbendampf-
maschine und "Normal-" (Stephenson-) Kes-
sel. Weitere Einschränkungen siehe weiter
unten.

Das Programm ist so geschrieben, daß es
auch mit wenig technisch-physikalischem Vor-
wissen benutzt werden kann. Es ist außer-
dem so angelegt worden, daß zur sachgerech-
ten Anwendung die meisten Übersichtstabel-
len ausreichen, die in (deutschen) Büchern für
Eisenbahnfreunde bzw. Modelleisenbahner

abgedruckt sind. Lokomotiven, deren Dimen-
sionen nur im englischen Maßsystem vorlie-
gen, können ebenfalls berechnet werden. Es
ist nur darauf zu achten, daß alle Werte konse-
quent dem gleichen Maßsystem angehören.
Dabei ist zu berücksichtigen, daß im "anglo-
amerikanischen Modus" alle Heizflächen, wie
dort üblich, als "wasserseitig" vermessen an-
genommen werden. Die entsprechenden Um-
rechnungen werden automatisch vorgenom-
men.

Eine exakte Berechnung der Leistung ist mit
einem Einzelprogramm auf dem HP-41 nicht
möglich. Zum einen wegen des Umfangs an
Einzelrechnungen, zum anderen, weil (für den
"Hobbyeisenbahner") in der Regel die meis-
ten der dazu erforderlichen Dimensionen
nicht bekannt bzw. zugänglich sein dürften.
Denn im größten Teil der für den interessierten
Leser zugänglichen Bücher sind vielfach nur
die "wichtigsten" oder "Hauptdimensionen"
einer Lokomotive aufgelistet. Das Programm
stützt sich daher auf "vereinfachte Rechnun-
gen", wie sie früher - in der Zeit vor der Einfüh-
rung elektronischer Hilfsmittel - weit verbreitet
waren und die sich mit den "Hauptdimensio-
nen" begnügen. Eine Abweichung von bis zu

10 % ist deswegen möglich. In der Regel betragen die Abweichungen aber nicht mehr als 5 % - soweit man das überhaupt feststellen kann. Jedenfalls genügt die Genauigkeit, um die richtige "Größenordnung" der Werte oder die "Leistungsklasse" einer Lokomotive zu errechnen.

Um die Steuerung gering zu halten, errechnet das Programm einen Durchschnittswert aus deutschen, französischen und amerikanischen Formeln, die in der Bewertung einzelner Dimensionen und Faktoren des öfteren voneinander abweichen¹⁾.

Eine ausführliche Darlegung der Formeln, trotz der "vereinfachten" Berechnung, würde hier zu weit führen (und nicht alle Leser wären davon beglückt). Wer sich dafür näher interessiert, kann sich mit Fragen an mich wenden.

So viel sei hier dennoch gesagt:

Die deutschen und die amerikanischen Formeln berechnen zuerst aus den Kesseldimensionen die Dampfmenge und dann aus den Maßen der Dampfmaschine (Zylinder) sowie Dampfdruck und -temperatur den Dampfverbrauch.

Aus der einfachen Beziehung $N_i = D/d_i$ ergibt sich dann schließlich die Leistung in PS (die jeder, der möchte, in KW umrechnen kann!). Die französische Rechnung ist einfacher, hier errechnet sich die Leistung aus den wichtigsten Dimensionen direkt.

Zum Abschluß noch kurz eine Erklärung zu den Leistungen, bei denen es sich hier immer um die indizierte, die Zylinder-Leistung handelt:

Das Programm errechnet z.B. für die preußische Lokomotive P 8 eine Leistung von 1450 PSi. Interessierte Leser werden in der Regel niedrigere Angaben in Erinnerung haben. Obermayer²⁾ gibt z.B. 1180 PSi an, was etwa mit den offiziellen Angaben der Reichsbahn übereinstimmt. Die französischen Eisenbahnverwaltungen hingegen rechneten für ihre P 8 mit etwa 1350 - 1380 PS am Radreifen³⁾. Das stimmt mit der Leistung von 1520 PSi überein, die das Programm mit XEQ B errechnet - also mit den französischen Formeln. In Tests hingegen erreichte die Lok jedoch (für kurze Zeit) fast 2000 PSi⁴⁾.

Die Erklärung liegt darin, daß zum einen kurzfristige Höchstleistungen nicht mit der höchsten Dauerleistung und zum anderen die preußische Staatsbahn und die Deutsche Reichsbahn extrem vorsichtige Leistungsberechnungen bzw. -angaben machten. Diese Leistungsangaben könnte man fast als "Fahrplan-PS" oder als "Wartungs- und Unterhaltungs-PS" bezeichnen. Überhaupt fällt auf, daß die "deutschen" viel "vorsichtigere" Ergebnisse errechnen als "ausländische" Formeln.

Das Hauptprogramm errechnet deshalb die Dampferzeugung nach den (österreichischen) Formeln von Sanzin. Als Unterprogramme sind aber noch eine vereinfachte Leistungsbeziehung (XEQ J) unter Einbeziehung der "quasi-amtlichen" Reichsbahn "Kesselobergrenze" von 57 kg/qmh angefügt⁵⁾ sowie die Berechnung der Dampferzeugung nach den (preußischen) Formeln (XEQ I) nach Strahl angefügt. Diese Berechnungen, die in die ausführlicheren anderen Rechengänge des Hauptprogramms nicht einbezogen werden, ergeben für die P 8 etwas über 1100 bzw. 1200 PSi. Diese beiden Werte sind technisch kein Widerspruch, denn im Unterprogramm I wird mit einer sehr hohen Kesselanstrengung $A = 5$ gerechnet (♣ Zeile 942). (Für die Interessierten: Die Amerikaner nehmen noch wesent-

lich höhere Kesselbelastungen als "normal" an!) Bei einer Kesselanstrengung von $A = 4$ ergibt sich dann ebenfalls ein Wert von etwas über 1100 PSi. Wer sich weiter mit diesen Fragen beschäftigen will, mag die "Kesselanstrengung" A z.B. Register 20 zuweisen, um zu sehen, wie sich die Ergebnisse ändern. für weitere Fragen stehe ich gerne zur Verfügung.

Und nun zur praktischen Anwendung:

Benötigt werden ein HP-41 CV oder HP-41 C mit Quadram, das X-Functionsmodul und ein Drucker.

Zur Eingabe:

1. Nach Eingabe des Programms mit Magnetkarten kann es mit "XEQ e" oder mit "XEQ 'PSL'" gestartet werden.
2. Auf die Frage "Name/Gattung" können mit bis zu 24 Alpha-Zeichen der Name und/oder Gattung eingegeben werden, was bei einer Berechnungsreihe die Unterscheidung der Ergebnisse erleichtert. Mit "XEQ D" kann dies umgangen werden.
3. "INCH?" ist die Frage nach dem Maßsystem. "JA" ist Antwort, wenn man englische/amerikanische Maße eingeben möchte. "N" (oder jedes beliebige andere Zeichen), wenn metrische Maße eingegeben werden.
4. "VERBUND?" fragt, ob die Lokomotive nach dem Verbundsystem arbeitet. Antwort "JA" oder "N".
5. "VERBRENNUNGSK?" möchte wissen, ob die Feuerbüchse eine Verbrennungskammer und/oder Wasserkammern (Nicholson's thermic syphons) besitzt (gilt hauptsächlich für "neuere" amerikanische Loks!). Antwort "JA" oder "N".
6. Auf die Frage nach "VORWAERMER" muß mit einer numerischen Eingabe geantwortet werden. Und zwar mit:
 - 1 ♠ kein Vorwärmer
 - 1,08 ♠ Abdampf injektor (England!)
 - 1,1 ♠ Oberflächenvorwärmer
 - 1,12 ♠ Mischvorwärmer
 - 1,14 ♠ Heiln Mischvorwärmer
7. Auf die Frage nach dem "HEIZWERT" muß der Heizwert in kcal/kg oder in BTU/lb (bei englischen Maßen!) des Brennmaterials eingegeben werden - soweit bekannt. Angaben über den Heizwert können "besseren" Lexika oder technischen Nachschlagewerken entnommen werden. In Deutschland wurde in der Regel mit Steinkohle mit 7000 kcal/kg, in Österreich mit 6250 kcal/kg gerechnet. Bei englisch/amerikanischen Maßeinheiten kann man von 13300 - 13500 BTU/lb als Norm ausgehen.
8. "ZYLINDER?" = Anzahl der Zylinder.
9. "HV=?" P gesamte Verdampfungsheizfläche in qm bzw. sq.ft (squarefeet)
10. "HB=?" = Heizfläche der Feuerbüchse (und Verbrennungskammer etc., wenn vorhanden) in qm bzw. sq.ft
11. "HUe=?" = Heizfläche des Überhitzers in qm bzw. sq.ft. Eingabe von 0, wenn kein Überhitzer vorhanden.
12. "R=?" = Rostfläche in qm bzw. sq.ft
13. "RL=?" = Rohrlänge in m. Liegt in der Regel zwischen 4 - 6 m. Vor allem für die amerikanischen Formeln wichtig. In deutschen wird diese Angabe (deshalb?) häufig weggelassen. Kann notfalls mit Hilfe von Zeichnungen und Photographien annähernd geschätzt werden.
14. "d=?" = Durchmesser der Zylinder in cm bzw. inch. Bei Verbundlokomotiven ist der Durchmesser der Niederdruckzylinder einzusetzen.

15. "P=?" = Kesseldruck in kp/cm² (oder in bar) bzw. lb/sq.ft
16. "D=?" = Durchmesser der Treibräder in m bzw. inch.

Ausgaben:

1. Ni = (rechnerische) höchste, indizierte Dauerleistung in PS (die Indices d, f, a, r dienen zur Unterscheidung der Berechnungsart).
2. D = erzeugte Dampfmenge pro Stunde in kg/h.
3. Hbe = Heizflächenbelastung in kg/qmh (bei Unterprogramm "J" einheitlich die RB "Kesselgrenze" von 57 kg/qmh).
4. di = indizierte "günstigste" Dampfverbrauch bei der "günstigsten" Geschwindigkeit. (Wird hier allerdings nicht gesondert berechnet)

Einschränkungen:

Das Programm rechnet nur (annähernd) richtig innerhalb folgender Grenzen:

1. "Normal"-gebaute normalspurige Lokomotiven, also nicht für "Sonderbauarten" irgendwelcher Art. Die Formeln basieren weiterhin auf der Annahme, daß bei der Konstruktion keine wesentlichen Fehler unterlaufen sind. Alle Abweichungen von den "üblichen" Baugrundsätzen schlagen sich schnell in "Abweichungen" nieder. Durch die Berechnung eines Mittelwertes aus verschiedenen Formeln werden Verzerrungen zwar gemildert, aber nicht gänzlich vermieden.
2. Kesseldruck etwa von 12 - 20 kp/cm². Bei Naßdampflokomotiven kann bis etwa 10 kp/cm² gerechnet werden. Bei Lokomotiven mit noch niedrigeren Drücken werden die Abweichungen zu groß, d.h. es ergeben sich zu hohe PS-Werte.
3. Die Rostfläche sollte 1,6 m nicht viel unterschreiten. Durch eingefügte Korrekturfaktoren kann bis etwa 1,4 qm gerechnet werden.
4. Die Rohrlänge sollte 3 m nicht (viel) unterschreiten.
5. Die Berechnung des Unterprogramms I gilt nur für 'normale' ("deutsche") Steinkohlen. Bei anderen Brennstoffen müssen einige (empirische) Koeffizienten geändert werden.

Tastenbelegung der Unterprogramme:

- XEQ A: "deutsche" Berechnung
 XEQ B: "französische" Berechnung
 XEQ C: "amerikanische" Berechnung
 XEQ D: Kurzeingabe der Dimensionen (Nr. 9 - 16), wenn die Bedingungen 1 - 8 bereits festgelegt sind und nicht geändert werden sollen. keine Ausgabe der Gattungsbezeichnungen oder Namen.
 XEQ E: Erneuter Start des Programms ohne Veränderung von Parametern. Ermöglicht auch die Berechnung verschiedener Varianten nach manueller Veränderung einzelner Dimensionen.
 XEQ I: Berechnung der Dampferzeugung nach Strahl und Meineke/Röhrs.
 XEQ J: Verkürzte Berechnung mit der Reichsbahn "Kesselgrenze" von 57 kg/qmh.

Abschluß bzw. Verlassen des Programms:

Nach der Frage "ENDE?" kann das Programm mit R/S verlassen werden. Damit werden die "Zwischenfiles" im X-Function Modul wieder gelöscht! Oder mit XEQ n können die einzelnen Unterprogramme zum Verleich der einzelnen Ergebnisse abgefragt werden.

Speicherbelegung:

- R00 - R09, R20: mehrfach genutzte Rechenregister
 R10: Rohrlänge
 R11: ges. Heizfläche

R12: Überhitzer Heizfl.
 R13: Feuerbüchsen Heizfl.
 R14: Rostfläche
 R15: Zylinderdurchmesser
 R16: Rechenregister
 R17: Kesseldruck
 R18: Raddurchmesser
 R19: Heizwert

Anmerkungen:

- 1) Henschel - Lokomotiv Taschenbuch (1935) u. (1960), Krupp Taschenbuch für den Lokomotivingenieur (1939)
- 2) H.J. Obermayer, Taschenbuch Deutsche Dampflokomotivem (1969)
- 3) J. Gilot, Les Locomotives a vapeur de la S.N.C.F. Region Est (1976)
- 4) Garbe, Die Dampflokomotiven der Gegenwart, 560ff (1923)
- 5) Vgl. Giesl-Gieslingen (1976)

Literatur:

- (nur die Handbücher, keine Fachaufsätze)
1. O. Grove, Die Lokomotive im Allgemeinen und die Entwicklung ihrer Grundverhältnisse; in: E. Heusinger von Waldegg, Handbuch für spezielle Eisenbahn-Technik (1875)
 2. R. Garbe, Die Dampflokomotiven der Gegenwart (1920)
 3. R. Garbe, Die zeitgemäße Heißdampflokomotive (1924)
 4. J. Jahn, Die Dampflokomotive (1923)
 5. Henschel Lokomotiv-Taschenbuch, Ausgaben 1935 und 1960
 6. Krupp-Taschenbuch für den Lokomotivingenieur (1939)
 7. L. Marks, Mechanical Engineer's Handbook (1941)
 8. R. Johnson, The Steam Locomotive (1945)
 9. E. Sauvage et. A. Chapelon, La machine locomotive (1947)
 10. F. Meineke u. F. Röhrs, Die Dampflokomotive (1949)
 11. Eine kurze Einführung gibt: A. Giesl-Gieslingen, Lokomotiv-Athleten (1976)

```
48+LBL E
FS?C 08 XEQ 17 SF 05
52+LBL 16
0 STO 20 XEQ C ST+ 20
XEQ A ST+ 20 XEQ B
ST+ 20 FC? 00 XEQ 02
```

```
63+LBL 01
CF 05 XEQ 20 XEQ 30
RCL 20 3 / ARCL X
"-F PS" 105 XTOA ACA
PRBUF GTO 33
```

```
77+LBL D
SF 11 XEQ 00 GTO 16
```

```
81+LBL 17
FC? 09 XEQ 18
```

```
84+LBL 00
"HV=?" PROMPT STO 11
"HB=?" PROMPT STO 13
"HUe=?" PROMPT X=0?
SF 02 STO 12 "R=?"
PROMPT STO 14 "RL=?"
PROMPT STO 10 "d=?"
PROMPT STO 15 "P=?"
PROMPT STO 17 "D=?"
PROMPT STO 18 RTN
```

```
112+LBL 18
SF 25 "LN" PURFL 10
CRFLAS SF 09
"NAME/GATTUNG?" AON
PROMPT APPREC AOFF
RTN
```

```
125+LBL A
FS? 00 XEQ 31 XEQ 28
XEQ 23 FS? 02 GTO 04
FS? 01 GTO 03 GTO 05
```

```
135+LBL 03
XEQ 21 XEQ 25 617 -
854 X<>Y / XEQ 24 *
STO 04 FC? 03 GTO 00
GTO 06
```

```
149+LBL 00
1,25
```

```
151+LBL 04
ST* 04 GTO 06 XEQ 19
1,5 RCL 14 X>Y?
GTO 06 1,15 ST* 04
GTO 06
```

```
162+LBL 05
XEQ 21 XEQ 22 RCL 00
RCL 00 RCL 07 / -
XEQ 24 * FC? 03
GTO 01 1,12 *
```

```
176+LBL 01
STO 04
```

```
178+LBL 06
RCL 03 FS? 02 XEQ 29
RCL 03 RCL 04 /
FS? 05 RTN STO 09
XEQ 30 ARCL 09 "-F PSd"
ACA PRBUF GTO 32
```

```
194+LBL 23
5525 RCL 11 RCL 14 /
35 + / RCL 11 *
FS? 04 XEQ 00 RCL 01
* RCL 02 * ,93 *
STO 03 RTN
```

```
214+LBL 00
1,18 * RTN
```

```
218+LBL B
FS? 00 XEQ 31 18
STO 07 FS? 02 GTO 00
2 ST+ 07
```

```
227+LBL 00
FC? 01 GTO 01 1
ST+ 07
```

```
232+LBL 01
RCL 18 1,5 X>Y?
GTO 02 1 ST+ 07
```

```
239+LBL 02
FC? 04 GTO 03 2
ST+ 07
```

```
244+LBL 03
RCL 12 RCL 11 RCL 13
- 3 1/X * + RCL 13
+ RCL 17 * RCL 14 *
SQRT RCL 07 * 1,1 *
STO 09 14 RCL 17
X<=Y? XEQ 07 FS? 01
GTO 04 FS? 03 XEQ 06
GTO 00
```

```
274+LBL 04
FC? 03 XEQ 08
```

```
277+LBL 00
FC? 02 GTO 05 XEQ 29
RCL 09 * STO 09
```

```
284+LBL 05
RCL 09 FS? 05 RTN
XEQ 30 ARCL 09 "-F PS"
102 XTOA ACA PRBUF
GTO 33
```

```
296+LBL 06
,9 ST* 09 RTN
```

```
300+LBL 07
RCL 02 ST* 09 RTN
```

```
304+LBL 08
,87 ST* 09 RTN
```

```
308+LBL C
XEQ 26 XEQ 27 FS? 02
GTO 00 XEQ 21
```

```
314+LBL 00
RCL 13 FC? 00 GTO 01
FS? 04 GTO 03 55
GTO 04
```

```
322+LBL 01
FS? 04 GTO 02 617
GTO 04
```

```
327+LBL 02
854 GTO 04
```

```
330+LBL 03
80
```

```
332+LBL 04
* STO 03 RCL 11
RCL 13 - RCL 09 *
ST+ 03 XEQ 09 ST* 03
RCL 02 ST* 03 FS? 01
GTO 00 FS? 03 XEQ 05
```

```
349+LBL 00
XEQ 13 XEQ 11 FS? 01
XEQ 12 FS? 02 XEQ 14
1,5 RCL 14 FS? 00
XEQ 01 X>Y? GTO 02
,94 ST* 03
```

```
364+LBL 02
FS? 03 XEQ 14 FS? 02
XEQ 29 RCL 03 RCL 04
/ FS? 05 GTO 06
STO 09 XEQ 30 FC? 00
GTO 02 ARCL 09 "-F HPa"
GTO 03
```

```
381+LBL 02
RCL 09 1,013 * ARCL X
"-F PSa"
```

```
CAT 1
LBL'PSL
END 2032 BYTES
.END. 05 BYTES
PRP "PSL"
```

```
01+LBL "PSL"
21 PSIZE 0 X<>F
XEQ 18 "JA" ASTO 00
AON "INCH?" PROMPT
ASTO X AOFF RCL 00
X=Y? SF 00 AON
"VERBUND?" PROMPT
ASTO X AOFF RCL 00
X=Y? SF 01 AON
"VERBRENNK?" PROMPT
ASTO X AOFF RCL 00
X=Y? SF 04 1
"VORWAERHER?" PROMPT
STO 02 "HEIZWERT?"
PROMPT FS? 00 SF 06
STO 19 "ZYLINDER?"
PROMPT 2 X<Y? SF 03
SF 08
```

387*LBL 03
ACA PRBUF SF 08
GTO 32

392*LBL 01
10,8 / RTN

396*LBL 05
,97 ST* 03 RTN

400*LBL 06
1,013 * RTN

404*LBL 13
RCL 17 FC? 00 XEQ 07
-,073 * RCL 17 X↑2
1,151 E-4 * + 38,025
+ STO 04 FC? 00
GTO 08 RTN

421*LBL 07
14,223 * RCL 17
STO 08 X<>Y STO 17
RTN

429*LBL 08
RCL 08 STO 17 RTN

433*LBL 09
RCL 19 FC? 06 XEQ 10
13300 / RTN

440*LBL 10
1,9 * RTN

444*LBL 11
FS? 02 RTN RCL 00
RCL 05 / STO 06
-3,595 * RCL 06 X↑2
1,766 * + RCL 06 3
Y↑X -,303 * + 3,132
+ ST* 04 RTN

468*LBL 12
,88 FS? 02 ,77 ST* 04
FC? 03 XEQ 00 RTN

476*LBL 00
1,2 ST* 04 RTN

480*LBL 14
,95 ST* 03 RTN

484*LBL 19
13,4 RCL 15 31,5 / -
RCL 17 LN 1,2 LN -
2,3026 / 1/X *
STO 04 FC? 01 RTN ,83
FS? 03 GTO 00 ,905

506*LBL 00
ST* 04 RTN RTN

510*LBL 20
FS?C 11 RTN CLA SF 17
252 XTOA 27 XTOA 38
XTOA 107 XTOA 49
XTOA 72 XTOA OUTA
CLA ADV ADV CF 29
SF 12 FIX 0 "LN" 0
SEEKPTA GETREC ACA
PRBUF FIX 1 SF 29
CF 12 RTN

544*LBL 21
107 RCL 11 RCL 13 -
RCL 13 / STO 16 10,2
X<=Y? GTO 01 CLX 12,2
* + 360 RCL 12 X<=0?
RTN RCL 11 / * +
STO 00 RTN

569*LBL 01
1 E↑X RCL 12 RCL 11
/ ,45 * Y↑X 309,5 *
10 - 13,5 RCL 16 -
12,12 * - STO 00 RTN

590*LBL 22
1 E↑X RCL 17 -,023 *
Y↑X 17,15 * STO 08
RCL 17 2,63 * 27,26
+ STO 07 RTN

607*LBL 24
1,24 RCL 15 250 / -
RTN

614*LBL 25
RCL 00 ,52 * 570 +
RTN

621*LBL 26
RCL 10 FC? 00 XEQ 01
-,678 * RCL 10 X↑2
,011 * + RCL 10 3
Y↑X -5,513 E-5 * +
10,716 + FC? 00
GTO 02 STO 09 RTN

644*LBL 01
STO 08 3,28 * STO 10
RTN

650*LBL 02
9,6 * STO 09 RCL 08
STO 10 RTN

657*LBL 27
RCL 17 FC? 00 GTO 01
14,223 /

663*LBL 01
1 + ,2 Y↑X 139,08 *
40,2 - STO 05 RTN

674*LBL 28
6250 RCL 19 FC? 00
GTO 01 ,556 *

681*LBL 01
%CH 1 % ,66 * 1 +
STO 01 RTN

691*LBL 29
11,8 RCL 10 - RCL 10
* RCL 10 SQRT 2 + /
2 + ,1 * 1 X<>Y
X<=Y? ST* 03 RTN

711*LBL 30
"N" 105 XTOA 32 XTOA
40 XTOA 109 XTOA 97
XTOA 120 XTOA 59
XTOA 112 XTOA 41
XTOA 32 XTOA 61 XTOA
ACA FMT CLA RTN

739*LBL 31
SF 25 "LD" PURFL 10
CRFLD 10,019 SAVERX
SF 10 RCL 10 ,305 *
STO 10 RCL 12 9,8
SF 25 / STO 12 RCL 11
11,73 / STO 11 RCL 13
LASTX / STO 13 RCL 14
LASTX ,9 * / STO 14
RCL 15 2,54 * STO 15
RCL 17 14,223 /
STO 17 RCL 18 ,0254 *
STO 18 RCL 19 ,556 *
STO 19 CF 00 RTN

789*LBL 32
"D=" ACA FMT CLA
RCL 03 STO 08 FS? 00
GTO 00 FS? 08 XEQ 07
STO 08 ARCL X XEQ 05
GTO 01

804*LBL 00
ARCL X XEQ 06

807*LBL 01
104 XTOA ACA PRBUF
CLA RCL 08 "Hbe=" ACA
FMT CLA RCL 11 FS? 07
XEQ 08 FS? 00 GTO 00
/ ARCL X XEQ 05 113
XTOA 109 XTOA GTO 02

831*LBL 00
/ ARCL X XEQ 06 115
XTOA 113 XTOA 102
XTOA

841*LBL 02
ACA PRBUF CLA "d"
105 XTOA "t=" ACA
FMT CLA RCL 04 FS? 00
GTO 00 FS? 08 XEQ 07
ARCL X XEQ 05 "tPS"
GTO 03

861*LBL 00
ARCL X XEQ 06 "tHP"

865*LBL 03
ACA PRBUF FS? 00
GTO 04 FS?C 07 SF 00
CF 08 ADV GTO 33

875*LBL 04
RCL 09 1,013 * STO 09
XEQ 30 ARCL 09 "tPSa"
ACA PRBUF CF 00 SF 07
GTO 32

888*LBL 05
32 XTOA 107 XTOA 103
XTOA 47 XTOA RTN

898*LBL 06
108 XTOA 98 XTOA 47
XTOA RTN

906*LBL 07
2,2 / RTN

910*LBL 08
,0929 * RTN

914*LBL 33
FS?C 10 XEQ 34 BEEP
ADV "ENDE?" PROMPT
"LN" SF 25 PURFL
CF 09 "LD" SF 25
PURFL CF 10 STOP

930*LBL 34
"LD" 0 SEEKPTA 10,019
GETRX SF 00 RTN

938*LBL I
FS? 00 XEQ 31 RCL 14
5 * 1 E6 * STO 06
RCL 19 / RCL 14 /

.05 * CHS 85 + 1 %
RCL 06 * STO 06
XEQ 25 20 - RCL 06
X<>Y / STO 03 SF 14
GTO 06

970*LBL J
FS? 00 XEQ 31 RCL 11
57 * STO 03

977*LBL 06
FS? 02 GTO 00 RCL 17
FS? 01 GTO 01 -.11 *
8.44 + GTO 02

988*LBL 00
XEQ 19 GTO 03

991*LBL 01
-.2 * 9.99 + FC? 03
XEQ 05

998*LBL 02
STO 04 XEQ 24 ST* 04
FS? 01 GTO 03 FS? 03
XEQ 04

1006*LBL 03
FS? 14 GTO 00 1.04
ST* 04

1011*LBL 00
RCL 03 RCL 04 /
STO 09 XEQ 30 ARCL 09
"I PS" FS?C 14 GTO 07
114 GTO 08

1023*LBL 07
115

1025*LBL 08
XTOA ACA PRBUF GTO 32

1030*LBL 04
1.08 ST* 04 RTH

1034*LBL 05
1.18 * RTN END

XEQ "PSL"

NAME/GATTUNG?

GREAT WESTERN RY. 4300

RUN

INCH?

JA RUN

VERBUND?

N RUN

VERBRENNK?

N RUN

84.1.16

VORWAERMER?

1.08 RUN

HEIZWERT?

13.800.0 RUN

ZYLINDER?

2.0 RUN

HV=?

1.478.36 RUN

HB=?

128.72 RUN

HUe=?

191.79 RUN

R=?

20.56 RUN

RL=?

11.5 RUN

d=?

18.5 RUN

P=?

200.0 RUN

D=?

68.0 RUN

GREAT
WESTERN RY.
4300

Ni (max;p) = 1.097,3 PSI

ENDE?

XEQ B

Ni (max;p) = 1.003,4 PSf

ENDE?

XEQ C

Ni (max;p) = 1.309,2 HPa

D= 26.520,81b/h

Hbe= 17,91b/saf

di= 20,31b/HP

Ni (max;p) = 1.326,2 PSa

D= 12.054,9 kg/h

Hbe= 87,8 kg/qm

di= 9,2 kg/PS

ENDE?

XEQ A

Ni (max;p) = 962,3 PSd

D= 8.067,8 kg/h

Hbe= 64,0 kg/qm

di= 8,4 kg/PS

ENDE?

XEQ I

Ni (max;p) = 1.007,8 PSs

D= 7.308,6 kg/h

Hbe= 58,0 kg/qm

di= 7,3 kg/PS

ENDE?

XEQ J
Ni (max;p) = 952,5 PSr
D= 7.183,8 kg/h
Hbe= 57,0 kg/qm
di= 7,5 kg/PS

ENDE?

XEQ "PSL"

NAME/GATTUNG?

BAUREIHE 57.10 PREUS.G10

RUN

INCH?

N RUN

VERBUND?

N RUN

VERBRENNK?

N RUN

VORWAERMER?

1.1 RUN

HEIZWERT?

7.000.0 RUN

ZYLINDER?

2.0 RUN

HV=?

146.0 RUN

HB=?

14.47 RUN

HUe=?

58.9 RUN

R=?

2.58 RUN

RL=?

4.7 RUN

d=?

63.0 RUN

P=?

12.0 RUN

D=?

1.4 RUN

Peter Bayerlein
Klarenthaler Str. 4
6200 Wiesbaden

BAUREIHE

57.10

PREUS.G10

Ni (max;p) = 1.445,2 PSI

ENDE?

XEQ A

Ni (max;p) = 1.438,8 PSd

D= 9.723,4 kg/h

Hbe= 66,6 kg/qm

di= 6,8 kg/PS

ENDE?

XEQ B

Ni (max;p) = 1.457,8 PSf

ENDE?

XEQ C

Ni (max;p) = 1.438,9 PSa

D= 11.204,3 kg/h

Hbe= 76,7 kg/qm

di= 7,9 kg/PS

ENDE?

XEQ I

Ni (max;p) = 1.223,3 PSs

D= 8.605,3 kg/h

Hbe= 58,9 kg/qm

di= 7,0 kg/PS

ENDE?

XEQ J

Ni (max;p) = 1.137,5 PSr

D= 8.322,0 kg/h

Hbe= 57,0 kg/qm

di= 7,3 kg/PS

ENDE?

RUN

Prisma zu spät?

Dieses Prisma ist etwas später erschienen als es sonst üblich ist. Die 36 Seiten kosten eben Zeit. Ich hoffe, diese Verspätung wieder aufzuholen.

Ich habe viele zustimmende Briefe zu der neuen Größe der Listings erhalten. Die Beschwerden über unlesbare Listings werden der Vergangenheit angehören, wenn alle Autoren schwarzes Thermopapier verwenden.

Die Autoren, die noch Magnetkarten von mir bekommen, möchten mir bitte eine Postkarte senden, auf der die Anzahl der Karten steht. Ich schicke dann die Karten. Bitte versteht diese Lösung. Mir wächst die Arbeit langsam über den Kopf!

Euer Rolf (500)

.END.

Neues LOAD BYTES Programm 1167

Seit der Veröffentlichung des "LOAD BYTES" im Dezember 1982 habe ich das Programm mehrfach überarbeitet. Es gab manchmal Probleme bei der Decodierung des b-Registers, die jetzt behoben sind. Die alte Programmversion sollte deshalb nicht mehr benutzt werden. Ein paar Bytes sind noch eingespart worden. Das Programm benutzt das Extended Functions Modul, aber keine Datenregister.

Bedienungshinweise:

1. Zu der Programmzeile gehen, wo die synthetischen Bytes ein- gegeben werden sollen.
2. PRGM on, eingeben: XEQ "LB", STOP und Pufferbytes. Als Pufferbytes können beliebige Bytes eingegeben werden. Es müssen mindestens 19 Pufferbytes eingegeben werden, um 1 ... 6 Bytes synthetisch laden zu können. Für jedes weitere angefangene Register (= 7 Bytes) müssen 7 zusätzliche Pufferbytes eingegeben werden.
3. Zur Programmzeile XEQ "LB" gehen, PRGM off, dann R/S.
4. Der Ton kennzeichnet den Programmstopp, um Dezimalwerte der Bytes einzugeben. In der Anzeige ist jeweils der zuletzt eingegebene Wert zu sehen.
5. Dezimalwert ohne ENTER eingeben, dann R/S. Auf diese Weise fortfahren, bis alle Bytes eingegeben sind.
6. Um Eingabe zu beenden: nur R/S ohne Zahleneingabe.
7. Nach dem BEEP: PRGM on, überflüssige Zeilen löschen (rest- liche Pufferbytes und XEQ "LB" und STOP)

```

27♦LBL 00
28 FS? 22
29 TONE 9
30 FS?C 22
31 STOP
32 FC? 22
33 ,
34 XTOA
35 X<>Y
36 RDN
37 DSE Y
38 GTO 00
39 7
40 X<> c
41 X<> [
42 STO IND L
43 X<> [
44 X<> c
45 X<>Y
46 CLA
47 DSE L
48 FS? 22
49 GTO 00
50 BEEP
51 END
    
```

```

LBL "LB"
END
83 BYTES
    
```

Anz. Bytes: 149
MK: Spur 2
Synt. Befehle: X/N
SIZE: 000
Geräte-Konf.: HP-41

```

01♦LBL "STREIK"
02 SF 11
03 OFF
04 "ICH BIN"
05 AVIEW
06 PSE
07 "MUEDE,"
08 AVIEW
09 PSE
10 "LASS MICH"
11 AVIEW
12 PSE
13 "IN RUHE"
14 AVIEW
15 PSE
16 SF 11
17 OFF
18 "HOERST DU?"
19 AVIEW
20 PSE
21 "ICH WILL"
22 AVIEW
23 PSE
24 "MEINE RUHE"
25 AVIEW
26 PSE
27 SF 11
28 OFF
29 "RUTSCH MIR"
30 AVIEW
31 PSE
32 "DEN BUECKEL"
33 AVIEW
34 PSE
35 "RUNTER"
36 AVIEW
37 PSE
38 "ADIEU"
39 AVIEW
40 PSE
41♦LBL 01
42 SF 11
43 OFF
44 GTO 01
45 END
    
```

```

LBL "STREIK"
END 149 BYTES
.END. 87 BYTES
    
```

Paul Strauch
Steinackerstr. 2
CH-8953 Dietikon
☎ 01/7407675

8. Wenn man eine falsche Zahl eingegeben hat und das bemerkt, bevor R/S gedrückt wurde: (CLX), dann richtige Eingabe. Der Stack darf nicht durch ENTER, RDN oder mehrere Eingaben verändert werden!

Warnungen

1. Das Programm warnt nicht, wenn der Puffer überschritten wird.
 2. Der STACK, LASTX und ALPHA dürfen nicht verändert werden, während das Programm benutzt wird.
 3. Das Programm sollte nicht angehalten werden. (Zwischen den Zeilen 40 und 44 erhält man MEMORY LOST)
 4. Wird das Programm direkt von der Tastatur aus gestartet, so stoppt es in Zeile 08 mit NO-NEXISTENT.
- Happy Synthetic Programming!

Gerhard Kruse (1004)
Drosselpfad 1
5100 Aachen-Brand

Der streikende Taschenrechner 1168

Dieses Programm löst Verblüffung und Heiterkeit aus. Dem nicht versierten Anwender des HP-41 wird das Wiedereinschalten wahrscheinlich Mühe bereiten.

Vorbereitung: XEQ "STREIK", danach den Rechner weitergeben.

```

01♦LBL "LB"
02 CLA
03 RCL b
04 X<> [
05 ALENG
06 7
07 ATOX
08 GTO IND Z
    
```

```

09♦LBL 04
10 2
11 MOD
12 256
13 *
14 ALENG
15 2
16 -
17 ATOX
18 *
19 X<0?
20 CLX
21 +
    
```

```

22♦LBL 03
23 2
24 -
25 FRC
26 SF 22
    
```

Neue QTH-Kenner 1169

Ich habe das Programm "Neue QTH-Kenner" genannt, um eine Verwechslung mit dem Programm (PRISMA 1/82) zu vermeiden, da beide Programme auf unterschiedlichen QTH-Kenner-Definitionen beruhen. Die neuen QTH-Kenner ermöglichen eine eindeutige Kennung eines jeglichen Ortes auf der Erde. Dieses Programm berechnet QTH-kenner aus den geographischen Daten (geogr. Länge/Breite) bzw. ordnet einem QTH-Kenner die Mitte eines 5 x 2.5 Bogenminuten großen Feldes zu. Auch die Entfernung bzw. Richtungswinkel zwischen zwei Orten kann an Hand des QTH-Kenners oder der geog. Daten berechnet werden.

Der QTH-kenner besteht aus einer sechsstelligen Zeichenkombination.

A A X X A A

A: Buchstaben

X: Zahl (1...9)

Die geog. Daten, westl. Länge, südl. Breite sind negativ einzugeben.

Format: g.mmss

Die Entfernung, Richtungswinkel werden durch Großkreisberechnung ermittelt.

DX: Entfernung; Richtungswinkel (Nord: 0, Süd: 180)

Das Programm ist teilweise einem Basic Programm nachempfunden, welches in der Zeitschrift mc 9/83 s. 61 zu finden ist. Zur Um-

wandlung von Charakter in Decimal vis versa werden zwei synthetische Routinen CD,DC verwendet, welche aus dem Buch Calculator Tips & Routines, John. Dearing, Corvallis Software Inc stammen.

Status: SIZE 011; Prog. Speicher 528 Bytes Rechnerkonfiguration: (Drucker), (Kartenleser), 1 Memorymodul

Speicher: Reg. 00 - 10 werden benützt.

Flags: 01 benutzt

02 bei DX,DX? wenn geog. Daten, nicht QTH-Kenner verwendet werden.

21 benutzt

Beispiel:

1) Wie lautet der QTH-Kenner für Hamburg (Breite: 53°33' nord; Länge: 9°58' ost)

1a) Wie lautet der QTH-Kenner für New York (40°44' nord; 73°59'west)?

2) Ort eines QTH-kenners: JO 43 XN ?

3) Entfernung zwischen zwei Orten mit QTH-Kennern ?

eigener QTH: JO 43 XN; fremder QTH: FN 30 AR

4) Entfernung zwischen (schon eingegebenen) eigenen QTH (s.3) und neuem fremden QTH JG 88 WA ?

5) Entfernung zwischen zwei Orten mit geographischen Daten von Hamburg und New York

6) Entfernung zwischen (alten) eigenem Ort (s.5) und fremden Ort ?

Jürgen Schmidt
Gemseneck 18
2000 Hamburg 54

01 *LBL "QTH	49 ENTER↑	98 RDN	144 ST- 02
02 "L<?="	50 ENTER↑	99 65	145 ST- 01
03 PROMPT	51 RCL 02	100 +	146 ST- 01
04 STO 00	52 X<0?	101 STO 02	147 12
05 FPC	53 +	102 CLA	148 ST/ 05
06 100	54 FC? 01	103 1.006	149 ST/ 06
07 *	55 +	104 STO 00	150 24
08 INT	56 5	105 *LBL 02	151 1/X
09 STO 02	57 /	106 RCL IND	152 RCL 05
10 RCL 00	58 INT	00	153 +
11 HR	59 65	107 XEQ "DC"	154 RCL 03
12 180	60 +	108 ISG 00	155 2
13 +	61 STO 05	109 GTO 02	156 *
14 STO 00	62 "B<?="	110 AVIEW	157 +
15 20	63 PROMPT	111 FIX 4	158 RCL 01
16 /	64 90	112 CLX	159 +
17 INT	65 +	113 CF 01	160 HMS
18 STO 03	66 HR	114 RTN	161 "L<="
19 65	67 STO 00	115 *LBL "ORT	162 ARCL X
20 +	68 HMS	"	163 AVIEW
21 STO 01	69 FRC	116 "QTH?"	164 48
22 RCL 00	70 100	117 AON	165 1/X
23 20	71 *	118 PROMPT	166 RCL 06
24 RCL 03	72 HR	119 AOFF	167 +
25 *	73 STO 02	120 *LBL "T"	168 RCL 04
26 -	74 60	121 6	169 +
27 2	75 X<>Y	122 STO 00	170 RCL 02
28 /	76 X<0?	123 *LBL 01	171 +
29 INT	77 +	124 SF 21	172 "B<=" "
30 FIX 0	78 2.5	125 XEQ "CD"	173 HMS
31 43	79 /	126 STO IND	174 ARCL X
32 +	80 INT	00	175 AVIEW
33 STO 03	81 65	127 DSE 00	176 RTN
34 RCL 02	82 +	128 GTO 01	177 *LBL "GDx
35 CF 01	83 STO 06	129 65	"
36 RCL 00	84 RCL 00	130 ST- 01	178 SF 02
37 ENTER↑	85 ENTER↑	131 ST- 02	179 *LBL "Dx"
38 ENTER↑	86 ENTER↑	132 ST- 05	180 "EIGENER
39 2	87 10	133 ST- 06	QTH?"
40 /	88 /	134 48	181 AON
41 INT	89 INT	135 ST- 03	182 FC? 02
42 2	90 STO Z	136 ST- 04	183 PROMPT
43 *	91 10	137 10	184 FC? 02
44 X<>Y	92 *	138 ST* 01	185 XEQ "T"
45 INT	93 -	139 ST* 02	186 FS? 02
46 X=Y?	94 INT	140 2	187 XEQ 09
47 SF 01	95 48	141 ST* 01	188 HR
48 60	96 +	142 ST/ 06	189 STO 07
	97 STO 04	143 90	190 X<>Y

Nr.	Operation	Daten	Taste(n)	Resultat
1	Programm einlesen			
2	Umwandlung geog. Daten in QTH Kenner		XEQ QTH	L<?=?
3	Geog. Länge d. Standorts eingeben (L;°,MMSS)	L	R/S	B<?=?
4	Geog. Breite d. Standorts eingeben (B;°,MMSS)	B	R/S	QTH-Kenner
5	Umwandlung QTH-Kenner in geog. Daten		XEQ ORT	(AON)
6	Eingabe QTH-Kenner (im Alpha-Modus)	QTHKenner	R/S	L<=?
			R/S	°,MMSS
			R/S	B<=?
			R/S	°,MMSS
7	Berechnung Entfernung, Winkel		XEQ GDx	L<?=?
8	Eigenen Standort L (°,MMSS)	L	R/S	B<?=?
	Eigenen Standort B (°,MMSS)	B	R/S	L<?=?
	Fremder Standort L (°,MMSS)	L	R/S	B<?=?
	Fremder Standort B (°,MMSS)	B	R/S	DX=xx KM
	DX: Entfernung; Richtungswinkel		R/S	χ=?
			R/S	°,MMSS
10	neuer fremder Standort (wenn geog. Daten : SF 02) weiter wie 9		XEQ DX?	L<?=?
		
11	Berechnung Entfernung, Richtungswinkel aber mit QTH-Kenner	QTH-Kenn.	XEQ DX	EIG. QTH?
			R/S	L<=?
			R/S	°,MMSS
			R/S	B<=?
			R/S	°,MMSS
			R/S	FREMDER QTH?
		QTH-Kenn.	R/S	L<=?
			R/S	°,MMSS
			R/S	B<=?
			R/S	°,MMSS
			R/S	DX=xx KM
			R/S	χ=?
			R/S	°,MMSS

Nr.	Daten	Tasten	Resultat	Kommentare
1		XEQ QTH	L<?=?	
	9.58	R/S	B<?=?	
	53.33	R/S	JO43XN	QTH-Kenner
1a		XEQ QTH	L<?=?	
	-73.59	R/S	B<?=?	
	40.44	R/S	FN30AR	
2		XEQ ORT	QTH?	
	JO43XN	R/S	L<>=9.5730	geo.Länge(g,mmss)
		R/S	B<>=53.3345	geo.Breite(g,mmss)
3		XEQ DX	EIGEN. QTH?	
	JO43XN	R/S	L<>=9.5730	
		R/S	B<>=53.3345	
		R/S	FREMD. QTH?	
	FN30AR	R/S	L<>=-73.5730	
		R/S	B<>=40.4345	
		R/S	DX=6120 KM	Entfernung zwischen QTH-Kennern
		R/S	χ=293.1152	Richtungswinkel zw. QTH-Kennern
4		XEQ DX?	FREMD. QTH?	
	JG88WA	R/S	L<>=17.5230	
		R/S	B<>=-21.5845	
		R/S	DX=8430 KM	
5		XEQ GDx	L<?=?	Eigen. St.ort Länge
	9.58	R/S	B<?=?	Eigen. St.ort Breite
	53.33	R/S	L<?=?	Fremder St. Länge
	-73.53	R/S	B<?=?	Fremder St. Breite
	40.44	R/S	DX=6122. KM	Entfernung in km zw. Orten.
		R/S	χ=293.1351	Richtungswinkel zw. Orten.
6		SF 02		
		XEQ DX?	L<?=?	Fremder St. Länge
	-73.5730	R/S	B<?=?	Fremder St. Breite
	40.4345	R/S	DX=6121 KM	Entfernung
		R/S	χ=293.1240	Richtungswinkel

```

191 HR
192 STO 08
193 LBL "DX?"
"
194 AON
195 "FREMDER
QTH?"
196 FC? 02
197 PROMPT
198 FC? 02
199 XEQ "T"
200 FS? 02
201 XEQ 09
202 HR
203 STO 09
204 X<>Y
205 HR
206 STO 10
207 RCL 08
208 -
209 CHS
210 COS
211 X<>Y
212 COS
213 *
214 RCL 07
215 COS
216 *
217 RCL 09
218 SIN
219 RCL 07'
220 SIN
221 *
222 +
223 ACOS
224 STO 00
225 D-R
226 40009
227 *
228 2
229 /
230 PI
231 /
232 .5
233 +
234 INT
235 FIX 0
236 "DX="
237 ARCL X
238 " KM"
239 AOFF
240 AVIEW
241 FIX 4
242 XEQ "AN"
243 RTN
244 LBL "AN"
245 RCL 09
246 SIN
247 RCL 07
248 SIN
249 RCL 00
250 COS
251 *
252 -
253 RCL 07
254 COS
255 /
256 RCL 00
257 SIN
258 /
259 ACOS
260 RCL 08
261 RCL 10
262 -
263 SIN
264 X<0?
265 GTO 10
266 RDN
267 360
268 X<>Y
269 -
270 HMS
271 XEQ 08
272 RTN
273 LBL 10
274 RDN
275 HMS
276 XEQ 08
277 RTN
278 LBL 08
279 "L="
280 ARCL X
281 AVIEW
282 RTN
283 LBL 09
284 AOFF
285 "L<?="
286 PROMPT
287 "B<?="
288 PROMPT
289 RTN
290 LBL "C)"
291 "T-+****"
↑ (247,127,0,
7,42,42,42,42)
"
292 RCL [
293 FS? 10
294 "T*"
295 STO [
296 CLX
297 X<> ]
298 SIGN
299 CLX
300 X<> \
301 "T****"
302 X<> [
303 X<> L
304 X<> \
305 INT
306 ST+ ]
307 RDN
308 6
309 ST+ ]
310 RDN
311 E1
312 ST+ L
313 X<> L
314 ST+ ]
315 CLX
316 X<> ]
317 RTN
318 LBL "DC"
319 INT
320 256
321 MOD
322 LASTX
323 +
324 OCT
325 X<> d
326 FS?C 11
327 SF 12
328 FS?C 10
329 SF 11
330 FS?C 09
331 SF 10
332 FS? 07
333 SF 09
334 FS? 06
335 SF 08
336 X<> d
337 X<> [
338 RCL \
339 "T*"
340 X<> ]
341 X<>Y
342 STO \
343 X<> ↑
344 "T*"
345 STO ↑
346 RDN
347 X<> ]
348 X<> \
349 STO [
350 RDN
351 RTN
352 .END.

```

zeilen auf die Breite der Y-Achse, unabhängig von XINC.
Der Winkel, der durch ATAN bestimmt wird, errechnet sich aus

$$\varphi = \arctan \frac{\Delta y - 7}{Y_{MAX} - Y_{MIN}}$$

Programmkorrekturen:
Lösche die Zeilen 10 und 09.
Ändere Zeile 04 von 80 auf 7.
Neue Programmlänge: 196 Bytes.
Neue SIZE ... 014

Ansonsten ein Kompliment an Carlo für die Idee mit den Sonderzeichen.

Mit freundlichen Grüßen

Walter Lutz (1659)
Habsburgerallee 16
6000 Frankfurt 60

MPlot

Ich habe das in Prisma Nr. 5 abgedruckte Programm »MPlot« (1062) und die in Prisma Nr. 6 veröffentlichte Korrektur (1089) ausprobiert und mußte dabei feststellen, daß die Korrektur die fehlerhafte Version sein muß! Die abgedruckten Beispiele und weitere eigene Testbeispiele ergaben nur mit der in Nr. 5 abgedruckten Fassung sinnvolle Ergebnisse. Ich nehme an, daß eine Vertauschung vorliegt, die sich aus dem Verwechseln der Beiträge von Carlo ergeben haben dürfte.

Mit freundlichen Grüßen

Daniel Sterner (428)
Am Kellersbusch 8
6072 Dreieich 4

HP-41 programmiert sich selber

Wenn man mit den Modulen GAMES und PPC-ROM nicht aufpaßt!
Wer von den beiden Herstellern, HP oder PPC das Ei gelegt hat, weiß ich nicht, jedenfalls kann sich das Ei bei Gelegenheit sehr schnell in eine Handgranate verwandeln.
Beide Module weisen nämlich die XROM-Nummern 10.XX auf!
Versuch: Steck mal das GAMES in Port 1 und das PPC in Port 2. Pgm on: XEQ 'SIZE? Anzeige: XROM 'SIZE?.
Entfernt jetzt das GAMES und geht wieder in den PGM-Modus. Anzeige: XROM 'ML !!!
Erklärung: Beide Routinen besitzen die XROM-Nummer 10.12!

Wenn Ihr den Speicher gerade leer habt, könnt Ihr ja einmal R/S drücken. Der Rechner wird sich selber programmieren. (Es ist ihm dabei aber höchst gleichgültig, ob der Speicherplatz belegt ist oder nicht.)
Das Ganze läßt sich natürlich umgehen, indem man GAMES immer einen Steckplatz "tiefer" einsteckt als das PPC-ROM.

Trotzdem - be careful!!
Happy programming

Horst (782)

HP-IL

Liebe Mitglieder,
es ist wieder an der Zeit, einen IL-Artikel zu schreiben (bereits erschienene Artikel findet man in PRISMA 82.12, 83.2 und 83.8). Jetzt geht es immer mehr zur praktischen Seite des Ganzen, und dazu benötigen wir noch einige Hilfsmittel. Das erste ist die (möglichst) vollständige Liste aller Funktionen, die die Aussendung von IL-Meldungen zur Folge haben, und die dazugehörigen IL-Sequenzen. Diese Liste ist hier aufgeführt. Die Funktionsnamen sind fettgedruckt, in Klammern folgt, ob sie im IL- oder im X I/O-Modul zu finden sind. Für die IL-Meldungen werden die bereits eingeführten Kürzel verwendet; sie sind oft von Zahlen gefolgt, die dann den dezimalen Wert des betroffenen Bitfeldes des Datenbytes der Meldung angeben. Ist dieser Wert nicht bekannt, so verwende ich folgende Abkürzungen:

sel : Adresse der primären Einheit (Durch SELECT festgelegt.)

CR : Carriage Return, Byte mit dem Wert 13
LF : Line Feed, Byte mit dem Wert 10
x : Wert im X-Register
a1 : erster bis bis
an : n-ter Charakter aus dem ALPHA-Register

Für alle Sequenzen, die mit IDY00 beginnen, gilt: Diese Meldung dient dazu, Geräte in der Schleife aufzuwecken. Es kann sein, daß diese Meldung mehrmals gesendet wird und daß ihr Wert verschieden von 0 ist: IDY??.
Die Sequenz selbst wird in zwei Spalten gedruckt, von denen die linke die IL-Meldungen, die der Controller, sendet angibt, und die rechte die Meldungen, die die primäre Einheit sendet.

AID (X-I/O)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
TADsel	TADsel	
RFC	RFC	
SAI	DABnn	nn=AID der prim. Einheit
DABnn	ETO	
UNT	UNT	
RFC	RFC	

CLRDEV (X-I/O)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
LADsel	LADsel	
RFC	RFC	
SDC	SDC	Selected Device Clear
RFC	RFC	
UNL	UNL	
RFC	RFC	

CLRLOOP (X-I/O)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
DCL	DCL	Device Clear
RFC	RFC	

MPlot

das Programm »MPlot« 1062 (Prisma 83.5.08.) von Carlo (1294) enthält leider einen gravierenden Fehler in der Berechnung des Sonderzeichens. In der Originalfassung müssen ca. 80 X-Einheiten auf die Länge der Y-Achse kommen, wenn die Winkel der Druckzeichen stimmen sollen.
In meiner korrigierten Fassung fallen 7 Druck-

DDL (X-I/O)

DDLx DDLx
RFC RFC

DDT (X-I/O)

DDTx DDTx
RFC RFC

DEVL (X-I/O)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
LADsel LADsel
RFC RFC
DDLx DDLx
RFC RFC
UNL UNL
RFC RFC

DEVT (X-I/O)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
TADsel TADsel
RFC RFC
DDTx DDTx
RFC RFC
UNT UNT
RFC RFC

ID (X-I/O)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
TADsel TADsel
RFC RFC
SDI DABa1
DABa1 ***
*** verschiedene Datenbytes
*** DABan
DABan ETO UNT
UNT
RFC RFC

Das Gerät sendet seine Device ID (eine Kette von Alphazeichen, die die Gerätebezeichnung darstellt). Der Rechner speichert bis zu 24 Zeichen im ALPHA Register ab und wartet, bis er ein ETO erhält.

INA (IL)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
RFC RFC
TADsel TADsel
RFC RFC
SDA DABa1
DABa1 ***
*** beliebige Datenbytes
*** DABan
NRD NRD
DABan ETO
UNT UNT
RFC RFC

Flag 17 set : Der Rechner übergeht alle CR und sendet NRD nach Empfang eines LF oder wenn 24 Zeichen eingegangen sind. Flag 17 clear : Der Rechner wartet in jedem Fall auf 24 Zeichen, wobei CR und LF mitgezählt und ins ALPHA-Register geschrieben werden.

IND (IL)

Die Sequenz ist dieselbe wie bei INA, nur werden hier die empfangenen Bytes nicht ins ALPHA Register geschrieben, sondern als numerische Eingabe ins X-Register interpretiert.

INSTAT (IL)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
TADsel TADsel
RFC RFC
SST DABa1
DABa1 ***

*** ETO
UNT UNT
RFC RFC

Das angesprochene Gerät darf mehr als ein Statusbyte senden. Es wird aber nur das erste zurückbehalten. Sein dezimaler Wert wird ins X-Register geschrieben, und die Flags 0 bis 7 entsprechend gesetzt oder gelöscht.

IN... (X-I/O)

IDY00 IDY00 *
AAU AAU *
RFC RFC *
AAD1 AAD(sel+1) *
TADsel TADsel *
RFC RFC *
SDA DABa1
DABa1 *** beliebige Datenbytes
*** DABan
NRD NRD
DABan ETO
UNT UNT *
RFC RFC *

Die Bedingung unter der der Rechner NRD sendet, hängt von der Funktion ab: Vor dem im ALPHA-Register abgelegten String wird ein "D" geschrieben. Deshalb werden nie mehr als 23 Bytes akzeptiert.

INAN :

Es werden N Charakter eingelesen, wobei N im X-Register angegeben wird (N<24).

INAE : DABan = ENDan: wenn eine END-Meldung empfangen wird.

INAC : DABan = DABx: wenn der Bytewert gleich dem X Wert ist.

INACL : Alle CR werden übergangen und LF bricht den Empfang ab.

INXB : Es wird nur ein Byte akzeptiert, und dessen Wert ins X-Register geschrieben.

Im ADROFF-Modus werden die Meldungen, die von einem "*" gefolgt werden, nicht gesendet.

LAD (X-I/O)

LADx LADx
RFC RFC

LISTEN (IL)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
LADx LADx
RFC RFC

Da keine dieser Meldungen den Listener-Status von einem Gerät entfernen können, kann man LISTEN mehrmals hintereinander ausführen, um mehrere Geräte zu Listenern zu deklarieren. Eine darauffolgende Meldung der ACG-Gruppe (z.B. TRIGGER oder CLRDEV) spricht alle diese Listener gleichzeitig an. Siehe auch OUTA, OUT...

LOCAL (IL)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
LADsel LADsel
RFC RFC
GTL GTL Go To Local
RFC RFC
UNL UNL
RFC RFC

LOCK (X-I/O)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
LLO LLO Local LockOut
RFC RFC

NLOOP (X-I/O)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)

Der Rechner kann aus der zurückerhaltenen AAD Meldung erkennen, wieviele Geräte sich in der Schleife befinden.

NOTREM (X-I/O)

IDY00 IDY00
AAU AAU
RFC RFC
AAD1 AAD(sel+1)
NRE NRE Not Remote Enable
RFC RFC

OUTA (IL)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
LADsel	LADsel	
RFC	RFC	
DABa1	DABa1	
***	***	beliebige Datenbytes
DABan	DABan	
UNL	UNL	
RFC	RFC	

Die ersten Meldungen verändern nicht den Status bereits deklarerierter Listener. Man kann also mit LI-STEEN oder LAD mehrere Listener deklarieren, und anschließend mit OUTA den ALPHA-Inhalt an diese Geräte und zusätzlich die primäre Einheit schicken. Danach sind alle Geräte im Ruhezustand (wegen UNL). OUTA sendet keine NULL-Bytes. OUTA hängt ein CR/LF an den Text an, wenn Flag 17 = 0.

OUT... (X-I/O)

IDY00	IDY00	*
AAU	AAU	*
RFC	RFC	*
AAD1	AAD(sel+1)	*
LADsel	LADsel	*
RFC	RFC	*
DABa1	DABa1	
***	***	beliebige Datenbytes
DABan	DABan	
UNL	UNL	*
RFC	RFC	*

Wieviele Datenbytes gesendet werden, hängt von der Funktion ab: Bei allen OUTA.. Funktionen wird der erste - vom NULL-Byte verschiedene - Charakter nicht gesendet. OUTAN : N BYtes werden gesendet, wobei N im X Register angegeben ist (N<24). OUTAE : Der ganze ALPHA-Registerinhalt wird gesendet, dabei wird der letzte Charakter nicht als DABan, sondern als END-Meldung ENDan gesendet. OUTAC : Der ganze ALPHA-Registerinhalt wird gesendet, und die Daten-Meldung DABx angehängt. OUTACL : Wie bei OUTAC, nur werden hier DAB(CR) und DAB(LF) angehängt. So kann mit "D", OUTACL ein Zeilenvorschub bewirkt werden. OUTXB : Es wird nur eine Daten-Meldung DABx gesendet. Im ADROFF-Modus werden die Meldungen, die von einem "" gefolgt werden, nicht gesendet.

POLL (X-I/O)

IDY00	IDY00
AAU	AAU
RFC	RFC
AAD1	AAD(sel+1)
IDY00	IDY??

Der dezimale Wert des Datenbytes der zurückkehrenden IDY-Meldung wird ins X-Register geschrieben.

POLLD (X-I/O)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
LADsel	LADsel	
RFC	RFC	
PPD	PPD	Parallel Poll Disable
RFC	RFC	
UNL	UNL	
RFC	RFC	

POLLE (X-I/O)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
LADsel	LADsel	
RFC	RFC	
PPEX	PPEX	Parallel Poll Enable
RFC	RFC	
UNL	UNL	
RFC	RFC	

POLLUNC (X-I/O)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
PPU	PPU	Paral.Poll Unconfig
RFC	RFC	

PWRDN (IL)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
LPD	LPD	Loop Power Down
RFC	RFC	

Dies ist die einzige Kommando-Meldung, die nicht vor Weitergabe des RFC ausgeführt wird, sondern erst danach (Wenn das Gerät einmal ausgeschaltet ist, kann es ja keine Meldung mehr weitergeben - auch keine RFC!!).

PWRUP (IL)

Siehe Einschaltsequenz.

RCLSEL (X-I/O)

Diese Funktion sendet Keine Meldungen über die Schleife. Sie ruft nur die durch SELECT abgespeicherte Adresse der primären Einheit ins X Register zurück.

REMOTE (IL)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
REN	REN	Remote ENable
RFC	RFC	
IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
LADsel	LADsel	
RFC	RFC	
CMD00	CMD00	
RFC	RFC	
UNL	UNL	
RFC	RFC	

SELECT (IL)

Diese Funktion sendet keine Meldungen über die Schleife. Die Adresse der primären Einheit wird im IL-Chip abgespeichert, von wo sie jederzeit abrufbar ist, um in LAD oder TAD-Meldungen integriert zu werden. Siehe auch RCLSEL.

SEND (X-I/O)

CMDx	CMDx	Meldungen der CMD
RFC	RFC	

SRQ? (X-I/O)

IDY00	IDY00
AAU	AAU
RFC	RFC
AAD1	AAD(sel+1)
IDY00	IDY??

Die Identify-Meldung (C0-Bit) wird auf ihrem Weg durch die Schleife verändert, wenn ein Service Request vorliegt. Der Rechner erhält dann eine IDYnn(SRQ) zurück.

STAT (X-I/O)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
TADsel	TADsel	
RFC	RFC	
SST	DABa1	
DABa1	***	beliebige Datenbytes
***	***	
***	DABan	
DABan	ETO	
UNT	UNT	
RFC	RFC	

Die Primäre Einheit sendet alle ihre Status-Bytes. Falls es mehr als 23 sind, werden nur diese im ALPHA-Register abgelegt. Vor den empfangenen String setzt der Rechner immer ein "S".

STOPIO (IL)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
IFC	IFC	InterFace Clear
RFC	RFC	

TAD (X-I/O)

TADx	TADx
RFC	RFC

TRIGGER (IL)

IDY00	IDY00	
AAU	AAU	
RFC	RFC	
AAD1	AAD(sel+1)	
LADsel	LADsel	
RFC	RFC	
GET	GET	Group Execute Trigger
RFC	RFC	
UNL	UNL	
RFC	RFC	

UNL (X-I/O)

UNL	UNL
RFC	RFC

UNT (X-I/O)
 UNT UNT
 RFC RFC

XFER... (X-I/O)

IDY00	IDY00	*
AAU	AAU	*
RFC	RFC	*
AAD1	AAD(sel+1)	*
TADsel	TADsel	*
RFC	RFC	*
LADx	LADx	*
RFC	RFC	*
SDA	DABa1	
DABa1	***	
***	***	beliebige Datenbytes
***	DABan	
NRD	NRD	
DABan	ETO	
UNT	UNT	
RFC	RFC	
UNL	UNL	*
RFC	RFC	*

Die Bedingung, unter der der Rechner den Daten-Transfer abbricht, hängt von der Funktion ab:

XFERN : Es werden N Daten-Meldungen abgewartet.

XFERE : Es wird nach einer END-Meldung abgebrochen.

XFERC : Es wird nach einer DABx abgebrochen.

XFERCL : Es wird nach DAB(LF) abgebrochen.

XFER : Hier unterbricht der Rechner den Datenfluß nicht (keine NRD-Meldung)! Er wartet ab, bis ihn eine ETO-Meldung erreicht.

Im ADROFF-Modus werden die Meldungen, die von einem "*" gefolgt werden, nicht gesendet. Das heißt, daß dann der im X-Register angegebene Wert nicht als Listener-Adresse in die Sequenz eingeht. Dennoch ist hier Vorsicht geboten: Ein Fehler im X-I/O-Modul veranlaßt, daß dieser Wert dennoch auf Legalität (0..x-31) geprüft wird, was eventuell zu einer Fehlermeldung führen kann.

Wenn man sich die Sequenzen genauer ansieht, erkennt man, daß deklarierte Talker und Listener nur am Ende jeder Sequenz mit UNT und/oder UNL in den Ruhezustand versetzt werden. Das hat zur Folge, daß Funktionen wie OUTA (IL) und OUT,* (X-I/O) durch bereits deklarierte Talker (mit TAD deklariert) in ihrer Ausführung gestört werden, was nach einer längeren Wartepause zu einem TRANSMIT ERR führt.

Die Drucker- und Laufwerkfunktionen sind nicht aufgeführt. Der Grund dafür ist, daß die dazugehörigen IL-Sequenzen sehr lang und oft kompliziert sind. Damit wir dennoch ein Bild davon bekommen wie das Laufwerk gesteuert wird, folgt nun eine Reihe von kleinen Programmen, die es uns erlauben, die File-Information, die im Directory gespeichert ist, zu lesen und zu verändern. Diese Programme setzen den Besitz des X-I/O-Moduls voraus. Manch einer wird sehr schnell feststellen, daß ich die Unterprogramme mit lokalen (numerischen) Label hätte versehen können, um Programmbytes zu sparen. Ich glaube jedoch, daß wir bei diesem schwierigen Thema, das das IL-Protokoll darstellt, versuchen sollten, so klar wie möglich zu sein, und das wird hier durch die Verwendung von globalen (Alpha-) Label erreicht.

Noch eine wichtige Anmerkung, bevor es losgeht: Das Betriebssystem sucht ständig nach einem Drucker, um die Stellung des MODE-Schalters festzustellen. Die dabei ausgesendeten IL-Meldungen stören unsere Programme erheblich. Um das zu umgehen, stellen wir den MODE-Schalter auf MAN und vermeiden es, die Programme während ihrer Ausführung zu unterbrechen.

Nun zum Kassettenlaufwerk. Es dient der Speicherung von Programmen, numerischer und alphanumerischer Information - kurz von Daten. Das Fassungsvermögen einer Kassette beträgt 128 kByte (131 072 Byte). Damit der Zugriff auf ein bestimmtes Byte erleichtert wird, ist das Band physikalisch in bestimmte Abschnitte eingeteilt, von denen jeder einen kleinen Teil der Gesamtinformation enthält. Da sind zunächst zwei Spuren: 0 und 1. Jede Spur ist in 256 Records eingeteilt (von 0 bis 255), die wiederum jeweils 256 Bytes fassen. Außerdem steht in jedem Record die Recordnummer und eine Quersumme der 256 Bytes. Diese zuletzt genannte Information ist nur der Schreib- und Leseelektronik zugänglich, die damit die hohe Qualität der Datenspeicherung garantiert. Der Schreib- und Lesemechanismus kann nur ganze Records aufs Band schreiben oder davon lesen. Während größere Computer diese Records immer ganz übernehmen und in ihrem RAM-Speicher ablegen, um dann einzelne Bytes zu interpretieren oder zu verändern, ist man hier einen anderen Weg gegangen. Der Grund ist der relativ kleine Speicher im HP-41. Das Laufwerk enthält einen Pufferspeicher (Puffer 0) der die Aufgabe eines Zwischenspeichers erfüllt. Von hier aus kann der HP-41 jedes einzelne Byte individuell über die Schleife lesen oder verändern.

Um ein Byte ansprechen zu können müssen wir dem Laufwerk seine genaue Position angeben können. Dazu benötigen wir die Bandposition des Records: Spur und Recordnummer (trc# und rec#) und die Position des Bytes in diesem Record (oder im Puffer): Bytenummer (byt#). Die Bandposition können wir mithilfe des DDL4-Kommandos angeben, die Bytenummer mithilfe von DDL3 (setzt den Bytezeiger für den Puffer). Sei ms die Adresse unseres Laufwerks, dann reicht folgende Sequenz aus, die Position des Bytes trc#/#rec#/#byt# eindeutig anzugeben:

LADms	LADms
RFC	RFC
DDL4	DDL4
RFC	RFC
DABtrc#	DABtrc#
DABrec#	DABrec#
DDL3	DDL3
RFC	RFC
DABbyt#	DABbyt#
UNL	UNL
RFC	RFC

Um die aktuelle Position des Laufwerks bestimmen zu können, steht uns das DDT3-Kommando zur Verfügung:

TADms	TADms
RFC	RFC
DDT3	DDT3
RFC	RFC
SDA	DABtrc#
DABtrc#	DABrec#
DABrec#	DABbyt#
NRD	NRD
DABbyt#	ETO
UNT	UNT
RFC	RFC

Die drei empfangenen Bytes geben uns trc#, rec# und byt# an. Da das Laufwerk diese drei Bytes ständig wiederholt, muß man den Datenfluß mit NRD unterbrechen.

Ist das Laufwerk richtig positioniert, kann man Information von Puffer 0 auslesen oder hineinschreiben. Dabei inkrementiert sich der Bytezeiger automatisch. Überschreitet er dabei das Ende des Puffers, passiert folgendes: Beim Lesen wird der nächste Record vom Band in den Puffer kopiert; beim Schreiben wird der gegenwärtige Record aufgezeichnet und das Band positioniert sich auf den nächsten Record.

Die genaue Bedeutung der verschiedenen geräteabhängigen Talker- und Listener-Kommandos (DDT und DDL) ist aus den Bedienungshandbüchern der verschiedenen Geräte ersichtlich.

Hier die versprochenen Programme:

PRDIR: 1170

Dieses Programm druckt die gesamte im Directory enthaltene Information aus. Es setzt ein Kassettenlaufwerk und eine "Sichteinheit" (Bildschirm oder Drucker) in der Schleife voraus. Der MODE-Schalter des Druckers sollte auf MAN stehen. Das Programm wird mit XEQPRDIR gestartet; Eingaben sind nicht nötig.

Registerbelegung: **SIZE=14**

Reg	Inhalt: (Siehe auch Progr. RDE)
00	Adresse der alten prim. Einheit
01	nicht benutzt
02	nicht benutzt
03	trc#
04	rec#
05	byt#
06	ersten 6 Buchstaben des Filenamens
07	7. Buchstabe des Filenamens
08	TT
09	PP
10	LL
11	KK
12	S
13	Schleifenkontrollwort

Zeile 2-3: sucht eine Sichteinheit und setzt Flag 21 entsprechend; **4-5:** verzweigt zur Fehlermeldung, falls nicht gefunden; **6-9:** sucht einen Massenspeicher und verzweigt zur Fehlermeldung, falls nicht gefunden; **10-17:** Bestimmt die maximale Anzahl von Directory Einträgen NNN und druckt sie aus; **18-20:** druckt eine Leerzeile; **21-25:** errechnet das Kontrollwort 1, NNN und legt es in R13 ab; **26-31:** setzt Laufwerk in Lesemodus auf Byte 0/2/0; **Zeile 32:** Kopf der Hauptschleife; **33-35:** liebt einen Directory-Eintrag und druckt die Information aus; **36-37:** falls Ende des Directory erreicht, schließe ab; **38-39:** inkrementiert Kontrollwort und verzweigt zum Kopf der Schleife; **40-46:** schließt den Lesevorgang ab, druckt "END OF DIR", schließt die Schleife; **Zeile 47-48:** generiert Fehlermeldung 3; **49-52:** generiert Fehlermeldung 2 (oder 3) und schließt Schleife.

SELVIEW: 1171

Dieses Programm sucht nach einem Video Interface. Findet es keines, sucht es einen Drucker. Das gefundene Gerät wird als primäre Einheit deklariert. **Zeile 2-3:** sucht Video Interface; **4-5:** falls gefunden, gehe zu LBL 01; **6-7:** sucht Drucker; **8-9:** falls nicht gefunden, kehre mit X=0 zum Hauptprogramm zurück; **10-13:** selektiert die gefundene Sichteinheit, versetzt sie in den Einschaltzustand und setzt Flag 21, das später zur Ausgabesteuerung dient.

SELAID: 1172

Dieses Programm sucht nach einer Einheit mit einer AID wie sie im X-Register angegeben ist. Wird eine solche Einheit gefunden, wird sie zur primären Einheit deklariert. Die Adresse der alten primären Einheit wird in R00 abgelegt. Benötigt: X, T, L, Flags 32 und 34. **Zeile**

	CAT 1	PRP. "PRDIR"	PRP "RDE"	PRP "PREAD"
LBL*PRDIR		4:00PM 17.12	4:03PM 17.12	4:02PM 17.12
END	144 BYTES	<u>01+LBL "PRDIR"</u>	<u>01+LBL "RDE"</u>	<u>01+LBL "PREAD"</u>
LBL*RDX		CF 21 XEQ "SELVIEW"	7 INAN ATOXL ATOXL	XEQ 01 TAD 2 DDT
LBL*WDX		X=0? GTO 05 16	XTOAL 255 - X=0? RTN	XEQ "BUSY?" LAD 3 DDL
END	128 BYTES	XEQ "SELAID" X=0?	ASTO 06 ASHF ASTO 07	RCL 05 OUTXB RCLSEL
LBL*CLRTRN		GTO 04 XEQ "DIRE?" UNT	3 XEQ 10 STO 08 2	TAD RTN
END	19 BYTES	RCL 00 LAD "DNEWM: "	XEQ 10 STO 09 2	
LBL*RDE		RDN ARCL X OUTACL "D"	XEQ 10 STO 10 8	<u>15+LBL "PWRIT"</u>
LBL*PRINFO		OUTACL UNL .1 % 1 +	XEQ 10 STO 11 INXB	XEQ 01 LAD 3 DDL
END	221 BYTES	STO 13 0 STO 03	STO 12 INXB FC? 21	RCL 05 OUTXB 6 DDL
LBL*WDE		STO 05 2 STO 04	RTN	PSE PSE RTN
END	93 BYTES	XEQ "PREAD"		
LBL*SELVIEW			<u>31+LBL "PRINFO"</u>	<u>27+LBL 01</u>
END	34 BYTES	32+LBL 01	FIX 0 CF 29 UNL UNT	RCLSEL LAD 4 DDL
LBL*PREAD		RCLSEL TAD XEQ "RDE"	RCL 00 LAD	RCL 03 OUTXB RCL 04
LBL*PWRIT		X=0? GTO 02 ISG 13	"DEINTRAG: " ARCL 13	OUTXB
LBL*BUSY?		GTO 01	OUTACL "DNAME: "	<u>36+LBL "BUSY?"</u>
END	110 BYTES		ARCL 06 ARCL 07 OUTACL	<u>37+LBL 05</u>
LBL*ERROR		40+LBL 02	"DSTART: " ARCL 09	INSTAT RDN FS? 05
END	107 BYTES	UNT RCL 00 LAD	OUTACL "DLAENGE: "	GTO 05 RCLSEL FC? 04
LBL*SELAID		"DEND OF DIR" OUTACL	ARCL 10 OUTACL	RTN 4 GTO "ERROR" END
END	27 BYTES	GTO "RESEL"	"DKOPIE: " ARCL 11	PRP ""
LBL*DIRE?			OUTACL "DTYP: "	
END	29 BYTES	47+LBL 05	ARCL 08 OUTACL	4:06PM 17.12
LBL*RESEL		1	"DSTATUS: " RCL 12	<u>01+LBL "ERROR"</u>
END	23 BYTES		X<>FIO FS? 03 "FS,"	GTO IND X
.END.	03 BYTES	49+LBL 04	FS? 01 "FA," FS? 00	
		2 + GTO "ERROR" END	"FP" OUTACL X<>FIO	03+LBL 01
		PRP "RDX"	"D" OUTACL UNL RCLSEL	"FALSCH EING." GTO 10
			RTN	
		4:52PM 17.12		06+LBL 02
BEISPIEL :		<u>01+LBL "RDX"</u>		"KEIN LAUFW." GTO 10
		CF 08 GTO 04	73+LBL 10	
			INAN INXB 256 * INXB	
XEQ "PRDIR"		<u>04+LBL "WDX"</u>	+ END	09+LBL 03
		SF 08		"KEIN DRUCKER" GTO 10
			PRP "WDE"	
			4:03PM 17.12	12+LBL 04
NEWM: 104		06+LBL 04	<u>01+LBL "WDE"</u>	"BANDFEHLER" GTO 10
		STO 13 CF 21 16	"D" ARCL 06 ARCL 07	
EINTRAG: 1		XEQ "SELAID" X=0?	"f" 10 OUTAN	15+LBL 05
NAME : 17.DEC		GTO 01 2 GTO "ERROR"	RCL 08 0 XEQ 10	"FL NICHT GEF"
START : 15			RCL 09 2 XEQ 10	
LAENGE: 11		15+LBL 01	RCL 10 2 XEQ 10 32769	
KOPIE : 336		XEQ "DIRE?" RCL 13	ENTER↑ 6 XEQ 10	17+LBL 10
TYP : 57408		X>Y? GTO 02 X<=0?	RCL 11 0 XEQ 10	XEQ "CLRTRN" XEQ "RESEL"
STATUS:		GTO 02 1 - 8 / 2 +	RCL 12 OUTXB 32 OUTXB	AVIEW BEEP END
		STO 04 FRC 256 *	8 DDL RTN	PRP "SELAID"
		STO 05 FS? 08 GTO 03		4:01PM 17.12
EINTRAG: 2		XEQ "PREAD" XEQ "RDE"	31+LBL 10	<u>01+LBL "SELAID"</u>
NAME : PRDIR		GTO "RESEL"	1 - X<0? GTO 11 0	RCLSEL STO 00 RDN
START : 26			OUTXB RDN GTO 10	FINDAID X=0? RTN
LAENGE: 1				SELECT MANIO ADROFF
KOPIE : 144		38+LBL 03		END
TYP : 57472		XEQ "PWRIT" XEQ "WDE"	40+LBL 11	PRP "DIRE?"
STATUS:		GTO "RESEL"	RDN 256 / OUTXB FRC	4:02PM 17.12
			256 * OUTXB END	<u>01+LBL "DIRE?"</u>
			PRP "SELVIEW"	0 STO 03 STO 04 19
EINTRAG: 3		42+LBL 02	4:01PM 17.12	STO 05 XEQ "PREAD"
NAME : SELVIEW		1 GTO "ERROR" END	<u>01+LBL "SELVIEW"</u>	INXB 8 * END
START : 27		4:05PM 17.12	-48 FINDAID X=0?	PRP "RESEL"
LAENGE: 1		<u>01+LBL "CLRTRN"</u>	GTO 01 -32 FINDAID	4:02PM 17.12
KOPIE : 34		02 6	X=0? RTN	<u>01+LBL "RESEL"</u>
TYP : 57472		03+LBL 01		UNT UNL RCL 00 SELECT
STATUS:		04 DSE X	10+LBL 01	ADRON AUTOIO END **
		05 XEQ 01	SELECT SF 21 END	
END OF DIR		06 END		

2-3: legt die Adresse der alten primären Einheit in R00 ab; **4-5:** holt die AID ins X-Register zurück und sucht die entsprechende Einheit; **6-7:** falls nicht gefunden, kehre mit X=0 zum Hauptprogramm zurück; **8-11:** selektiere die gefundene Einheit und schalte die Schleife auf Manual- und ADROFF-Modus.

RESEL: 1173

Dieses Programm schließt das Hauptprogramm ab, indem es die alte primäre Einheit deklariert und die Schleife auf Automatik umschaltet. Zerstört T.

DIRE?: 1174

Dieses Programm ließt das Byte 0/0/19 vom Band und multipliziert seinen Wert mit 8. Das gelesene Byte gibt die Anzahl der für Directory-Einträge reservierten Records an. Da für jeden Eintrag 32 Bytes benötigt werden, gibt die Zahl der Records $\times 8$ die maximale Anzahl der anzulegenden Files an. Benötigt: X,Y,Z,T,L, Flags 0-7.

Zeile 2-6: legt die Byte Position in R03 bis R05 ab; **7:** positioniert das Band auf die gewünschte Stelle und versetzt das Laufwerk in den Lesemodus; **8:** liest ein Byte aus Puffer 0; **9-11:** multipliziert den Wert mit 8.

PREAD und PWRIT: 1175

Diese beiden Programme positionieren das Band auf das in R03 bis R05 angegebene Byte und versetzen das Laufwerk in den Lese- oder "partiellen Schreibmodus". Das Laufwerk muß als primäre Einheit deklariert sein. Benötigt: X,Y,Z,T, Flags 0-7.

Zeile 2: positioniert das Band auf einen bestimmten Record; **3-5:** versetzt das Laufwerk in den Lesemodus; **6:** wartet, bis das Laufwerk mit dem Lesen eines Record fertig ist; **7-11:** setzt den Bytezeiger für Puffer 0; **12-14:** macht das Laufwerk zum Talker. Zeile 16: wie Zeile 2; **17-21** setzt den Bytezeiger für Puffer 0; **22-26:** setzt den part. Schreibmodus und wartet, bis ein Record nach Puffer 0 kopiert ist. **Zeile 27-35:** positioniert das Band auf einen bestimmten Record.

Zeile 36-41: liest das Statusbyte des Laufwerks und prüft, ob dieses noch "busy" (am Arbeiten) ist; dieser Vorgang wird solange wiederholt, bis Flag 5 clear; **42:** da INSTAT das Laufwerk in den Ruhezustand versetzt, wird es später nötig, dieses entweder zum Talker oder zum Listener zu deklarieren; um dabei einige Bytes zu sparen, wird hier die Adresse ins X Register geschrieben; **43-44:** falls kein Bandfehler vorliegt, gehe zurück zum Hauptprogramm; **45-46:** generiert Fehlermeldung 4.

RDE: 1176

Dieses Programm liest bei der gegenwärtigen Bandposition beginnend einen Directory-Eintrag von der primären Einheit und speichert die darin enthaltene Information in R06 bis R12 ab. Ein solcher Eintrag ist 32 Bytes lang und ist folgendermaßen aufgebaut: jeder Buchstabe entspricht einem Byte

NNNNNNNNNTT00PP00LL000000CCKKSE

N...N ist das Namenfeld, der HP-41 benützt nur die ersten 7 Stellen. TT gibt den File-Typ an:

TYP	hex	dec
ASCII	00 01 =	1
Write All	E0 40 =	57408
Key File	E0 50 =	57424
Status File	E0 60 =	57440
Programm File	E0 80 =	57472
Data File	E0 D0 =	57552

PP kodiert die Position des ersten Records eines Files: 1. Byte = trc#, 2. Byte = rec#: Jedes File beginnt mit einem neuen Record.

LL kodiert die Anzahl Records, die das File belegt.

CC Diese beiden Bytes sind immer hex 80 01. KK gibt die Kopierinformation an. Das ist die Anzahl Register (bei Programmen die Anzahl Bytes), die man im Rechner braucht um das File lesen zu können.

S ist der Filestatus, in dem steht, ob ein File gesichert ist und ob ein Programm privat ist oder "autostartet".

E ist das letzte Byte, es ist immer hex 20. Alle anderen Bytes sind NULL-Bytes (hex 00).

Da wir beim Lesen dieser Information immer wieder einige Bytes überspringen müssen, habe ich ein kleines Unterprogramm geschrieben, das diese Arbeit übernimmt: LBL 10. Benötigt: X,Y,Z,T,L,ALPHA. Das Laufwerk bleibt im Lesemodus.

Zeile 2-4: ließt den Namen und zerstört das von INAN vorgesezte "D"; **5-10:** prüft, ob der erste Buchstabe des Namenfeldes ein Byte hex FF (dec 255) ist: dann ist eine unbeschriebene Bandstelle erreicht, mit anderen Worten das Ende des benutzten Platzes im Directory: in diesem Fall Rückkehr zum Hauptprogramm mit X=0; **11-13:** speicht den Namen ab; **14-16:** überspringt die übrigen Bytes des Namenfeldes und liest TT; **17-19:** überspringt 2 Bytes und liest PP; **20-22:** überspringt 2 Bytes und liest LL; **23-25:** überspringt 8 Bytes und liest KK; **26-27** liest S; **28:** überspringt das letzte Byte; **29-72:** falls Flag 21 clear: kein Ausdruck; falls Flag 21 set: drucke die Fileinformation aus, Druckeradresse steht in R00.

Zeile 74: überspringt die im X-Register angegebene Anzahl von Bytes; **75-80:** liest 2 Bytes und errechnet $(1 \text{ Byte} * 256 + 2 \text{ Byte})$.

WDE: 1177

Dieses Programm schreibt einen Directory Eintrag auf das Band, beginnend mit der gegenwärtigen Bandposition. Es ist die Umkehrung von RDE. Die Registerbelegung ist dieselbe. Benötigt werden X,Y,Z,T,L, ALPHA. Das Laufwerk bleibt im partiellen Schreibmodus.

Zeile 2-7: schreibt den Filenamen; **8-25:** schreibt die übrige Fileinformation; **26-27:** schreibt das letzte Byte des Eintrags; **28-30:** schließt den Schreibvorgang ab. **Zeile 31-35:** dekrementiert X-Wert (=Anzahl der NULL Bytes) und prüft, ob noch ein NULL-Byte angehängt wird: falls nein, gehe zu LBL 11; **36-39:** hängt ein NULL-Byte an und kehrt zur Schleife zurück; **40-49:** teilt den Y-Wert in zwei Bytes und sendet diese ans Laufwerk.

CLRTN: 1178

Löscht alle Rücksprungadressen.

Zeile 2: Initialisierung zum Löschen von 6 Rücksprungadressen; **3:** Kopf der Schleife; **4-5:** dekrementiert Zähler und ruft sich selbst solange als Unterprogramm auf bis x=0; **6:** löscht die Rücksprungadressen durch mehrmalige Ausführung.

RDX und WDX: 1179

Diese beiden Programme erlauben das Lesen oder Schreiben des x-ten Directory-Eintrags. Als Eingabe ist nur die Nummer des Eintrags im X-Register nötig. Das Programm sucht nach einem Laufwerk, somit braucht dieses nicht bereits als primäre Einheit deklariert zu sein.

Zeile 2-3: Flag 8 clear = Lesen; **5:** Flag 8 set = Schreiben; **7:** Zwischenspeichern der Eintragsnummer; **8:** wir wollen keinen Ausdruck beim Aufruf von RDE; **9-15:** suche ein Laufwerk und - falls nicht gefunden - generiere Fehlermeldung 2; **16-21:** Legalitätskontrolle der Eingabe: $0 < x <= NNN$; **22-32:** Errechnen der Position des Fileeintrags auf dem Band; **33-34:** falls Schreiben, gehe nach LBL 03; **35-37:** Lesen des Eintrags und Schließen der Schleife; **38-41:** Schreiben des Eintrags und Schließen der Schleife; **42-45:** generiert Fehlermeldung 1.

ERROR: 1180

Dieses kleine Programm erzeugt die verschiedenen Fehlermeldungen. Welche Meldung gezeigt wird, ist von der Eingabe im X-Register abhängig. Damit keine Rücksprungadressen im Stapel bleiben, wird dieser vollständig geleert. Die Schleife wird geschlossen, indem sie auf Automatik und ADRON-Modus geschaltet wird.

Zeile 2-16: schreibt verschiedene Fehlermeldungen ins ALPHA-Register; **17-19:** löscht Rücksprungadressen und schließt die Schleife; **20-22:** gibt optisch und akustisch die Fehlermeldung aus.

Diese Programme bilden einen gemeinsamen Block. PRDIR benötigt alle anderen Programme außer RDX, WDX und WDE. RDX und WDX benötigen alle außer PRDIR. Während PRDIR zur manuellen Verwendung gedacht ist, können RDX und WDX auch als Unterprogramme aufgerufen werden.

DUMP 1181

Das folgende Programm darf nicht zusammen mit den obengenannten Programmen in den Rechner geladen werden, da es dieselben globalen Label verwendet, die aber nicht dieselben Funktionen ausführen.

DUMP: Dieses Programm erlaubt uns jedes beliebige Byte auf einer Kasette "anzusehen" und zu verändern. Da das Programm nach einem Laufwerk sucht, ist es nicht nötig, dieses als primäre Einheit zu deklarieren. Sind mehrere Laufwerke in der Schleife, wird das Programm auf das zugreifen, das als erstes auf die augenblickliche primäre Einheit folgt.

Das Programm wird mit XEQ"DUMP" gestartet. Nach einer kurzen Pause meldet es sich mit dem Prompt: "START : T ^ A R ^ B". Hier gibt man mit ENTER zuerst die Spur, dann den Record und das Byte ein und startet mit R/S. Das Programm liest nun den entsprechenden Record vom Band und legt ihn in einem ASCII-File ab (16 AS Records mit je 17 Buchstaben: das vorangehende "D" von INAN

LBL*DUMP 60*LBL 74
 LBL*NEXTKEY 3 GTO 05
 LBL*SHIFT
 LBL*ZIFFERN 63*LBL 62
 LBL*← 4 GTO 05
 LBL*SSST
 LBL*GTO 66*LBL 63
 LBL*ZIFF C 5 GTO 05
 LBL*MODISP
 LBL*OFF 69*LBL 64
 LBL*NEXTBYT 6 GTO 05
 LBL*GETBYT
 LBL*DISPTRB 72*LBL 52
 LBL*DISPBYT 7 GTO 05
 LBL*CHGBYT
 LBL*WRITREC 75*LBL 53
 LBL*READREC 8 GTO 05
 LBL*SEEKREC
 LBL*BUSY? 78*LBL 54
 LBL*ERROR 9 GTO 05
 LBL*CLRTN
 .END. 864 BYTES 81*LBL 35
 GTO 35

PRP "DUMP"

3:55PM 17.12
 01*LBL "DUMP"
 RCLSEL STO 00 16
 FINDAID X=0? GTO 97
 SELECT CLRDEV MANTO
 ADROFF 256 STO 01 10
 STO 02 CF 08 SF 09
 CF 21 SF 28 CF 29
 "RECORD" SF 25 FLSIZE
 42 FS?C 25 X=Y?
 FS?C 25 PURFL X=Y?
 CRFLAS 101X STO 03
 GTO 34

34*LBL "NEXTKEY"

35*LBL 99

TONE 9

37*LBL 01
 SF 04 TONE 5 GETKEY
 CF 04 X=0? GTO 01
 GTO IND X

45*LBL "SHIFT"

46*LBL 31
 FC?C 01 SF 01 GTO 99

50*LBL "ZIFFERN"

51*LBL 82

0 GTO 05

54*LBL 72

1 GTO 05

57*LBL 73

2 GTO 05

60*LBL 74
 3 GTO 05

63*LBL 62
 4 GTO 05

66*LBL 63
 5 GTO 05

69*LBL 64
 6 GTO 05

72*LBL 52
 7 GTO 05

75*LBL 53
 8 GTO 05

78*LBL 54
 9 GTO 05

81*LBL 35
 GTO 35

83*LBL 34
 GTO 34

85*LBL "←"
 86*LBL 44

FS?C 01 GTO 08 0
 FS? 09 GTO 05 X<> 06
 X<> 07 STO 08 GTO 07

96*LBL 05
 GTO 05

98*LBL "SSST"
 99*LBL 35

FC? 09 XEQ 90 RCL 03
 1 FS? 01 CHS +

GTO 09

108*LBL "GTO"
 109*LBL 34

FC? 09 XEQ 90
 "START : T↑R↑B" PROMPT

X<> Z RCL 01 * +
 RCL 01 * + GTO 09

122*LBL "ZIFF C"
 123*LBL 05

FC?C 09 GTO 06 0
 STO 07 STO 08 RDN

130*LBL 06
 X<> 08 X<> 07 STO 06

134*LBL "MODISP"
 135*LBL 07

ATOXR ATOXR ATOXR
 ARCL 06 ARCL 07

ARCL 08 AVIEW GTO 99

144*LBL "OFF"

145*LBL 08

FC? 09 XEQ 90 RCL 03

RCL 01 / FS?C 08

XEQ 91 ADRON AUTOIO

RCL 00 SELECT CLD

GTO 98

159*LBL "NEXTBYT"

160*LBL 09

X<> 03 RCL 01 / INT

RCL 03 RCL 01 / INT

X<>Y X=Y? GTO 10

FS?C 08 XEQ 91 RCL 03

RCL 01 / XEQ 92

178*LBL "GETBYT"

179*LBL 10

"BYTE HOLEN" AVIEW

RCL 03 RCL 01 MOD 16

/ INT SEEKPT GETREC

RCL 03 16 MOD 1 +

STO 05 ATOXX STO 04

198*LBL "DISPTRB"

"ANZEIGE" AVIEW CLA

FIX 3 RCL 03 RCL X

RCL 01 MOD - LASTX

X<>Y RCL 01 / RCL X

RCL 01 MOD - LASTX

X<>Y RCL 01 / 3 101X

ST/ T ST/ Z / ARCL Y

ARCL Z "↑" " ARCL X

ATOXL 4 AROT ATOXL 6

AROT ASHF

236*LBL "DISPBYT"

SF 09 FIX 0 RCL 04 1

% INT ARCL X LASTX

FRC RCL 02 * INT

ARCL X LASTX FRC

RCL 02 * INT ARCL X

AVIEW GTO 99

258*LBL 02

259*LBL 03

260*LBL 04

261*LBL 11

262*LBL 12

263*LBL 13

264*LBL 14

265*LBL 15

266*LBL 21

267*LBL 22

268*LBL 23

269*LBL 24

270*LBL 25

271*LBL 32

272*LBL 33

273*LBL 41

274*LBL 42

275*LBL 43

276*LBL 51

277*LBL 61

278*LBL 71

279*LBL 81

280*LBL 83

281*LBL 84

GTO 99

283*LBL "CHGBYT"

284*LBL 90

"TAUSCHEN" AVIEW RCLPT

INT SEEKPT RCL 04

RCL 06 RCL 02 *

RCL 07 + RCL 02 *

RCL 08 + X=Y? RTN

SF 08 GETREC DELREC

RCL 05 YTOAX INSREC

RTN

309*LBL "WRITREC"

310*LBL 91

"SCHREIBEN" AVIEW

XEQ 93 LAD 2 DDL 0

SEEKPT 16 ENTER↑

321*LBL 16

VIEW Y GETREC OUTAN

DSE Y GTO 16 RTN

328*LBL "READREC"

329*LBL 92

"LESEN" AVIEW XEQ 93

TAD 2 DDT CF 08

"RECORD" CLFL 16

ENTER↑

341*LBL 17

VIEW Y INAN APPREC

DSE Y GTO 17 RTN

348*LBL "SEEKREC"

349*LBL 93

RCLSEL LAD 4 DDL

RCL Z RCL 01 / OUTXB

FRC RCL 01 * OUTXB

362*LBL "BUSY?"

363*LBL 94

INSTAL FS? 05 GTO 94

RCLSEL FC? 04 RTN

370*LBL "ERROR"

371*LBL 97

"ERROR" AVIEW

374*LBL "CLRTN"

375*LBL 98

"SCHLUSS" FC? 50 AVIEW

7

380*LBL 18

DSE X XEQ 18 .END.

bleibt auch hier erhalten, damit uns die NULL-Bytes nicht verloren gehen). Von hier holt sich das Programm das gewünschte Byte und bringt es zur Anzeige:
"T.RRR.BBB bbb"

Flag 4 zeigt an, daß das Programm auf einen Tastendruck wartet. Folgende Tasten sind dann aktiv:

0-9: der Wert des angezeigten Bytes kann verändert werden.

SHIFT: schaltet den Shiftindikator (Flag 1) um.

←: bei SHIFT aus: dient zur Korrektur der eingegebenen Ziffern oder - falls noch keine Zifferntaste gedrückt wurde - zum Löschen des Bytes. **Bei SHIFT an: schließt das Programm ab.**

SST: geht zum nächsten (SHIFT aus) oder zum vorherigen (SHIFT an) Byte.

GTO: meldet sich mit dem START-Prompt und erlaubt so das Springen zu jedem beliebigen Byte.

Um das Programmlisting lesbarer zu machen, habe ich eine ganze Reihe ALPHA-Label eingefügt, die über die Bedeutung der Abschnitte Auskunft geben. Manche Unterprogramme schreiben eine Meldung in die Anzeige, damit man sehen kann, welche Schritte gerade erledigt werden. Hier die detaillierte Beschreibung:

Registerbelegung: **SIZE=9**

Reg	Inhalt:
00	Adresse der alten prim. Einheit
01	256
02	10
03	Bytenummer
04	Bytwert
05	Position des Byte im ALPHA-Register
06	1. Digit
07	2. Digit
08	3. Digit

Flags	Bedeutung wenn set:
1	Shiftanzeige
4	Programm wartet auf Tastendruck
8	Record muß geschrieben werden
9	Byte wurde nicht geändert

Zeile 1-11: initialisiert die Schleife; **12-20:** initialisiert den Rechner; **21-30:** initialisiert das ASCII-File RECORD; **31-32:** forciert den Abschnitt NEXTBYTE, einen Record vom Band zu lesen; **33:** verzweigt zum GTO-Abschnitt.

Zeile 34-36: leitet die Tastenabfrage ein; **37-43:** Abfrageschleife: solange eine Taste gedrückt werden kann ist Flag 4 an (Sichtkontrolle); **37:** verzweigt zur Tastenroutine;

Zeile 45-49: schaltet die Shiftanzeige (Flag 1) um.

Zeile 50-80: Zifferneingaberoutinen, werden bei Abschnitt ZIFFC fortgeführt;

Zeile 81-84: beschleunigen die Suche nach LBL 35 und 34.

Zeile 85: Löschtasten-Routine; **86-88:** falls geschift, gehe zum Abschnitt OFF; **89-91:** falls noch keine Zifferneingabe erfolgte, lösche das ganze Byte; **92-95:** falls eine Zifferneingabe bereits erfolgte, lösche nur das letzte Digit;

Zeile 96-97: verkürzt die Sprungweite für GTO 05 (Zeilen 53-80).

Zeile 98-101: Austausch des Bytes falls nötig; **102 107:** Inkrementieren oder Dekrementieren der Bytenummer.

Zeile 108-111: Austausch des Bytes falls nötig; **112 121:** Eingabe-Prompt und errechnen der Bytenummer.

Zeile 122-129: falls Eingabe des ersten Digits: lösche die beiden anderen Digits (Das Löschen von Flag 9 forciert den Austausch der Bytes.); **130-133:** Einschleiben des neuen Digits von links; **134-143:** Ändern der Anzeige.

Zeile 144-147: Austausch des Bytes falls nötig; **148 152:** Schreiben des Records falls nötig; **153-158:** Wiederherstellen des alten Schleifenzustands.

Zeile 159-171: prüft, ob neues Byte im gleichen Record liegt wie das alte; **172-177:** falls nicht, schreibt alten Record, falls nötig, und liest neuen Record.

Zeile 178-197: ließt das Byte aus dem EM-File RECORD und legt es in R04 ab.

Zeile 198-235: zerlegt die Bytenummer in trc#, rec# und byt# und schreibt sie ins ALPHA-Register.

Zeile 236-257: zerlegt den Bytwert in drei Digits und hängt sie ans ALPHA-Register an.

Zeile 258-282: Leerroutine für die nicht belegten Tasten.

Zeile 283-308: tauscht das alte Byte im EM-File gegen das neue aus, falls sie verschieden sind. Das Setzen von Flag 8 forciert das Schreiben des EM-File auf das Band.

Zeile 309-316: positioniert das Band und bereitet es für den Schreibvorgang vor; **317-320:** positioniert das EM-File und initialisiert die Schreibschleife LBL 16; **321-327:** kopiert den EM-File auf das Band.

Zeile 328-335: positioniert das Band und bereitet es für den Lesevorgang vor; **336:** reinitialisiert das "Schreibflag"; **337-340:** initialisiert das EM-File und die Leseschleife LBL 17; **341-347:** kopiert einen Record vom Band in das EM-File.

Zeile 348-361: positioniert das Band auf den im X Register angegebenen Record.

Zeile 362-366: diese Schleife läuft solange, bis das Laufwerk nicht mehr "busy" ist; **367:** ruft Adresse des Laufwerks, um dieses später zum Listener oder Talker deklarieren zu können; **368-369:** zurück zum Hauptprogramm, falls kein Bandfehler vorliegt.

Zeile 370-373: erzeugt Fehlermeldung und initialisiert CLRTN.

Zeile 374-383: löscht alle Rücksprungadressen und schließt das Programm ab.

Jeder, der sich diese Programme näher ansieht, wird bemerken, daß sie bezüglich Speicherplatzbelegung und Schnelligkeit nicht optimiert sind, obwohl dies an manchen Stellen vielleicht leicht zu realisieren gewesen wäre. Der Grund dafür ist, daß diese Programme dem Erlernen des HPIL-Protokolls dienlich sein sollen. Dazu ist es nötig, den Programmablauf nachvollziehen zu können, und das ist bei den hier vorliegenden Programmen sicherlich möglich. Jedem bleibt es nun überlassen, sie zu optimieren - vielleicht können wir dann bald eine wesentlich kürzere und schnellere Version von DUMP abdrucken (aber bitte mit all dem Bedienungskomfort, den DUMP jetzt bietet!). Nun möchte ich abschließen und Euch schönes Experimentieren und Programmieren mit HPIL wünschen.

Wolfgang Baltes (155)
Guardinistraße 143
8000 München 70

CCD-ROM

Liebe Freunde, fast ist es geschafft !

Dank Eurer Mithilfe und Euren zahlreichen Zuschriften und Vorschlägen hat es uns nicht an sehr guten Funktionen gemangelt und das CCD-MODUL ist zum größten Teil fertiggestellt. Die Verhandlungen mit HP bezüglich der Herstellung des Moduls sind in vollem Gange. Wahrscheinlich findet Ihr im nächsten Prisma schon Zahlkarten für Eure Vorbestellungen. Auslieferungstermin wird voraussichtlich Mitte des Jahres sein. Da wir ein so großes Projekt natürlich nicht vorfinanzieren können, bitten wir darum, vom Subskriptionsangebot regen Gebrauch zu machen, und nicht zu warten, bis das Modul im Handel erhältlich ist, zumal es dann wesentlich teurer sein wird. Falls keine 600 Bestellungen zustande kommen, wird das Geld, das Ihr auf ein Sonderkonto überwiesen habt, zurückerstattet. Natürlich sind auch Händleranfragen erwünscht. Ein zusätzlicher Anreiz: Die Besteller des 200., 400. und 600. Moduls bekommen das CCD-MODUL umsonst. Alles Weitere zur Organisation des Projekts wird im nächsten Prisma erscheinen.

Hier noch ein paar weitere Funktionen:

LCON : LowerCase ON (Kleinbuchstabenmodus an)

LCOFF : LowerCase OFF (// aus)

Diese Funktionen ermöglichen es, Kleinbuchstaben direkt über das Tastenfeld einzugeben. Im LCON-Modus ist das Tastenfeld vierfach belegt. Mit den USER und SHIFT Tasten wird zwischen den Belegungen umgeschaltet. Alle vom Display darstellbaren Sonderzeichen sind vorhanden, sowie alle Kleinbuchstaben und einige Sonderzeichen wie das ESCAPE-Byte und so weiter. Wird ein Kleinbuchstabe eingegeben erscheint in der Anzeige solange der entsprechende Großbuchstaben, bis die Taste losgelassen wird. Erst dann wird dieser umgewandelt und meistens (außer "a" bis "e") als "Starburst" dargestellt. ASTO, ARCL, CLA sind weiterhin aktiv. Eine weitere Funktion (HEX) erlaubt die Eingabe eines beliebigen Bytes ohne den Modus zu verlassen! Dazu ein Beispiel:

Eingabe: Anzeige:

A	a
B	ab
C	abc
HEX	abc_
2	abc2_
9	abc)

Das herrliche dabei ist, daß es auch im Programmmodus funktioniert. Dieser Modus erschlägt somit alle Probleme der synthetischen Textzeilen auf einen Schlag.

ASN : neue Tastenzuordnungsfunktion

Nach XEQASN" erscheint in der Anzeige: ASN_... Jetzt kann ALPHA gedrückt werden und die Funktion geht in die uns bereits bekannte ASN Funktion über. Will man die Zuordnungsbytes als Dezimalbytes eingeben, so tippt man einfach den Wert des ersten davon ein. Will man die Eingabe in Hexadezimalzahlen machen, so tippt man zuerst ein "H"; zum Beispiel: RCL M hex 90 75

.END.

Eingabe: Anzeige:

H ASN H__
 9 ASN H9__
 0 ASN H90 __
 7 ASN H90 7__
 5 ASN H90 75 __
 SHIFT COS ASN H90 75 -24

Sobald jetzt dieselbe Taste gedrückt wird, wird RCL M ausgeführt.

Kennt man nur die XROM-Nummer einer Funktion, so tippt man ein X ein gefolgt von den Nummern. Mit dieser Funktion werden alle synthetischen Programme die diese Aufgabe bisher erfüllten überflüssig.

XQS : neue Execute-Funktion

Diese Funktion fragt wie die ASN Funktion und erlaubt die gleichen Eingaben. Anstatt die Funktion einer Taste zuzuordnen, wird sie hier ausgeführt.

Weitere Funktionen aus dem Synthetikblock des CCD MODUL sind PEEKB (hole Byte), PEEKR (hole Register), POKEB (speichere Byte) und POKER (speichere Register), SFX und CLFX (setze und lösche Flag X für alle Flags bis 55) usw..

Weitere Funktionen werden im nächsten Prisma vorgestellt. Zum Schluß noch ein Appell an alle, die die Synthetik erlernen wollen: Leichter als mit dem CCD-MODUL geht es nicht!!! Wartet mit dem Experimentieren bis unser ROM erscheint: 1. braucht Ihr dann keine dicken Bücher über Synthetik zu lesen; 2. erspart Ihr Euch jede Menge MEMORYLOSTs und 3. macht es viel viel mehr Spaß, mit dem CCD MODUL zu arbeiten - und wir müssen es ja schließlich wissen, denn wir können uns schon nicht mehr vorstellen, ohne dieses Modul zu arbeiten.

Bis bald Euer Wilfried (1637)

.END.

Programm zur Lösung der 2. geodätischen Hauptaufgabe nach der Gauß'schen Mittenbreitenformel

1. Allgemeines

Bei der Auswertung und Darstellung der im Rahmen der Landesvermessung anfallenden Vermessungsergebnisse zur Bestimmung von Lagefestpunkten spielt das Rotationsellipsoid als Bezugsellipsoid eine zentrale Rolle. Leider lassen sich die geometrischen Beziehungen auf einem Ellipsoid nicht mehr so einfach darstellen, wie dies auf der Kugel der Fall ist. So ist es nicht verwunderlich, daß in der einschlägigen Literatur von 50 verschiedene Verfahren mit mehr oder weniger großem numerischen Aufwand beschrieben wird. C.F. Gauß hat, um zu bequemen Rechenformeln für die beiden geodätischen zu kommen, das Prinzip der Mittelbreite $B_m = 1/2 (B_1 + B_2)$ angewandt, was beispielsweise im Gegensatz zu dem Verfahren der Legendre'schen Reihen (1), den Vorteil seiner Kürze und größeren Reichweite hat.

Das vorliegende Programm löst folgende Problemstellung:

gegeben auf einem Bezugsellipsoid: geographische Breite B_1 und geographische Länge L_1 des Punktes P_1 , dto. für Punkt P_2
gesucht: Länge S der geodätischen Linie (= g.L. kürzeste Verbindung zwischen zwei

Punkten auf dem Rotationsellipsoid, außerdem ist an jedem Punkt der g.L. das Azimut konstant). P_1P_2 sowie die Azimute A_1 und A_2 dieser Linie in ihren Endpunkten P_1 und P_2 .

2. Benötigte Rechenformata:

$$S \cdot \sin A = N \cdot l \cos B_m \left\{ 1 - \frac{1}{24} L \left((1 - \sin B_m)^2 + \frac{v^2 - q(v^2 - 1)t^2}{v^4} \cdot b^2 \right) \right\}$$

$$S \cdot \cos A = M \cdot b \cos \frac{1}{2} \left\{ 1 + \frac{1}{24} \left((1 - 2(v^2 - 1)) (1 - \cos B_m)^2 + \frac{(v^2 - 1)(1 - t^2) \cdot 3}{v^4} b^2 \right) \right\}$$

$$\Delta A = 1 \sin B_m \left\{ 1 + \frac{1}{24} \left(2v^2 (1 \cos B_m)^2 + \frac{3 + 8(v^2 - 1) b^2}{v^4} \right) \right\}$$

Mit:

$$B_m = 1/2 (B_1 + B_2) \quad v^2 = 1 + e^{12} \cos^2 B_m = 1 + \gamma^2$$

$$b = \tan B_m \quad M = \frac{C}{\sqrt{3}}$$

$$l = L_2 - L_1 \quad N = \frac{C}{v}$$

$$b = B_2 - B_1 \quad c, e^{12} \text{ siehe Konstanten!}$$

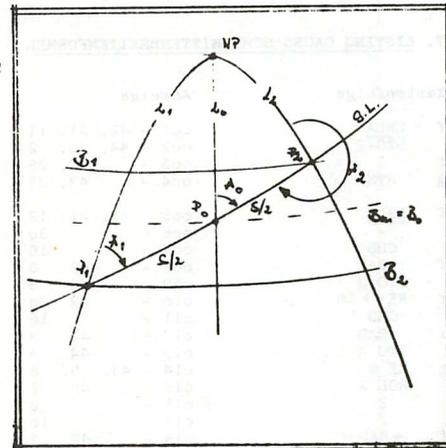
Womit ergibt sich:

$$s = \sqrt{b^2 \sin^2 A + S^2 \cos^2 A} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} R \rightarrow P$$

$$A_0 = \arctan \frac{S \sin A}{S \cos A}$$

$$A_1 = A_0 - \frac{\Delta A}{2}$$

$$A_2 = A_0 + \frac{\Delta A}{2} \pm 180^\circ$$



3. Konstanten einiger Rotationsellipsoiden

Bessel-Ellipsoid	internationales Ellipsoid	krassowsky-Ellipsoid
c = 63897336,849 m	c = 6399936,608 m	c = 6399698,902
e ¹² = 0,006719219	e ¹² = 0,006768170	e ¹² = 0,006738525

4. Einschränkungen - Genauigkeit

Das Programm kann keine beliebigen Entfernungen verarbeiten, seine Grenze liegt in unseren Breiten (zwischen 47° < B < 55°) bei ca. 100 km.

Bei maximaler Entfernung liegen die Genauigkeiten bei der Strecke im mm-Bereich und bei

den Azimuten bei ca. 1 bis 0,1 mgon. Fordert man höhere Genauigkeiten, so muß man den Entfernungsbereich verkleinern.

Bem.: Setzt man e¹² = Null und für C einen Radius R rein, so kann man das Programm zu Berechnungen auf der Kugel mit dem Radius R verwenden.

5. Zur Handhabung des Programms

Vorbereitung: Eingabe folgender Werte

a) e¹² in R₀ (zweite numerische Exzentrizität)

b) c in R₁ (Polkrümmungshalbmesser)
 c) Programm eingeben/und 0 Sto 2!!

1. Auf USER schalten
2. Eingabe der geographischen Länge des ersten Punktes
3. Eingabe der geographischen Breite des ersten Punktes
4. Speichern der Werte
5. Eingabe der geog. Länge des 2. Punktes
6. Eingabe der geog. Breite des 2. Punktes
7. Speichern des Wertes
8. Berechnung starten
9. Ausgabe der Strecke in Meter
10. Ausgabe des Azimuts A₁ in Altgrad
11. Ausgabe des Azimuts A₁ in Gon
12. Ausgabe des Azimuts A₂ in Altgrad
13. Ausgabe des Azimuts A₂ in Gon
14. Nächste Aufgabe vorbereiten
15. dann weiter mit Schritt 2

Eingabe	Key	Anzeige
	f USER	
L ₁ (HMS)	g H	
B ₁ (HMS)	g H	
	A	
L ₂ (HMS)	g H	
B ₂ (HMS)	g H	
	A	
	B	
		S (m)
	R/S	A ₁ (HMS)
	R/S	A ₁ (GON)
	R/S	A ₂ (HMS)
	R/S	A ₂ (HMS)
	R/S	0.00

6. Beispiel

a) Bestimmen Sie die Länge der geodätischen Linie, Azimut (A₁) und Gegenazimut (A₂) zwischen den Punkten Bonstetten und Wellham, deren geographische Koordinaten auf dem Bessel-Ellipsoid sich ergeben zu
 B_{BONST.} = 48° 26' 45."4355
 L_{BONST.} = 10° 42' 39."3215
 B_{WELL.} = 43° 48' 35."6813
 L_{WELL.} = 11° 03' 45."1103

Vorbereitung (analog 5.)
 0,006719219 in R 0
 6398786,849 in R 1
 0 in R 2 (vor Programmstart muß dies immer ausgeführt werden)

Ergebnis:
 S = 47835,624 m
 A₁ = 32° 05' 33".01 (88) ≙ 35,65833 (9) gon
 A₂ = 212° 21' 07".90 (6) ≙ 235,94688 (47) gon

b) Zum Vergleich führe man die Berechnung auf einer Kugel mit dem Radius R = C des Bessel-Ellipsoides durch ϕ in R 0
 6398786,849 in R₁
 0 in R₂

Ergebnis:
 S = 48006,460 m
 A₁ 32° 01' 00".48(90) = 35,574225 gon
 A₂ = 212° 16' 35".37(7) = 235,86277(06) gon

Der Unterschied zwischen Ellipsoid und Kugel mit dem Radius-Polbrpmungshalbmesser beträgt in der Strecke doch immerhin über 170 m.

c) Setzt man den mittleren Radius der Breite B_m ein, so erhält man:

$$R_m = \sqrt{M \cdot N} = \sqrt{\frac{C^2}{V^2}} = \frac{C}{V^2} = \underline{6380059,53 \text{ m}}$$

mit: B_m = 48°, 62793289
 R_m in R₁
 S = 47865,959(64) m
 A₁ = 32° 01' 00".48(90) = 35,574225 gon
 A₂ = 212° 16' 35".36(7) = 235,862770(6) gon
 In diesem Fall beträgt der Unterschied "nur noch" ca. 30 m.

7. LISTING GAUSS'SCHE MITTENBREITENFORMEL

Tastenfolge	Anzeige
f LBLA	001 - 42, 21, 11
STO+2	002 - 44, 40, 2
f I	003 - 42, 25
g RTN	004 - 43, 32
f LBLB	005 - 42, 21, 12
-	006 - 30
CHS	007 - 16
f →RAD	008 - 42 3
STO 3	009 - 44 3
f RE<>Im	010 - 42 30
CHS	011 - 16
f →RAD	012 - 42 3
STO 4	013 - 44 4
g CF 8	014 - 43, 5, 8
RCL 2	015 - 45 2
2	016 - 2
/	017 - 10
f →RAD	018 - 42 3
STO 2	019 - 44 2
g RAD	020 - 43 8
GSB 0	021 - 32 0
ENTER	022 - 36
1	023 - 1
-	024 - 30
STO 6	026 - 44 6
9	026 - 9
*	027 - 20
RCL2	028 - 45 2
tan	029 - 25
g x ²	030 - 43 11
*	031 - 20
-	032 - 30
GSB 0	033 - 32 0
g x ²	034 - 43 11
/	035 - 10
RCL 4	036 - 45 4
g x ²	037 - 43 11
STO 4	038 - 44 4
*	039 - 20
RCL 2	040 - 45 2
SIN	041 - 23
RCL 3	042 - 45 3
*	043 - 20
g x ²	044 - 43 11
-	045 - 30
2	046 - 2
4	047 - 4
/	048 - 10
1	049 - 1
+	050 - 40
RCL 2	051 - 45 2
COS	052 - 24
RCL 3	053 - 43 3
*	054 - 20
*	055 - 20
GSB 2	056 - 32 2
/	057 - 10
STO 5	058 - 44 5
1	059 - 1
RCL 6	060 - 45 6
2	061 - 2
*	062 - 20
-	063 - 30
RCL 2	064 - 45 2
COS	065 - 24
RCL 3	066 - 45 3
*	067 - 20
g x ²	068 - 43 11
*	069 - 20
RCL 6	070 - 45 6
3	071 - 3
*	072 - 20
1	073 - 1
RCL 2	074 - 45 2
tan	075 - 25
g x ²	076 - 43 11
-	077 - 30
*	078 - 20
GSB 0	079 - 32 0
g x ²	080 - 43 11
/	081 - 10
RCL 4	082 - 45 4
*	083 - 20

Bemerkung	Tastenfolge	Anzeige	Bemerkung
Eingabe von L,B.	+	084 -	40
Aufbau des imaginären Stäck's	2	085 -	2
	4	086 -	4
	/	087 -	10
	1	088 -	1
gleichzeitige Berechnung von l,b und Umwandlung in RAD	+ GSB 1	089 -	40
	/	090 -	32 1
	LCL 4	091 -	10
	√x	092 -	45 4
	*	093 -	11
	RCL 3	094 -	20
	2	095 -	45 3
Verlassen des Complex-Mode	/	096 -	2
	COS	097 -	10
		098 -	24
		099 -	20
1/2 (B ₁ +B ₂) = B _m	f x<>6	100 -	42, 4, 6
	8	101 -	8
	*	102 -	20
	3	103 -	3
	+	104 -	40
RAD-Mode	GSB 0	105 -	32 0
v ²	g x ²	106 -	43 11
	/	107 -	10
	RCL 4	108 -	45 4
	*	109 -	20
	GSB 0	110 -	32 0
	2	111 -	2
	*	112 -	20
	RCL 2	113 -	45 2
	COS	114 -	24
	RCL 3	115 -	45 3
	*	116 -	20
	g x ²	117 -	43 11
	*	118 -	20
	2	119 -	40
	/	120 -	2
	4	121 -	4
	/	122 -	10
	1	123 -	1
	+	124 -	40
	RCL 2	125 -	45 2
	SIN	126 -	23
	*	127 -	20
	RCL 3	128 -	45 3
	*	129 -	20
	STO 2	130 -	44 2
	RCL 5	131 -	45 5
	RCL 6	132 -	45 6
	g →P	133 -	43 1
	R/S	134 -	31
	x<>y	135 -	34
	g TEST 2	136 -	43, 30, 2
	GSB 3	137 -	32 3
	STO 3	138 -	44 3
	RCL 2	139 -	45 2
	2	140 -	2
	/	141 -	10
	STO 2	142 -	44 2
	-	143 -	30
	g →DEG	144 -	43 3
	f HMS	145 -	42 2
	R/S	146 -	31
	GSB 4	147 -	32 4
	R/S	148 -	31
	RCL 3	149 -	45 3
	RCL 2	150 -	45 3
	+	151 -	40
	g TT	152 -	43 26
	+	153 -	40
	g LAST X	154 -	43 36
	2	155 -	2
	*	156 -	20
	g TEST 8	157 -	43, 30, 8
	-	158 -	30
	g TEST 7	159 -	43, 30, 7
	R	160 -	33
	g →DEG	161 -	43 3
	f HMS	162 -	42 2
	R/S	163 -	31
	GSB 4	164 -	32 4
	R/S	165 -	31
	.	166 -	48
	STO 2	167 -	44 2

EL
PRGM REGS NEEDED: 13

REIHE 1 (1-5) CCD BARCODES
REIHE 2 (5-16) CCD BARCODES
REIHE 3 (16-25) CCD BARCODES
REIHE 4 (26-36) CCD BARCODES
REIHE 5 (37-46) CCD BARCODES
REIHE 6 (47-56) CCD BARCODES
REIHE 7 (57-58) CCD BARCODES

ELLIPSE
PRGM REGS NEEDED: 14

REIHE 1 (1-2) CCD BARCODES
REIHE 2 (2-9) CCD BARCODES
REIHE 3 (10-19) CCD BARCODES
REIHE 4 (19-26) CCD BARCODES
REIHE 5 (27-37) CCD BARCODES
REIHE 6 (38-46) CCD BARCODES
REIHE 7 (46-50) CCD BARCODES
REIHE 8 (51) CCD BARCODES

SX
PRGM REGS NEEDED: 6

REIHE 1 (1-6) CCD BARCODES
REIHE 2 (6-12) CCD BARCODES
REIHE 3 (13-16) CCD BARCODES

PRTN
PRGM REGS NEEDED: 48

REIHE 1 (1-4) CCD BARCODES
REIHE 2 (5-9) CCD BARCODES
REIHE 3 (10-14) CCD BARCODES
REIHE 4 (14-16) CCD BARCODES
REIHE 5 (16-20) CCD BARCODES
REIHE 6 (21-27) CCD BARCODES
REIHE 7 (27-30) CCD BARCODES
REIHE 8 (30-37) CCD BARCODES
REIHE 9 (37-47) CCD BARCODES
REIHE 10 (47-54) CCD BARCODES
REIHE 11 (54-59) CCD BARCODES
REIHE 12 (60-65) CCD BARCODES
REIHE 13 (66-71) CCD BARCODES
REIHE 14 (71-72) CCD BARCODES
REIHE 15 (72-76) CCD BARCODES

REIHE 16 (76-81) CCD BARCODES
REIHE 17 (81-84) CCD BARCODES
REIHE 18 (85-94) CCD BARCODES
REIHE 19 (95-101) CCD BARCODES
REIHE 20 (102-109) CCD BARCODES
REIHE 21 (110-115) CCD BARCODES
REIHE 22 (116-124) CCD BARCODES
REIHE 23 (125-132) CCD BARCODES
REIHE 24 (133-140) CCD BARCODES
REIHE 25 (140-149) CCD BARCODES
REIHE 26 (150) CCD BARCODES

WND
PRGM REGS NEEDED: 17

REIHE 1 (1-4) CCD BARCODES
REIHE 2 (5-10) CCD BARCODES
REIHE 3 (10-16) CCD BARCODES
REIHE 4 (16-23) CCD BARCODES
REIHE 5 (24-34) CCD BARCODES
REIHE 6 (35-37) CCD BARCODES
REIHE 7 (37-41) CCD BARCODES
REIHE 8 (42-44) CCD BARCODES
REIHE 9 (44-47) CCD BARCODES

PXR
PRGM REGS NEEDED: 8

REIHE 1 (1-5) CCD BARCODES
REIHE 2 (5-10) CCD BARCODES
REIHE 3 (10-18) CCD BARCODES
REIHE 4 (18-24) CCD BARCODES

CAT 2
PRGM REGS NEEDED: 12

REIHE 1 (1-2) CCD BARCODES
REIHE 2 (3-6) CCD BARCODES
REIHE 3 (6-13) CCD BARCODES
REIHE 4 (14-20) CCD BARCODES
REIHE 5 (20-26) CCD BARCODES
REIHE 6 (26-34) CCD BARCODES
REIHE 7 (34) CCD BARCODES

SW-2
PRGM REGS NEEDED: 17

REIHE 1 (1-4) CCD BARCODES
REIHE 2 (4-10) CCD BARCODES
REIHE 3 (10-12) CCD BARCODES
REIHE 4 (13-19) CCD BARCODES
REIHE 5 (20-25) CCD BARCODES
REIHE 6 (25-31) CCD BARCODES
REIHE 7 (32-38) CCD BARCODES
REIHE 8 (39-47) CCD BARCODES
REIHE 9 (47-48) CCD BARCODES

MEMORY
PRGM REGS NEEDED: 113

REIHE 1 (1-2) CCD BARCODES
REIHE 2 (2-7) CCD BARCODES
REIHE 3 (7-12) CCD BARCODES
REIHE 4 (12-17) CCD BARCODES
REIHE 5 (18-22) CCD BARCODES
REIHE 6 (22-28) CCD BARCODES
REIHE 7 (28-34) CCD BARCODES
REIHE 8 (34-39) CCD BARCODES
REIHE 9 (39-46) CCD BARCODES
REIHE 10 (46-53) CCD BARCODES
REIHE 11 (53-60) CCD BARCODES
REIHE 12 (61-70) CCD BARCODES
REIHE 13 (70-77) CCD BARCODES
REIHE 14 (78-86) CCD BARCODES
REIHE 15 (87-93) CCD BARCODES
REIHE 16 (93-98) CCD BARCODES
REIHE 17 (98-106) CCD BARCODES
REIHE 18 (106-115) CCD BARCODES
REIHE 19 (116-124) CCD BARCODES
REIHE 20 (124-132) CCD BARCODES
REIHE 21 (132-137) CCD BARCODES
REIHE 22 (137-141) CCD BARCODES
REIHE 23 (141-144) CCD BARCODES
REIHE 24 (144-150) CCD BARCODES
REIHE 25 (151-159) CCD BARCODES
REIHE 26 (160-164) CCD BARCODES
REIHE 27 (165-170) CCD BARCODES
REIHE 28 (170-178) CCD BARCODES

REIHE 29 (179-180) CCD BARCODES
REIHE 30 (189-196) CCD BARCODES
REIHE 31 (197-204) CCD BARCODES
REIHE 32 (204-207) CCD BARCODES
REIHE 33 (208-215) CCD BARCODES
REIHE 34 (215-223) CCD BARCODES
REIHE 35 (224-230) CCD BARCODES
REIHE 36 (230-237) CCD BARCODES
REIHE 37 (238-246) CCD BARCODES
REIHE 38 (246-254) CCD BARCODES
REIHE 39 (255-266) CCD BARCODES
REIHE 40 (266-272) CCD BARCODES
REIHE 41 (272-277) CCD BARCODES
REIHE 42 (278-282) CCD BARCODES
REIHE 43 (283-291) CCD BARCODES
REIHE 44 (291-299) CCD BARCODES
REIHE 45 (300-307) CCD BARCODES
REIHE 46 (308-312) CCD BARCODES
REIHE 47 (312-315) CCD BARCODES
REIHE 48 (315-322) CCD BARCODES
REIHE 49 (322-328) CCD BARCODES
REIHE 50 (329-336) CCD BARCODES
REIHE 51 (337-341) CCD BARCODES
REIHE 52 (342-348) CCD BARCODES
REIHE 53 (348-355) CCD BARCODES
REIHE 54 (356-360) CCD BARCODES
REIHE 55 (361-368) CCD BARCODES
REIHE 56 (368-372) CCD BARCODES
REIHE 57 (373-378) CCD BARCODES
REIHE 58 (378-385) CCD BARCODES
REIHE 59 (385-393) CCD BARCODES
REIHE 60 (393-402) CCD BARCODES
REIHE 61 (402-404) CCD BARCODES

PSL
PRGM REGS NEEDED: 291

REIHE 1 (1-5) CCD BARCODES
REIHE 2 (5-10) CCD BARCODES
REIHE 3 (10-18) CCD BARCODES
REIHE 4 (18-22) CCD BARCODES
REIHE 5 (23-26) CCD BARCODES

REIHE 6 (26-34) CCD BARCODES
REIHE 7 (34-37) CCD BARCODES
REIHE 8 (37-40) CCD BARCODES
REIHE 9 (40-42) CCD BARCODES
REIHE 10 (43-50) CCD BARCODES
REIHE 11 (50-56) CCD BARCODES
REIHE 12 (57-62) CCD BARCODES
REIHE 13 (62-67) CCD BARCODES
REIHE 14 (68-79) CCD BARCODES
REIHE 15 (79-79) CCD BARCODES
REIHE 16 (79-84) CCD BARCODES
REIHE 17 (85-89) CCD BARCODES
REIHE 18 (90-96) CCD BARCODES
REIHE 19 (96-102) CCD BARCODES
REIHE 20 (102-108) CCD BARCODES
REIHE 21 (108-114) CCD BARCODES
REIHE 22 (114-119) CCD BARCODES
REIHE 23 (119-122) CCD BARCODES
REIHE 24 (122-128) CCD BARCODES
REIHE 25 (129-134) CCD BARCODES
REIHE 26 (135-140) CCD BARCODES
REIHE 27 (140-149) CCD BARCODES
REIHE 28 (148-154) CCD BARCODES
REIHE 29 (154-160) CCD BARCODES
REIHE 30 (160-167) CCD BARCODES
REIHE 31 (168-174) CCD BARCODES
REIHE 32 (174-183) CCD BARCODES
REIHE 33 (184-190) CCD BARCODES
REIHE 34 (190-195) CCD BARCODES
REIHE 35 (195-204) CCD BARCODES
REIHE 36 (204-212) CCD BARCODES
REIHE 37 (213-220) CCD BARCODES
REIHE 38 (220-227) CCD BARCODES
REIHE 39 (228-234) CCD BARCODES
REIHE 40 (235-243) CCD BARCODES
REIHE 41 (243-255) CCD BARCODES
REIHE 42 (255-265) CCD BARCODES
REIHE 43 (265-271) CCD BARCODES
REIHE 44 (272-278) CCD BARCODES
REIHE 45 (278-286) CCD BARCODES

REIHE 46 (287-291) CCD BARCODES
REIHE 47 (291-297) CCD BARCODES
REIHE 48 (298-306) CCD BARCODES
REIHE 49 (306-312) CCD BARCODES
REIHE 50 (312-319) CCD BARCODES
REIHE 51 (319-325) CCD BARCODES
REIHE 52 (326-333) CCD BARCODES
REIHE 53 (334-342) CCD BARCODES
REIHE 54 (343-349) CCD BARCODES
REIHE 55 (350-354) CCD BARCODES
REIHE 56 (355-360) CCD BARCODES
REIHE 57 (361-366) CCD BARCODES
REIHE 58 (367-374) CCD BARCODES
REIHE 59 (375-379) CCD BARCODES
REIHE 60 (379-385) CCD BARCODES
REIHE 61 (385-390) CCD BARCODES
REIHE 62 (390-397) CCD BARCODES
REIHE 63 (397-403) CCD BARCODES
REIHE 64 (404-408) CCD BARCODES
REIHE 65 (409-419) CCD BARCODES
REIHE 66 (414-419) CCD BARCODES
REIHE 67 (420-426) CCD BARCODES
REIHE 68 (427-435) CCD BARCODES
REIHE 69 (436-441) CCD BARCODES
REIHE 70 (441-451) CCD BARCODES
REIHE 71 (451-456) CCD BARCODES
REIHE 72 (457-464) CCD BARCODES
REIHE 73 (464-470) CCD BARCODES
REIHE 74 (471-477) CCD BARCODES
REIHE 75 (477-484) CCD BARCODES
REIHE 76 (484-490) CCD BARCODES
REIHE 77 (490-495) CCD BARCODES
REIHE 78 (496-504) CCD BARCODES
REIHE 79 (504-511) CCD BARCODES
REIHE 80 (511-518) CCD BARCODES
REIHE 81 (518-524) CCD BARCODES
REIHE 82 (524-532) CCD BARCODES
REIHE 83 (532-538) CCD BARCODES
REIHE 84 (539-545) CCD BARCODES
REIHE 85 (545-553) CCD BARCODES

REIHE 86 (554-560) CCD BARCODES
REIHE 87 (561-573) CCD BARCODES
REIHE 88 (574-580) CCD BARCODES
REIHE 89 (580-585) CCD BARCODES
REIHE 90 (585-594) CCD BARCODES
REIHE 91 (594-600) CCD BARCODES
REIHE 92 (600-605) CCD BARCODES
REIHE 93 (606-612) CCD BARCODES
REIHE 94 (613-620) CCD BARCODES
REIHE 95 (621-625) CCD BARCODES
REIHE 96 (626-635) CCD BARCODES
REIHE 97 (635-638) CCD BARCODES
REIHE 98 (638-646) CCD BARCODES
REIHE 99 (646-654) CCD BARCODES
REIHE 100 (655-661) CCD BARCODES
REIHE 101 (661-668) CCD BARCODES
REIHE 102 (668-674) CCD BARCODES
REIHE 103 (675-679) CCD BARCODES
REIHE 104 (679-689) CCD BARCODES
REIHE 105 (690-698) CCD BARCODES
REIHE 106 (699-709) CCD BARCODES
REIHE 107 (710-716) CCD BARCODES
REIHE 108 (716-722) CCD BARCODES
REIHE 109 (722-727) CCD BARCODES
REIHE 110 (728-734) CCD BARCODES
REIHE 111 (734-741) CCD BARCODES
REIHE 112 (741-745) CCD BARCODES
REIHE 113 (746-753) CCD BARCODES
REIHE 114 (753-759) CCD BARCODES
REIHE 115 (760-771) CCD BARCODES
REIHE 116 (772-776) CCD BARCODES
REIHE 117 (776-782) CCD BARCODES
REIHE 118 (782-788) CCD BARCODES
REIHE 119 (789-796) CCD BARCODES
REIHE 120 (796-802) CCD BARCODES
REIHE 121 (802-806) CCD BARCODES
REIHE 122 (809-814) CCD BARCODES
REIHE 123 (815-821) CCD BARCODES
REIHE 124 (822-827) CCD BARCODES
REIHE 125 (828-834) CCD BARCODES

REIHE 126 (834-839) CCD BARCODES
REIHE 127 (839-846) CCD BARCODES
REIHE 128 (846-853) CCD BARCODES
REIHE 129 (853-858) CCD BARCODES
REIHE 130 (859-864) CCD BARCODES
REIHE 131 (864-870) CCD BARCODES
REIHE 132 (870-877) CCD BARCODES
REIHE 133 (877-882) CCD BARCODES
REIHE 134 (882-887) CCD BARCODES
REIHE 135 (887-893) CCD BARCODES
REIHE 136 (893-900) CCD BARCODES
REIHE 137 (900-907) CCD BARCODES
REIHE 138 (907-914) CCD BARCODES
REIHE 139 (915-919) CCD BARCODES
REIHE 140 (920-925) CCD BARCODES
REIHE 141 (926-932) CCD BARCODES
REIHE 142 (933-937) CCD BARCODES
REIHE 143 (938-944) CCD BARCODES
REIHE 144 (945-954) CCD BARCODES
REIHE 145 (954-963) CCD BARCODES
REIHE 146 (964-972) CCD BARCODES
REIHE 147 (972-980) CCD BARCODES
REIHE 148 (980-985) CCD BARCODES
REIHE 149 (985-992) CCD BARCODES
REIHE 150 (992-998) CCD BARCODES
REIHE 151 (999-1005) CCD BARCODES
REIHE 152 (1005-1010) CCD BARCODES
REIHE 153 (1011-1018) CCD BARCODES
REIHE 154 (1018-1024) CCD BARCODES
REIHE 155 (1024-1030) CCD BARCODES
REIHE 156 (1031-1036) CCD BARCODES
REIHE 157 (1037-1038) CCD BARCODES
LB
PRGM REGS NEEDED: 12
REIHE 1 (1-5) CCD BARCODES
REIHE 2 (6-14) CCD BARCODES
REIHE 3 (14-25) CCD BARCODES
REIHE 4 (26-33) CCD BARCODES
REIHE 5 (34-41) CCD BARCODES
REIHE 6 (42-49) CCD BARCODES
REIHE 7 (49-51) CCD BARCODES

STREIK
PRGM REGS NEEDED: 23

REIHE 1 (1-3) CCD BARCODES
REIHE 2 (4-7) CCD BARCODES
REIHE 3 (7-10) CCD BARCODES
REIHE 4 (10-13) CCD BARCODES
REIHE 5 (14-18) CCD BARCODES
REIHE 6 (18-21) CCD BARCODES
REIHE 7 (21-24) CCD BARCODES
REIHE 8 (24-29) CCD BARCODES
REIHE 9 (29-32) CCD BARCODES
REIHE 10 (32-35) CCD BARCODES
REIHE 11 (36-42) CCD BARCODES
REIHE 12 (43-45) CCD BARCODES

NEUER QTH-KENNER
PRGM REGS NEEDED: 90

REIHE 1 (1-3) CCD BARCODES
REIHE 2 (4-12) CCD BARCODES
REIHE 3 (13-23) CCD BARCODES
REIHE 4 (23-33) CCD BARCODES
REIHE 5 (34-45) CCD BARCODES
REIHE 6 (46-55) CCD BARCODES
REIHE 7 (56-63) CCD BARCODES
REIHE 8 (64-73) CCD BARCODES
REIHE 9 (74-82) CCD BARCODES
REIHE 10 (83-92) CCD BARCODES
REIHE 11 (93-103) CCD BARCODES
REIHE 12 (103-108) CCD BARCODES
REIHE 13 (108-115) CCD BARCODES
REIHE 14 (115-120) CCD BARCODES
REIHE 15 (120-125) CCD BARCODES
REIHE 16 (126-132) CCD BARCODES
REIHE 17 (132-138) CCD BARCODES
REIHE 18 (139-145) CCD BARCODES
REIHE 19 (146-153) CCD BARCODES
REIHE 20 (154-162) CCD BARCODES
REIHE 21 (163-172) CCD BARCODES
REIHE 22 (172-177) CCD BARCODES
REIHE 23 (177-180) CCD BARCODES
REIHE 24 (180-183) CCD BARCODES
REIHE 25 (184-190) CCD BARCODES

REIHE 26 (191-195) CCD BARCODES
REIHE 27 (195-197) CCD BARCODES
REIHE 28 (198-204) CCD BARCODES
REIHE 29 (205-217) CCD BARCODES
REIHE 30 (218-226) CCD BARCODES
REIHE 31 (227-236) CCD BARCODES
REIHE 32 (236-241) CCD BARCODES
REIHE 33 (241-245) CCD BARCODES
REIHE 34 (246-258) CCD BARCODES
REIHE 35 (259-268) CCD BARCODES
REIHE 36 (269-277) CCD BARCODES
REIHE 37 (278-285) CCD BARCODES
REIHE 38 (285-290) CCD BARCODES
REIHE 39 (290-292) CCD BARCODES
REIHE 40 (292-299) CCD BARCODES
REIHE 41 (300-304) CCD BARCODES
REIHE 42 (305-313) CCD BARCODES
REIHE 43 (313-318) CCD BARCODES
REIHE 44 (319-327) CCD BARCODES
REIHE 45 (327-333) CCD BARCODES
REIHE 46 (334-339) CCD BARCODES
REIHE 47 (340-346) CCD BARCODES
REIHE 48 (347-352) CCD BARCODES

PRDIR
PRGM REGS NEEDED: 22

REIHE 1 (1-3) CCD BARCODES
REIHE 2 (3-7) CCD BARCODES
REIHE 3 (7-10) CCD BARCODES
REIHE 4 (10-14) CCD BARCODES
REIHE 5 (14-19) CCD BARCODES
REIHE 6 (20-30) CCD BARCODES
REIHE 7 (31-35) CCD BARCODES
REIHE 8 (35-41) CCD BARCODES
REIHE 9 (41-44) CCD BARCODES
REIHE 10 (44-47) CCD BARCODES
REIHE 11 (48-52) CCD BARCODES
REIHE 12 (53) CCD BARCODES

RDX
PRGM REGS NEEDED: 20

REIHE 1 (1-4) CCD BARCODES

REIHE 2 (4-9) CCD BARCODES



REIHE 3 (10-14) CCD BARCODES



REIHE 4 (14-16) CCD BARCODES



REIHE 5 (16-26) CCD BARCODES



REIHE 6 (27-35) CCD BARCODES



REIHE 7 (35-37) CCD BARCODES



REIHE 8 (37-39) CCD BARCODES



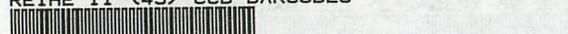
REIHE 9 (40-42) CCD BARCODES



REIHE 10 (43-44) CCD BARCODES



REIHE 11 (45) CCD BARCODES



CLRTN

PRGM REGS NEEDED: 4

REIHE 1 (1-4) CCD BARCODES



REIHE 2 (5-6) CCD BARCODES



RDE

PRGM REGS NEEDED: 33

REIHE 1 (1-5) CCD BARCODES



REIHE 2 (5-13) CCD BARCODES



REIHE 3 (13-21) CCD BARCODES



REIHE 4 (21-28) CCD BARCODES



REIHE 5 (29-31) CCD BARCODES



REIHE 6 (32-38) CCD BARCODES



REIHE 7 (38-40) CCD BARCODES



REIHE 8 (41-43) CCD BARCODES



REIHE 9 (43-45) CCD BARCODES



REIHE 10 (46-48) CCD BARCODES



REIHE 11 (48-51) CCD BARCODES



REIHE 12 (51-54) CCD BARCODES



REIHE 13 (54-57) CCD BARCODES



REIHE 14 (57-61) CCD BARCODES



REIHE 15 (62-66) CCD BARCODES



REIHE 16 (67-74) CCD BARCODES



REIHE 17 (74-79) CCD BARCODES



REIHE 18 (80) CCD BARCODES



WDE

PRGM REGS NEEDED: 15

REIHE 1 (1-4) CCD BARCODES



REIHE 2 (5-10) CCD BARCODES



REIHE 3 (10-17) CCD BARCODES



REIHE 4 (17-23) CCD BARCODES



REIHE 5 (24-29) CCD BARCODES



REIHE 6 (30-39) CCD BARCODES



REIHE 7 (39-46) CCD BARCODES



REIHE 8 (47-49) CCD BARCODES



SELVIEW

PRGM REGS NEEDED: 6

REIHE 1 (1-2) CCD BARCODES



REIHE 2 (2-8) CCD BARCODES



REIHE 3 (10-13) CCD BARCODES



PREAD PWRIT BUSY

PRGM REGS NEEDED: 17

REIHE 1 (1-3) CCD BARCODES



REIHE 2 (3-7) CCD BARCODES



REIHE 3 (8-15) CCD BARCODES



REIHE 4 (15-18) CCD BARCODES



REIHE 5 (19-28) CCD BARCODES



REIHE 6 (28-36) CCD BARCODES



REIHE 7 (36-40) CCD BARCODES



REIHE 8 (40-46) CCD BARCODES



REIHE 9 (46-47) CCD BARCODES



ERROR

PRGM REGS NEEDED: 16

REIHE 1 (1-4) CCD BARCODES



REIHE 2 (4) CCD BARCODES



REIHE 3 (5-7) CCD BARCODES



REIHE 4 (7-10) CCD BARCODES



REIHE 5 (10-13) CCD BARCODES



REIHE 6 (13-16) CCD BARCODES



REIHE 7 (16-18) CCD BARCODES



REIHE 8 (18-21) CCD BARCODES



REIHE 9 (22) CCD BARCODES



ELAID

PRGM REGS NEEDED: 5

REIHE 1 (1-3) CCD BARCODES



REIHE 2 (4-10) CCD BARCODES



REIHE 3 (11) CCD BARCODES



IRE?

PRGM REGS NEEDED: 5

REIHE 1 (1-5) CCD BARCODES



REIHE 2 (5-10) CCD BARCODES



REIHE 3 (11) CCD BARCODES



RESEL
PRGM REGS NEEDED: 4

REIHE 1 (1-3) CCD BARCODES

REIHE 2 (4-8) CCD BARCODES

DUMP
PRGM REGS NEEDED: 124

REIHE 1 (1-4) CCD BARCODES

REIHE 2 (5-11) CCD BARCODES

REIHE 3 (11-18) CCD BARCODES

REIHE 4 (18-22) CCD BARCODES

REIHE 5 (22-29) CCD BARCODES

REIHE 6 (30-34) CCD BARCODES

REIHE 7 (34-39) CCD BARCODES

REIHE 8 (39-45) CCD BARCODES

REIHE 9 (45-49) CCD BARCODES

REIHE 10 (49-50) CCD BARCODES

REIHE 11 (51-59) CCD BARCODES

REIHE 12 (59-66) CCD BARCODES

REIHE 13 (66-74) CCD BARCODES

REIHE 14 (74-81) CCD BARCODES

REIHE 15 (82-85) CCD BARCODES

REIHE 16 (85-92) CCD BARCODES

REIHE 17 (92-98) CCD BARCODES

REIHE 18 (98-104) CCD BARCODES

REIHE 19 (104-109) CCD BARCODES

REIHE 20 (109-112) CCD BARCODES

REIHE 21 (112-117) CCD BARCODES

REIHE 22 (118-122) CCD BARCODES

REIHE 23 (122-130) CCD BARCODES

REIHE 24 (131-134) CCD BARCODES

REIHE 25 (134-140) CCD BARCODES

REIHE 26 (141-144) CCD BARCODES

REIHE 27 (145-152) CCD BARCODES

REIHE 28 (152-159) CCD BARCODES

REIHE 29 (159-161) CCD BARCODES

REIHE 30 (162-172) CCD BARCODES

REIHE 31 (173-178) CCD BARCODES

REIHE 32 (178-180) CCD BARCODES

REIHE 33 (180-187) CCD BARCODES

REIHE 34 (188-196) CCD BARCODES

REIHE 35 (197-199) CCD BARCODES

REIHE 36 (199-204) CCD BARCODES

REIHE 37 (204-215) CCD BARCODES

REIHE 38 (216-225) CCD BARCODES

REIHE 39 (226-230) CCD BARCODES

REIHE 40 (231-236) CCD BARCODES

REIHE 41 (236-241) CCD BARCODES

REIHE 42 (242-252) CCD BARCODES

REIHE 43 (253-262) CCD BARCODES

REIHE 44 (263-270) CCD BARCODES

REIHE 45 (270-276) CCD BARCODES

REIHE 46 (277-282) CCD BARCODES

REIHE 47 (283-285) CCD BARCODES

REIHE 48 (285-289) CCD BARCODES

REIHE 49 (289-301) CCD BARCODES

REIHE 50 (302-309) CCD BARCODES

REIHE 51 (309-311) CCD BARCODES

REIHE 52 (311-313) CCD BARCODES

REIHE 53 (314-321) CCD BARCODES

REIHE 54 (322-328) CCD BARCODES

REIHE 55 (328-330) CCD BARCODES

REIHE 56 (330-335) CCD BARCODES

REIHE 57 (335-339) CCD BARCODES

REIHE 58 (339-346) CCD BARCODES

REIHE 59 (346-348) CCD BARCODES

REIHE 60 (348-355) CCD BARCODES

REIHE 61 (356-362) CCD BARCODES

REIHE 62 (362-366) CCD BARCODES

REIHE 63 (367-370) CCD BARCODES

REIHE 64 (370-374) CCD BARCODES

REIHE 65 (374-376) CCD BARCODES

REIHE 66 (376-382) CCD BARCODES

REIHE 67 (382-383) CCD BARCODES

Postvertriebsstück
Gebühr bezahlt

D 2856 E

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

Nr. 1 Januar 1984

Tastenfolge	Anzeige	Bemerkung	Tastenfolge	Anzeige	Bemerkung
g DEG	168 - 43 7				
g RTN	169 - 43 32				
f LBL 0	170 - 42, 21, 0	$v^2 = 1 + e^{12} \cos^2 B_m$	g RTN	198 - 40	
RCL 2	171 - 45 2			199 - 43 32	
Cos	172 - 24		f LBL 4	200 - 42, 21, 4	Umrechnung auf gon
g x ²	173 - 43 11		g Last x	201 - 43 36	
RCL 0	174 - 45 0		.	202 - 48	
*	175 - 20		9	203 - 9	
1	176 - 1		/	204 - 10	
+	177 - 40		g RTN	205 - 43 32	
g RTN	178 - 43 32				
f LBL 1	179 - 42, 21, 1	$1/M = \frac{v^3}{C}$			
GSB 0	180 - 32 0				
1	181 - 1				
.	182 - 48				
5x	183 - 5				
y ^x	184 - 14				
RCL 1	185 - 45 1				
/	186 - 10				
g RTN	187 - 43 32				
		$1/N = V/C$			
f LBL 2	188 - 42, 21, 2				
GSB 0	189 - 32 0				
V ^x	190 - 11				
RCL 1	191 - 45 1				
/	192 - 10				
g RTN	193 - 43 32				
f LBL 3	194 - 42, 21, 3				
g	195 - 43 26				
2	196 - 2				
*	197 - 20				

Peter Kiefer

Programm "Q"

Sie schickten mir das PRISMA-Heft vom Nov. 1983, wofür ich mich sehr bedanke. Ich fand darin einige interessante Sortierprogramme für den HP-41, darunter die Programme SHL und S-M mit einer sinnvollen Vorsortierung. Verbesserungen sind aber, auch ohne Vorsortierung, noch möglich. Die Laufzeit solcher langwierigen Programme wird entscheidend durch die Länge der am häufigsten durchlaufenen Schleife beeinflusst. Bei den abgedruckten Programmen haben diese Schleifen (z.B. Zeilen 14-2B bei BUL) 15 oder noch mehr Schritte.

Bei meinem beiliegenden Programm "Q" ist die entsprechende Schleife auf 9 Zeilen verkürzt (Z.19-27). Das Programm benötigt nur einen Hilfsspeicher R₀₀, die Wertereihe beginnt bei R₀₁. Nach einer kleinen Änderung, die diese Schleife garnicht und die Gesamtzeit nur minimal vergrößert, kann man die Reihe auch bei einem höheren Speicher beginnen lassen. Programm "Q" hat bei n = 20 die Laufzeiten 59-9 für den ungünstigsten und den günstigsten Fall. Im Mittel sind das 34 Sekunden, nicht mehr als beim Programm SHL mit Vorsortierung.

Bei meinem schnelleren Programm R ist dasselbe Verfahren mit einer Vorsortierung wie bei S-M kombiniert. Auch hier beginnt die Wertereihe bei R₀₁, und es wird wieder nur ein Hilfsspeicher R₀₀ benötigt, anstelle von 5 Speichern bei S-M. Die Laufzeiten für die extremen Fälle sind 30-24, gegenüber 36-22 bei S-M. Auch bei Zufallsreihen ist Programm R durchschnittlich etwa 10 % schneller als S-M.

Meine Vorschläge könnten bei Ihren Lesern und Ihren Freunden in Toulouse vielleicht Interesse finden.

```

01*LBL "Q"
02 "N?"
03 PROMPT
04 1 E3
05 /
06 ISG X
07 STO 00
08*LBL 03
09 ISG 00
10 GTO 04
11 STOP
12*LBL 04
13 RCL 00
14 INT
15 ENTER↑
16 DSE X
17 RCL IND Y
18*LBL 05
19 RCL IND Y
20 X<=Y?
21 GTO 06
22 STO IND T
23 RDN
24 DSE Z
25 DSE Y
26 GTO 05
27 STO 01
    
```

```

20 RCL Y
21 -
22 RCL IND X
23 RCL IND 00
24 X>Y?
25 GTO 02
26 STO IND Z
27 X<>Y
28 STO IND 00
29*LBL 02
30 R↑
31 ISG 00
32 GTO 01
33 RCL 00
34 FRC
35 ISG X
36 STO 00
37 X<>Y
38 GTO 00
39*LBL 03
40 ISG 00
41 GTO 04
42 STOP
43*LBL 04
44 RCL 00
    
```

Herbert Gudehus
Strandweg 53
2000 Hamburg 55