

485X

FAHRPLAN

auf dem HP 48 SX
von Günter Schlapka

Autoren: Günter Schlapka, Michael...
Das Programm FAHRPLAN ist ein...
Es ermöglicht die Planung von...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...



485X

Sortierung

von Listen und Menüs
von Michael Rittler

Das Programm SORT sortiert...
Es ermöglicht die Sortierung von...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...

485X

Listing von RUM-PRO

entgeltliche Version

Das Programm RUM-PRO...
Es ermöglicht die...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...

485X

Listing von VADR-A-1

Das Programm VADR-A-1...
Es ermöglicht die...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...

485X

Graphisto

Der Teufel steckt im Detail
Grafikobjekte auf dem HP48

Das Programm GRAPHISTO...
Es ermöglicht die...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...

41CX

HP41 Utilities

von Dr. Martin Hochenegger

Byles ist die CCD Initialis...
Das Programm Byles...
Es ermöglicht die...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...



Maschinensprache Teil II

von Georg Hoppes

Das Programm MASCHINENS...
Es ermöglicht die...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...

Objekte su

auf dem HP48

Das Programm OBJEKTE SU...
Es ermöglicht die...
Die Eingabe erfolgt über...
Die Ausgabe erfolgt über...
Das Programm ist in...
Die Installation erfolgt...
Die Dokumentation ist...
Die Kontaktadresse ist...
Die Preisangabe ist...
Die Bestelladresse ist...

Magazin

Auf der Suche nach der richtigen Mailbox

Nachruf

41er
Eingespannter Einfeldrahmen - Trapez o. Rechteck
HP41 Utilities

48SX

Fragestunde
Pas de deux - Teil VII

Graphisto

Grafikobjekte
RS232-Schnittstellenpaket
Statik auf dem HP48SX

Maschine & Sprache - Teil II

Sortierung von Listen und Menüs
RS232-HPIL-Interface

BlackJack

Objekte Suchen
Fahrplan
Osterdatum

71er

Osterdatum
ALMCCD1-Korrekturen

95er

Connectivity Pack

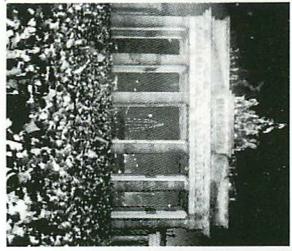
ATARI

Disketten '91
MS-DOS

Universelle ROM-Karte

- Paßwortkarte

Frieden zwischen Ost und West. Hunger in der Dritten Welt.



Ost und West wachsen zusammen. Doch über die Freude sollten wir die Dritte Welt nicht vergessen. Dort sind Millionen von Menschen nach wie vor vom Hungertod bedroht. Sie brauchen unsere Hilfe zur Selbsthilfe. Damit die Welt zusammenwächst. Ohne Hunger.



DEUTSCHE WELTHUNGERHILFE
Spendenkonto Sparkasse Bonn: 1 1 1

Deutsche Welthungerhilfe · Adenauerallee 134 · 5300 Bonn 1 · Tel.: 02 28 / 22 88 0



W&W Software Products GmbH

Bei uns gibt's keine Probleme, nur **Lösungen!**



COMPUTER
PRODUKTE
PARTNER-
LÖSUNGEN

HP-95LX + DERIVE = MATHE IN 3D!!

Die vielfältigen grafischen Darstellungsmöglichkeiten mathematischer Formeln und Berechnungen sind nur ein Teil dieses wirklich umfassenden Programms, das natürlich auch auf jedem PC lauffähig ist. Sein Berechnungspotential umfaßt praktisch 2000 Jahre Mathematik!

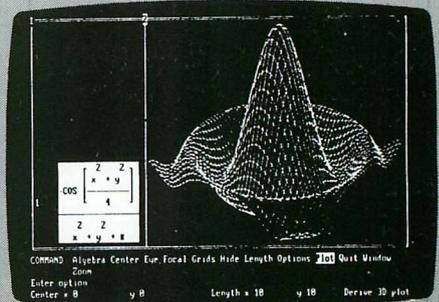
HP-95LX + DERIVE + 0,5 MB
DM 2.849,-

HP-95LX + DERIVE + 1 MB
DM 3.449,-

Bei Bestellungen vor dem **15.11.1991** wird DERIVE als Diskettensoftware auf RAM-Karten geliefert. Das ROM erhalten Sie als Zugabe **kostenlos!**

*Spätere Besteller erhalten DERIVE als ROM.

DERIVE bietet übersichtliche Hilfebildschirme, die Druckausgabe erfolgt direkt oder über eine Datei. Cursor-gesteuert ist die Auswahl von Teilausdrucken ebenso möglich wie das Editieren bereits existierender Ausdrücke. Die Algebra- und Graphikfenster sind über- und nebeneinander legbar. Durch das Interface zu DOS sind die Einsatzmöglichkeiten von DERIVE nahezu unbegrenzt!!!



HOTLINE: 02202/42021

W&W Software Products GmbH, Odenthaler Str. 214, 5060 Bergisch Gladbach 2, FAX: 32794

Inhalt

Magazin	
Editorial	3
Nachruf Gerhard Link	4
Auf der Suche nach der richtigen Mailbox	5
41er	
Eingespannter Einfeldrahmen Trapez- oder Rechteckform	63
HP41 Utilities	80
48SX	
Fragestunde zum 28S u. 48SX	21
Pas de deux - Teil VII	23
Graphisto Grafikobjekte auf dem 48SX	32
RS232-Schnittstellenpaket	37
Statik auf dem HP48SX	48
Maschine & Sprache - Maschinensprache - Teil II	53
Sortierung von Listen und Menüs	62
RS232-HPIL Interface Test	73
Umwandlungsprogramm TIFF --> GROB	76
BlackJack	77
Objekte suchen	82
Fahrplan	83
Osterdatum	85
Alternatives Schnittstellenpaket	95
71er	
Osterdatum	85
Alarmsystem für den HP71B ALMCCD1 - Korrekturen	86
95er	
Connectivity Pack	19
ATARI	
Disketten des Jahrs 1991	22
MS-DOS	
Universelle ROM-Karte am Beispiel Paßwortkarte	11
Clubbörse	96
Serviceleistungen	98
Inhalt	33
Impressum	97

Kein gutes Jahresende 1991

Das Jahr 1991 hat uns kein gutes Ende bereitet. Wir trauern um unseren ersten Vorsitzenden Gerhard Link, der am 22. Dezember zusammen mit seiner Frau und sechszwanzig anderen Menschen bei einem Flugzeugabsturz in der Nähe von Heidelberg tödlich verunglückte. Ein Sprecher auf der gemeinsamen Trauerfeier in Rüsselsheim drückte aus, was kaum treffender gesagt werden kann: "Der Tod ist aus seiner Anonymität herausgetreten ... und macht uns betroffen und sprachlos zugleich". Vierzehn Tage vor dem katastrophalen Unglück konnten wir Gerhard Link noch munter und zufrieden bei unserer CCD Mitgliederversammlung in Frankfurt am Main erleben. Trotzdem müssen wir uns mit dieser schmerzlichen Tatsache abfinden.

Mobile Datenverarbeitung

Seit Erscheinen des HP 95LX Palmtop PCs scheint die ewige Kluft zwischen Taschencomputer und PC-Anwendungen überwunden zu sein. Obwohl der 95er mit seinen eingebauten Programmen (Texteditor, Terminkalender, Adressbuch, Finanztaschenrechner, Lotus 1-2-3 Version 2.2, Datenkommunikationsprogramm und Dateimanager) eindeutig für den kaufmännischen und organisatorischen Bereich bestimmt ist, deckt er hervorragend einen Teil der Anwendungen ab, die viele Clubmitglieder schon zu Zeiten des technisch-wissenschaftlichen HP 71B portabel in der Jackentasche mit sich führten. Ich denke da besonders an die Adreßdatenverwaltung und die Textverarbeitung, mit der man sich auf dem einzeiligen Display abmühte. Der HP 95LX ist kleiner und hat für seine Winzigkeit ein enormes Display von 40 Spalten und 16 Zeilen. Als allgegenwärtiges elektronisches Notizbuch und kaufmännisches "Analyse-Instrument" mit voller Datenaustauschkapazität zum PC möchte ich den 95er nicht mehr missen.

Zur Programmierung von mathematischen und technisch-wissenschaftlichen Anwendungen eignet sich der HP 95LX allerdings nicht direkt. Da hilft auch der im Finanztaschenrechner integrierte sehr gute Gleichungslöser nur wenig. Der erfolgreiche Versuch eines Clubmitgliedes zeigt aber einen anderen interessanten Weg auf. Da der 95er PC-kompatibel ist, laufen auf ihm auch einfache DOS-Programme und Programmiersprachen wie z.B. Turbo Pascal 3. Leider ist der knappe RAM-Vorrat des 95er dann aber schnell erschöpft.

100 Seiten

Mit diesem PRISMA haben wir ein - im wahrsten Sinne des Wortes - "dickes Ende" für das CCD-Jahr '91 geschafft haben. Mit 100 Seiten ist es das umfangreichste Heft aller Zeiten, womit wir im Jahresschnitt betrachtet und trotz der langen "Sendepause" eine gute Leistung erbracht haben. Und das, obwohl der gesamte redaktionelle Aufwand einschließlich der Vorbereitung für die Drucklegung zur Zeit von durchschnittlich nur zweieinhalb Mitgliedern geleistet wird. Wir arbeiten daran, diese Situation zu verbessern.

Happy Programming
und ein gutes Jahr 1992
wünscht
Alf-Norman Tietze

Gerhard Link tödlich verunglückt

Rechtsanwalt Dipl.-Volkswirt Gerhard Link, seit April 1990 der 1. Vorsitzende des CCD, und seine Frau Helga Zagovec-Link sind am 22. Dezember 1991 bei einem Flugzeugabsturz tödlich verunglückt.

Gerhard Link trat 1986 dem CCD als Mitglied bei. Auf der Jahreshauptversammlung des CCD im April 1987 wurde er für drei Jahre als Kassenprüfer neben dem weiter amtierenden Achim Gmein gewählt. Als im Frühjahr 1990 Wolfgang Fritz einen Nachfolger für das Amt des 1. Vorsitzenden suchte, fand sich Gerhard Link dazu bereit, obwohl er gerade im Aufbau einer eigenen Kanzlei war. Auf der Jahreshauptversammlung am 21.4.1991 wurde er zum 1. Vorsitzenden des CCD gewählt und füllte dieses Amt mit seiner ganzen Person und Kompetenz aus. Vor allem großes Interesse hatte Gerhard Link für die Möglichkeiten der Telekommunikation, den Zugang zu Datenbanken wie Juris und zu Schwarzen Brettern juristischen Inhalts gezeigt.

Gerhard Link hatte Verbindungen mit Filmkreisen in seiner Heimatstadt Rüsselsheim und Künstlern der Hochschule für Gestaltung Offenbach. Als er und seine Frau kurz vor Weihnachten mit einem Rüsselsheimer Filmteam des Filmemachers Martin Kirchberger zu Aufnahmезwecken für einen Satirefilm an Bord der DC3 "Dakota", einem Oldtimer von 1942, waren, nahm das Verhängnis seinen Lauf. In der Nähe von Heidelberg geriet die niedrig fliegende Maschine bei schwierigen Sichtverhältnissen in die Baumwipfel des 496 m hohen Heiligenbergs und riß bei Ihrem Absturz Gerhard Link und seine Frau und weitere 26 Menschen in den Tod, nur 4 Personen überlebten. Ergebnisse der Ursachen-Ermittlung des Luftfahrtbundesamtes in Braunschweig werden erst für März erwartet.

An der Trauerfeier der Stadt Rüsselsheim für die Opfer und an der Bestattung des Ehepaars Link nahmen der 2. Vorsitzende Alf-Norman Tietze und der Schatzmeister Dieter Wolf teil.

Der CCD wird Gerhard Link ehrend gedenken.

Auf der Suche nach der richtigen Mailbox

Der CCD muß sich entscheiden zwischen Geo5, GeoD und Alternativen

von Wolfgang Fritz (CCD 125)

Zusammenfassung

A. Was ist E-Mailing

B. GeoD

1. Verbesserungen

2. Kostenerhöhungen

a) Alte Kostenstruktur

b) Neue Kostenstruktur für GeoD

C. CCD-Entscheidung: Zunächst Verbleiben in Geo5

D. Jetzige Lösung nur vorläufig?

1. Isolierung

2. Kontinuierliche Abnahme der E-Mail-Nutzung

3. Existenz leistungsfähiger Alternativen

E. Anforderungen an die Box

F. Unterstützung durch den CCD

Literatur

Zusammenfassung

Neben Telefon und Telefax sollte die E-Mail-Nutzung für jeden Geschäftsmann und Freiberufler selbstverständlich sein und bietet dem Hobbyisten interessante Informationszugänge. Der CCD hat die Bedeutung des elektronischen Nachrichtenversendens schon früh erkannt und ist seit Ende 1984 als Gruppe in einer der GeoNet-E-Mail-Systeme (Geo1) von Günther Leue, einem bekannten Vorreiter in Deutschland auf diesem Gebiet.

Gekoppelt mit Verbesserungen des E-Mailing (E-Mail-System: GeoD) wollte GeoNet im Herbst 1991 eine neue Kostenstruktur mit allerdings wesentlichen Verteuerungen für die Mailbox-

Teilnehmer einführen. Die Mindestgebühr der einzelnen Teilnehmer sollte von DM 10 auf DM 28 monatlich steigen, die Nutzungsgebühren würden sich praktisch mehr als verdreifachen. Der Mehrheit der CCD-Mailboxteilnehmer war die Kostenerhöhung zu hoch. Der CCD-Vorstand verhandelte und entschied sich für ein Verbleiben in der GeoNet-Mailbox zu den alten Bedingungen, damit aber auch auf den Verzicht der Verbesserungen von GeoD.

Das Verbleiben in GeoNet (jetzt: Geo5) ist womöglich nur eine vorläufige Lösung aus folgenden Gründen: Die CCD-Gruppe ist jetzt innerhalb der GeoNet-Systeme stärker als vorher isoliert von anderen Teilnehmern und auch von der technischen Entwicklung. Zweitens haben wir festgestellt, daß die Nutzung der GeoNet-Box im Laufe der Zeit immer mehr abgenommen hat, zunächst langsam, im letzten halben Jahr immer stärker. Damit ist ein alter Beschluß, daß der Club die monatliche Grundgebühr der Box übernimmt, auf der Mitgliederversammlung neu zu überdenken; werden die Kosten nicht mehr vom Club weitergetragen, bedeutet das für die z.Z. 33 Box-Teilnehmer eine monatliche Kostenerhöhung von 7 DM. Zum Dritten aber ist das Verbleiben in GeoNet vielleicht deswegen nur vorläufig, weil es heute leistungsfähige Alternativen zu GeoNet gibt.

So hat der Vorstand vor allem die Box unseres tüchtigen Clubfreundes André Koppel aus Berlin im Visier. Die Leistungsfähigkeit dieser Box kann entsprechend den Wünschen des CCD ausgebaut werden; die Teilnehmerkosten hängen natürlich von der Fülle der Wünsche ab, sie sind jedoch

wahrscheinlich deutlich geringer als die der GeoNet.

Andererseits legen einige der bisherigen CCD-Box-Teilnehmer Wert auf den Zugang zu Datenbanken und zu Schwarzen Brettern mit Spezialinformationen; wenn diese Angebote in einer Alternativ-Box nicht vorliegen, würden diese Mitglieder lieber in GeoNet verbleiben, womöglich gar in das (teurere) GeoD wechseln.

Der CCD muß sich entscheiden. Neben der wahrscheinlich in Zukunft entfallenden finanziellen Unterstützung durch die Übernahme der Grundgebühr kann der CCD weiterhin organisatorische Unterstützung geben. Zum Beispiel durch Verhandlungsführung, Rechnungsstellung u.ä. Unterstützung für GeoNet, André Koppels Berliner Box, für beides, oder für ganz etwas anderes? Der CCD muß sich entscheiden.

A. Was ist E-Mailing

Der CCD betreibt seit 1984 E-Mail, sprich "Ihnmehl". Das ist die elektronische Post, die aus Amerika kommt. Überall, wo am Arbeitsplatz ein Computer-Bildschirm steht, zu dem außer dem Stromkabel noch Datenleitungen führen, und wenn's auch nur das Telefon ist - da kann der Spruch aufblitzen: "Nachricht wartet". Klassische Großrechner-Terminals als "Datenendstationen" oder netzverbundene Personalcomputer stehen längst nicht mehr allein. Sogenannte lokale Netze (local area network, LAN) verbinden die PCs in einem Büro oder in einem Unternehmen, Weitbereichsnetze (wide area network, WAN) oder gar innerstädtische "Datenautobahnen" (metropolitan area network, MAN)

übertragen Daten breitbandig über Stadtteile hinweg, herkömmliche Datenpaketnetze wie das deutsche Datex-P-Netz, Sondernetze wie Bildschirmtext (Btx) oder einfach Punkt-zu-Punkt-Modemverbindungen lassen einen Computer mit anderen in Verbindung treten, von Büro zu Büro oder von Erdteil zu Erdteil. Was liegt da näher, als auch die Menschen hinter den Bildschirmen auf diesem Weg sich Notizen und Briefe schreiben zu lassen, dienstlich und privat. Das ist Electronic Mail, die elektronische Post.

In Deutschland gibt es kommerzielle Betreiber z.B. die private Dortmunder GeoNet und die postalische Telebox mit zusammen knapp 6000 Teilnehmern. Dann gibt es Btx mit fast 300.000 Teilnehmern und ungeahnt viele private Hobby-Boxen, etwa im Fido- oder Zerberus-Netz, die alle einem Kreis mehr oder weniger zahlender Teilnehmer dienen. Besonders bei amerikanischen Konzernen gibt es Electronic-Mail-Systeme, die einen Großteil unternehmensinterner Post erledigen und meist Übergänge zu öffentlichen Systemen haben. Hier hat jeder Mitarbeiter seine "Mailbox" am Arbeitsplatz, den elektronischen Postkasten. Man ist gehalten, ihn regelmäßig zu leeren, auf Besuch in anderen Niederlassungen oder gelegentlich modem-telefonisch auch von zu Hause aus. Selbst im Urlaub in Südtirol ist dann der deutsche Postkasten nur ein Telefongespräch nach Bozen weit weg.

Elektronische Nachrichten werden dabei wie herkömmliche Post behandelt, nur schneller und, per Protokoll, Bit für Bit unverfälscht. Die Computer sorgen dafür, daß alle Nachrichten vom Empfänger sofort als Daten weiterbearbeitet, also abgespeichert, mit Kommentar weiterversandt, ausgedruckt oder restlos umweltfreundlich gelöscht werden können.

Adressen werden elektronisch nachgeschlagen. Der Versand erfolgt mit Lichtgeschwindigkeit, man kann "eingeschrieben" versenden, "mit Rückschein" ist dann immer dabei. Dadurch weiß man, daß der Empfänger die Nachricht auch bekommen hat, selbst wenn er viele Zeitzonen weit weg ist. Automatisches Weiterleiten ist kein Problem.

Mailbox steht nicht im Wettbewerb zum Telefax (ca 870.000 Teilnehmer). Viele Box-Teilnehmer nutzen selbstverständlich ständig Telefax. Die Vorteile des Mailbox-Systems sind dennoch:

1. Zugang zu Telex (und damit Teletext), sehr preiswert und durch automatische Wahlwiederholung und Vorgabe eines Verteilers sehr komfortabel. Telex ist noch wichtig im Geschäftsverkehr mit afrikanischen Ländern und Ostländern, da die Telefon- und Faxverbindungen oft gestört sind. Im Störfall wiederholt das Mailbox-Telex die Nachricht automatisch, bis diese einwandfrei angekommen ist.

2. Zugang zu Datenbanken (wie Juris, Literatur, etc.)

3. Austausch von Dateien mit Geschäftspartnern zum Beispiel eilige Ausschreibungstexte, die dann beim Partner gleich am Rechner bearbeitet werden können. Leider fehlen dazu allerdings häufig die Geschäftspartner, die den Anschluß haben.

Der CCD hat die Bedeutung des elektronischen Nachrichtenversand schon früh erkannt und ist seit Ende 1984 als Gruppe in einer der GeoNet-E-Mail-Systeme (Geo1) von Günther Leue, dem oben erwähnten Vorreiter in Deutschland auf diesem Gebiet. Über 60 CCD-Mitglieder waren in der Vergangenheit Box-Teilnehmer in GeoNet.

B. GeoD

Im Herbst 1991 hat GeoNet umgestellt, "auf modernere Technik", heißt es in dem entsprechenden Rundschreiben. Das folgende ist, leicht gekürzt, dem Rundschreiben entnommen:

1. Verbesserungen

Wir möchten Sie auf drei wichtige Veränderungen hinweisen. GeoNet ermöglicht nun auch allen Teilnehmern der Geo1, Geo3 und Geo5 die Überführung auf E-Mail-Systeme der GeoII-Generation (GeoD-System). Diese neue Generation von E-Mail-Systemen basiert auf Hardware der Fa. Digital Equipment, bewahrt alle Vorteile der Geo-Technik, mit denen sich GeoNet weltweit einen Namen verschafft hat und bietet zusätzlich wesentlich bessere und umfassendere Leistungsmerkmale.

Erst die Nutzung modernster Technik und die Weiterentwicklung unserer E-Mail-Software hat es möglich gemacht, daß nun ein zentrales GeoNet E-Mail-System für alle GeoNet-Kunden zur Verfügung gestellt werden kann. Die Teilnehmer der Systeme Geo1, Geo3, Geo5 sowie die Teilnehmer der GeoNet-Tochtergesellschaft MBK MediaBox Köln GmbH werden alle auf ein System übernommen. Diese "Umstellung" soll bis spätestens 01.10.1991 abgeschlossen sein. Ein wesentlicher Vorteil dieses zentralen E-Mail-Systems ist, daß nun alle GeoNet-Teilnehmer lokal miteinander kommunizieren können, so daß InterMail-Kosten (für die Kommunikation zwischen unterschiedlichen GeoNet-Systemen) entfallen. Ab dem 1. Oktober sind tausende E-Mail-Nutzer lokal zu erreichen. Ein weiterer Vorteil besteht darin, daß auch die Zugriffe zu allen in Dortmund beheimateten Gateways -- ohne InterMail-Kosten abgewickelt werden können.

Mit der Inbetriebnahme des neuen Systems entfällt die bisher erhobene monatliche Gebühr für die Zurverfügungstellung einer X.400-Adresse in Höhe von DM 20,-. Vielmehr erhält jetzt jeder GeoD-Teilnehmer automatisch und ohne zusätzliche Grundkosten eine individuelle X.400-Adresse zugeteilt. Jedem Benutzer steht damit die Kommunikation mit mittlerweile schon vielen Millionen Teilnehmern in fremden Systemen -- privaten und öffentlichen -- in aller Welt offen. GeoNet nimmt diesem die Zukunft bestimmenden Standard die Komplexität. Sie verkehren mit fremden X.400-Teilnehmern genau so einfach, wie mit Teilnehmern im eigenen oder in Systemen des GeoMail-Verbunds.

2. Kostenerhöhungen

In dem Rundschreiben von GeoNet heißt es weiter:

In diesem Zusammenhang hat GeoNet -- nach reiflichen Überlegungen -- eine neue Entgeltstruktur verabschiedet. Die Gründe für eine Änderung der Gebührenstruktur sind überwiegend (von den allgemeinen Preissteigerungsraten sind wir natürlich auch betroffen) in der technischen Entwicklung zu suchen. Als Berechnungsbasis wurde bislang immer die Verbindungszeit mit dem E-Mail-System verwendet. Diese Berechnung wird je-

doch nicht mehr den Benutzeransprüchen gerecht.

Die neuen Entgelte für die Nutzung von GeoNet E-Mail-Systemen orientieren sich nicht mehr an den Verbindungsminuten, sondern an dem übertragenen Datenvolumen. Dies bedeutet: keine Grundgebühren, keine Berechnung der Verbindungsminuten. Selbstverständlich sind "on-line" abgerufene Hilfstexte von der Berechnung der Datenvolumen ausgeschlossen -- also kostenlos.

Soweit aus dem Rundschreiben.

len Einzelteilnehmer: Jeder CCD-Teilnehmer zahlte eine Box-Verbindungs-Gebühr von DM 0,15 pro Minute, monatlich aber mindestens DM 10 + MWSt, so daß pro Monat etwa 70 Minuten Verbindungsgebühr frei waren. (Hier nur am Rande erwähnt sind die Kosten des Teilnehmers an die Telekom für den Zugang zum Datex-P-Netz in Höhe von DM 15 pro Monat und die Datex-P-Verbindungsgebühren von grob DM 0,2 bis 0,28 pro Minute.) Darüber hinaus konnte man sich für DM 20 im Monat eine X.400-Adresse (s.o.) oder für ebenfalls DM

liches Mindestvolumen von - für den CCD - 140.000 Zeichen (= DM 28,-) pro Teilnehmer (gegenseitig für die ganze Gruppe verrechenbar, so daß ein Mehrvolumen eines Teilnehmers auf ein Mindervolumen eines anderen Teilnehmers angerechnet wird).

Selbst wenn man berücksichtigt, daß das Grundentgelt von 235 DM wegfällt, das der CCD pauschal zahlt, so bleiben doch noch recht erhebliche Verteuerungen für den Club als Ganzes und auch für jeden einzelnen Teilnehmer übrig. (Dazu kommt noch, daß wir bei der gegenseitigen Verrechnung des Aufkommens der einzelnen Teilnehmer untereinander keinen halbwegs gerechten und überschaubaren Weg gefunden haben. Egal wie wir es machen, wird es immer eine Gruppe von Teilnehmern geben, die sich zu recht beklagen, daß sie benachteiligt seien.)

Wenn ich zunächst die Grund- und Mindestentgelte außer acht lasse und nur den reinen "Arbeitspreis" betrachte, dann muß ein Teilnehmer, der mit einem langsamen Modem (300 bit pro Sekunde) zögerlich arbeitet, mit einer knappen Verdoppelung des Entgelts rechnen, bei flottem Arbeiten mit einem schnellen Modem kann im Extremfall der zehnfache Betrag herauskommen.

Die realen Verhältnisse sehen allerdings zur Zeit so aus, daß im Mittel das Mindest-Aufkommen von 140 KByte pro Person bei weitem nicht erreicht wird. Die abnehmende Tendenz der Nutzung der Mailbox drückt sich darin aus, daß von den 33 Teilnehmern 15 Teilnehmer in mindestens 3 Monaten sich überhaupt nicht in die Box eingewählt haben. Nur drei Teilnehmer hätten (auf das ganze Halbjahr berechnet) die neue Mindestgebühr von 6 x 28,- DM überschritten. Selbst wenn man die "Karteileichen" abzieht, liegen wir immer noch deutlich darunter. Und das dürfte sich bei der neuen Struktur sicher nicht verbessern, denn da überlegt man sich zweimal, was man runterlädt. Es dürften daher auf absehbare Zeit die Mindestentgelte das Entscheidende sein. Da stehen den 10 DM (Einzelner) plus ca. 7 DM (Club) alt die 28 DM neu gegenüber, also das über 1,6-fache.

Zu den gesamten Kommunikationskosten muß man auch die Telefon- und

Kostenart	Geo1 und Geo5	GeoD
Für den CCD: monatl. Grundgebühr	235,00	0,00
Für den Teilnehmer: monatl. Mindestgeb.	10,00	28,00
Minuten-Gebühr	0,15	0,00
Volumen-Gebühr pro angefangene 1000 Zeichen	0,00	0,20
Persönliche Telex- nummer	20,00	20,00
Persönliche X.400- Adresse	20,00	0,00
Sonderdienste wie: Datenbankzugriffe InterMail InterSwitch Telex-Versand Telefax-Versand	je nach Anfall gemäß den on-line-gespeicherten Tabellen	

Tabelle 1: Kostenvergleich.

Der CCD war von 1984 bis September 1991 in der Box Geo1, er ist ab Oktober 1991 zu den alten Bedingungen in der Geo5. Demgegenüber gestellt sind die Kosten für das modernere GeoNet-II-System mit der GeoD-Box.

a) Alte Kostenstruktur

Wie sahen bisher die Kosten aus: GeoNet verlangte vom CCD eine monatliche Grundgebühr von DM 235, die aufgrund eines Mitgliederbeschlusses der Club trug. Der Club streckte zudem gegenüber GeoNet die Teilnehmerkosten vor und stellte sie anschließend den Mitgliedern in Rechnung.

Für diese Eigenleistungen des Clubs erhielten seine Mitglieder in GeoNet günstigere Konditionen als die norma-

20 pro Monat eine persönliche Telexnummer in Luxemburg oder in den USA besorgen. Für den Telex-Versand, Fax-Versand und vor allem für Datenbank-Zugriffe (z.B. JURIS, Patentdatenbank, DIMDI, GENIOS) mußte extra bezahlt werden.

b) Neue Kostenstruktur für GeoD

Als neue Berechnungsgrundlagen (s. Tabelle 1) dienen nun "angefangene Einheiten von 1.000 übertragenen Zeichen" - die jeweils mit DM 0,20 berechnet werden -- sowie ein monat-

Datex-P-Gebühren für den Mailbox-Zugang zählen. Da das GeoNet-II-System wesentlich schneller reagieren soll als das bisherige, werden diese Kosten merklich sinken. Allerdings sind realistischerweise kaum mehr als 30 % Einsparung der Zugangskosten zu erwarten, also ein noch kleinerer Anteil der Gesamtkosten, der die Erhöhung der Mailboxkosten bei weitem nicht aufwiegt.

C. CCD-Entscheidung: Zunächst Verbleiben in Geo5

Aus verschiedenen Gründen konnte und wollte der CCD-Vorstand im September 1991 noch keine endgültige Entscheidung zur Zukunft der Mailbox-Kommunikation treffen.

Auf der einen Seite haben sich viele Mitglieder dahin ausgesprochen, daß sie auf keinen Fall eine Gebührenerhöhung mitmachen würden. Und etliche sagten, daß die Geo-Box deutlich an Anziehungskraft abgenommen habe und sie ohnehin an Aussteigen dachten.

Auf der anderen Seite gibt es einige Mitglieder, die nicht ohne weiteres auf die Dienste von GeoNet verzichten können, speziell was Telex und Datenbanken betrifft.

Weiter hat sich keiner ausgesprochen, daß er von den Neuerungen von GeoNet II profitieren würde.

Daher haben wir uns entschlossen, von der von GeoNet angebotenen Möglichkeit Gebrauch zu machen, auf einem alten System zu den alten Bedingungen weiterhin teilnehmen zu können.

Leider war ein völlig unveränderter Betrieb nicht möglich, wir mußten auf Geo5 umziehen. Die neue NUA lautet 45 6673 32945, die neue Modemwahlnummer lautet (0 66 73) 15 85.

D. Jetzige Lösung nur vorläufig?

Das Verbleiben in der Geo5 ist wahrscheinlich nur eine Übergangslösung. Zumindest sprechen die drei folgenden Gründe dafür.

1. Isolierung

In der Geo5 sind die CCD-Teilnehmer ziemlich allein. So schreibt ein CCD-Mitglied: "Mein Interesse galt eigentlich immer der allgemeinen Kommuni-

kation mit Gleichgesinnten und auch mal dem Lösen von Problemen über eine Anfrage bei den Kollegen." Das Beschicken der "Schwarzen Bretter" mit Nachrichten an ist rapide zurückgegangen, seitdem der Großteil der GeoNet-Nutzer in GeoD gewechselt hat. Auch in der technischen Entwicklung werden wir abgehängt sein; während wir früher schon mal gerne als "Versuchskaninchen" für Neuerungen vor deren offiziellen Einführung dienen konnten, wird das in der "alten" Geo5 nicht mehr sein.

2. Kontinuierliche Abnahme der E-Mail-Nutzung

Völlig unabhängig von den eben erwähnten geplanten Änderungen haben wir anhand der abzuführenden Entgelte festgestellt, daß die Nutzung der GeoNet-Box im Laufe der Zeit immer mehr abgenommen hat, zunächst langsam, im letzten halben Jahr immer stärker. Wenn das einen einzelnen Teilnehmer beträfe, dann bräuchten wir uns darüber nicht viel Gedanken machen. Es könnte viele recht persönliche Gründe haben. Offensichtlich ist es aber so, daß dieser Trend bei allen CCD-Teilnehmern gleichermaßen festzustellen ist, beim einen vielleicht etwas mehr, beim anderen etwas weniger. Weiterhin scheint dieser Trend sich nicht nur beim CCD zu zeigen, auch bei anderen Teilnehmern der Box stellen wir weniger Aktivität fest, vor allem die Bretter wurden, auch schon vor dem Wechsel im Herbst 1991, wesentlich schlechter beschickt als früher.

Da der Kommunikationsbedarf insgesamt sicherlich nicht gesunken ist, muß er sich auf andere Medien verlagert haben. Die Geo-Box muß an Anziehungskraft verloren haben und andere Medien und Systeme müssen zugelegt haben. Von mehreren Mitgliedern habe ich in der letzten Zeit offen gehört, daß ihnen die Geo-Box "nichts mehr bringt". Für etliche Dienste von GeoNet besteht bei vielen Teilnehmern anscheinend nur geringer oder gar kein Bedarf, während andere Anforderungen und Bedürfnisse vieler Mitglieder nur unzureichend von Geo befriedigt werden.

Andererseits sagten andere CCD-Mitglieder, daß sie Dienste von GeoNet nutzen, die für sie wichtig sind und die sie auf anderem Weg gar nicht oder

nur zu wesentlich schlechteren Bedingungen erhalten könnten. Hier sind vor allem Spezialbretter (mit den Pressemitteilungen der obersten Gerichtshöfe des Bundes (BGH, BFH, BSozG, BAG, BVerwG) und des Bundesverfassungsgerichtes zu nennen, was wohl nur im GeoNet geboten wird. Und zu nennen ist die von GeoNet gebotene Möglichkeit, auf Datenbanken (s. Tabelle 2) zugreifen zu können, es sei denn, man meldet sich bei den einzelnen Datenbasisbetreibern an, aber das ist für Wenignutzer zu teuer. Für solche Teilnehmer bleibt das Fazit: Wenn die denkbaren anderen Mailboxen den Zugriff auf Datenbanken und Fachbretter nicht bieten können, würden sie in GeoNet verbleiben, auch zu der höheren Mindestgebühr von 28,- DM / Monat.

3. Existenz leistungsfähiger Alternativen

Zum Dritten aber ist das Verbleiben in GeoNet vielleicht deswegen nur vorläufig, weil es heute leistungsfähige Alternativen zu GeoNet gibt. Aus diesem Grunde haben wir uns nach Alternativen zu GeoNet umgesehen und sind auch fündig geworden. So hat der Vorstand vor allem die Box unseres tüchtigen Clubfreundes André Koppel aus Berlin im Visier. Die Leistungsfähigkeit dieser Box kann entsprechend den Wünschen des CCD ausgebaut werden; die Teilnehmerkosten hängen natürlich von der Fülle der Wünsche ab, sie sind jedoch wahrscheinlich deutlich geringer als die der GeoNet.

E. Anforderungen an die Box

Es gilt zu klären, welche Anforderungen an eine Alternative gestellt werden sollen. Da anscheinend recht unterschiedliche Interessenlagen im Club herrschen und die bisher bekannt gewordenen Meinungen sicher nicht repräsentativ sind, kann und will der Vorstand nicht von oben herab eine Entscheidung treffen. Es sollen vielmehr alle Mailbox-Teilnehmer mitteilen, wie ihre Meinungen und Interessen aussehen.

Welche Dienste erwünscht? Vor allem wollen wir wissen, welche der verfügbaren Dienste von GeoNet von wem häufig/gelegentlich/nie benutzt werden und wie wichtig diese Dienste für den einzelnen sind. Und umgekehrt

*** Das II - DATENBANKANGEBOT auf einen Blick ****

Host (Anbieter und Sitz)	Datenbank(en) weitere Unterteilung
Echo, Luxembourg	Dianeguide
ESA/IRS, Italien	Inspec Energyline Pascal Abi/Inform Biosis CAB Compendex
Datastar, Schweiz	Abi/inform Business Compendex FT Company Information Management Contents MGMT
Dimdi, Deutschland	Biosis Canc CAB FSTA Meditec Medline
Fiz-Technik, Deutschland	Zentraldokumentation Elektrotechnik ABC der deutschen Wirtschaft
Genios, Deutschland	Handelsblatt Wirtschaftswoche
Infoline, Grossbritannien	Chemical Business News MMA, Management and Marketing Abstracts KBE, Key British Enterprises IML, Industrial Market Locations WOW, Who owns Whom
Profile(Datasolve)	World Reporter Grossbritannien International Vereinigte Staaten Finanzen und Oekonomie
Telesystemes, Frankreich	Pascal laline

Tabelle 2: Datenbanken, auf die von GeoNet her zugegriffen werden kann. Der Zugang wird über die Retrievalsprache Intelligent Interface (II) erleichtert.

wären auch Hinweise erwünscht, welche Dienste bei GeoNet nicht vorhanden sind, die aber andere Systeme bieten bzw. besser bieten.

Die Grunddienste wie persönliche Fächer oder schwarze Bretter sind kein Thema, die bietet jeder. Interessanter sind schon Zugang über Telefon, DA-TEX-P oder ISDN, oder Versenden und Empfang von FAXen. Entschei-

dend sind aber Kopplungen zu anderen Netzen und zu Datenbanken oder auch spezielle Fach-Bretter sowie die Nutzung von Telex.

Das Box-System von André Koppel aus Berlin, das wir ins Auge gefaßt haben, existiert schon auf einem hohen Niveau, ist aber noch weiter im Ausbau. Von daher können also leicht Anregungen und Wünsche einfließen.

Es gibt einen Zugang über Telefon bis 9600 bit/sekunde nach V.32, HST bis 14400 bps. Technisch ist auch ein DA-TEX-P-Zugang möglich, doch werden wohl die damit verbundenen nicht unerheblichen Kosten das verbieten. Ein Trailblazer-Modem kann gleichfalls kurzfristig bei Bedarf angeschlossen werden. Es existiert eine Anbindung ans FIDO-Netz, in Kürze ans UUCP-Netz, etliche andere Netze sind auch möglich.

Die Frage bezüglich Datenbanken ist noch nicht geklärt, doch vielleicht läßt sich da einiges machen. An die Datensicherheit ist ebenfalls gedacht. Die Paßwort-Datei ist verschlüsselt und ohne Paßwort ist keinerlei Zugriff auf persönliche Daten möglich. Das gilt auch ohne Einschränkungen für Super-Sysop und -Manager.

F. Unterstützung durch den CCD

Der CCD muß sich entscheiden. Bisher unterstützte er den Boxbetrieb finanziell durch Übernahme der monatlichen Grundgebühr; nach dem Absinken der Teilnehmerzahl steht das zur Disposition.

Der CCD kann und sollte aber weiterhin organisatorische Unterstützung bieten. Z.B. durch Übernahme der Rechnungsstellung wie bisher. Unterstützen vor allem aber durch Verhandlungsführung mit André Koppel von der Berliner Box oder Günther Leue von GeoNet. Mit unserem Clubmitglied André müßten die Box-Anforderungen und die daraus resultierenden Gebühren für die Teilnehmer besprochen werden. Mit Günther Leue, ob gegebenenfalls eine (wesentlich) kleinere CCD-Gruppe in GeoNet, dort wahrscheinlich im neuen GeoNet II-System, verbleiben kann und dennoch die finanziellen Sonderkonditionen des Clubs erhalten bleiben.

Für was sprechen sich die Clubmitglieder aus? Für die Unterstützung von GeoNet, von André Koppels Berliner Box, von beidem, oder für ganz etwas anderes? Der CCD muß sich entscheiden.

Literatur

Fritz Jörn, "Elektronische Nächstenliebe" ebnet Hierarchieschranken ein, FAZ, 29.10.91

Mitteilungen von W.Dworak, G.Link u.a. CCD-Mitgliedern

André Koppel's Mailbox

Liebe CCD-Mitglieder,

aufbauend auf dem Artikel von unserem Clubmitglied Wolfgang Fritz möchte ich Euch hier kurz erläutern, welche Leistungsmerkmale in einem kleineren privaten System realisierbar sind. Die Grundlage für dieses Angebot ist denkbar einfach. Ich beschäftige mich schon jahrelang beruflich mit allen möglichen Formen der Telekommunikation und betreibe seit geraumer Zeit ein eigenes Mailboxsystem. Dieses Mailboxsystem ist auch die Basis für die folgende Leistungszusammenfassung.

Zugänge: Zur Zeit ein Telefonzugang mit bis zu 14400 bit/s (V.21, V.22, V.22bis, V.32bis und HST), vier weitere Telefonzugänge sind beantragt und dürften etwa Anfang nächsten Jahres zur Verfügung stehen. Datex-P-Zugänge sind ohne technische Probleme realisierbar (auch kurzfristig). Die relativ hohen Einstiegskosten (5000,- DM) und die hohen laufenden Kosten (ca. 400,- DM monatlich) müssen jedoch gedeckt werden. Datex-P ist daher nur dann sinnvoll, wenn ein entsprechend hoher Bedarf vorhanden ist. Der ist bei den bisherigen BOX-Teilnehmern nicht gegeben, und es müßte sich zeigen, ob er im CCD groß genug ist.

Daten-Netze: Die Mailbox ist Node im Fido-Net. Das Fido-Net ist ein weltweiter Verbund von privaten Mailboxen mit einer hervorragenden Infrastruktur und einem recht reibungslosen und sehr kostengünstigen Nachrichtenversand. Unzählige Diskussionsforen mit einem sehr hohen Nachrichtenaufkommen auf den unterschiedlichsten Gebieten machen es zudem ausgesprochen interessant und informativ. Weltweit gibt es einige Millionen aktive Nutzer. Eine Anbindung der Mailbox an das UUCP-Netz wird zur Zeit gerade realisiert. Andere Netze können bei entsprechendem Interesse jederzeit hinzugenommen werden.

FAX-Gateway: Ein Fax-Gateway wurde schon testweise eingerichtet, ist jedoch zur Zeit wieder abgeschaltet, da kein Bedarf vorhanden ist. Da die Technik jedoch vorhanden ist, kann der FAX-Versand innerhalb eines Tages wieder realisiert werden. FAX-Empfang ist grundsätzlich auch möglich. Ein sehr großes Problem beim FAX-Empfang ist jedoch die Empfängererkennung. Selbst beim Einsatz einer leistungsfähigen OCR-Software wird es immer wieder vorkommen, daß ein Empfänger nicht

automatisch erkannt werden kann und daher der BIO-Computer gefordert ist (manuelle Nacharbeit).

Telex-Gateway: Telex ist technisch realisierbar. Hohe Einstiegskosten (3000,- DM) und eine hohe monatliche Grundgebühr (ca. 200,- DM) lassen jedoch wahrscheinlich kein brauchbares Kosten-/Nutzen-verhältnis aufkommen.

Nachrichtenversand: Jeder Anwender hat ein persönliches Postfach und kann auch Stellvertreter angeben. Sehr kostengünstig können Nachrichten innerhalb des Fido-Net und UUCP versandt werden. Dabei ist jedoch zu beachten, daß kommerzielle Meldungen in diesen Netzen nicht gern gesehen, teilweise ausdrücklich verboten sind. Ein Nachrichtenversand vom und zum GeoNet-System ist technisch ohne weiteres machbar.

Dateiversand: Bei der Mailbox handelt es sich um ein leistungsstarkes PC-System, das via Novell vernetzt ist. Beliebige Dateien (egal welche Rechnersysteme und/oder Formate) können daher mit allen derzeit gängigen Übertragungsprogrammen gesendet und empfangen werden. Es können also auf diesem Wege unproblematisch alle Dateien zwischen Geschäftsfreunden o.ä. als private Daten ausgetauscht werden. Daneben enthält die Mailbox eine sehr große und ständig wachsende Sammlung an Public-Domain und Shareware-Programmen, die von allen Anwendern größtenteils kostenlos empfangen werden können.

Datenbankanbindung: Theoretisch ist auch die Nutzung externer Datenbanken (wie bei GeoNet) möglich. Praktisch dürfte das jedoch an den enormen Kosten scheitern. Die gängigen Datenbankanbieter verlangen recht hohe monatliche Grundgebühren, die ja schließlich irgendwie wieder reinkommen müßten.

Kosten: So ein Mailbox-System verursacht ja auch Kosten. Obwohl ich das System nicht kommerziell betreibe, also keinen Gewinn damit machen will, entstehen doch Kosten z.B. durch Postgebühren und Geräteabnutzung, die wieder hereinkommen müssen. Viele reine Hobby-Boxen finanzieren sich durch Spenden oder Geschäftsböden laufen auf das Werbe-Konto. Ich halte es jedoch für beide Seiten für besser, mit genau festgelegten Gebühren zu arbeiten. Für den geschil-derten Leistungsumfang (also ohne

Datex-P, Telex und Datenbanken) ist an eine Grundgebühr (oder Mindestgebühr) von 5 bis 10 DM im Monat und eine Volumengebühr von 1 bis 2 Pfennig pro Kilobyte gedacht. Die genaue Struktur und Höhe muß entsprechend den Leistungsmerkmalen mit dem CCD-Vorstand festgelegt werden.

André Koppel

Anmerkungen von Werner Dworak

Das Datex-P-Netz wurde nicht so weiterentwickelt, wie es einmal aussah. Insbesondere gibt es keinen flächendeckend einheitlichen Tarif und ein Zugang mit 9600 bit/s ist noch nicht in Sicht. Daher ist es für jemanden, der abseits eines PADs wohnt, auf alle Fälle billiger, eine Box direkt anzuschließen als über Datex-P. Und mit einem schnellen Modem kann auch an einem PAD-Standort der Telefon-Zugang vorteilhafter sein. Von daher ist es fraglich, ob sich ein Datex-P-Anschluss für den Teilnehmer und für den Box-Betreiber rechnet. In Anlehnung an die (vermutlich wegfallende) Grundgebühr an GEONET wäre es denkbar, in der Box von André den Datex-P-Anschluß vom Club zu unterstützen, falls der Bedarf groß genug ist. Allerdings sieht es nach den derzeitigen Erkenntnissen nicht danach aus, daß Datex-P sich lohnt. In den sonstigen Fällen sind ja die Gebühren von André so niedrig, daß ein Zuschuß vom Club nicht nötig und sinnvoll ist.

Bei Telex und Datenbanken dürften kaum Chancen bestehen, daß genügend Teilnehmer zusammen kommen, daß man die entstehenden Kosten vernünftig umlegen kann. Da ist der CCD vermutlich zu klein.

Ein weiteres Clubmitglied, Axel Florstedt aus Essen, betreibt ebenfalls eine Mailbox, die am FIDO-NET hängt. Sein System ist etwas kleiner und er kann weniger Sonderleistungen anbieten, aber sonst gelten ähnliche Verhältnisse wie bei André. Jedoch für die Leute in Axels Nahbereich ist das allemal interessant. Ein Datenaustausch mit André ist vorgesehen, so dass Mitteilungen eines Mitglieds stets in beiden Boxen zu lesen sein werden. In gleicher Weise können sich vielleicht auch weitere Systeme anschließen.

Fortsetzung Seite 52

Universelle ROM-Karte

am Beispiel Paßwortkarte

von Werner Dworak, Alexander und Dieter Wolf

Wie schütze ich meinen Rechner vor fremdem Zugriff? Aus dieser Fragestellung entstand eine Problemlösung, die geeignet ist, auch andere Programme beim Booten des PC zu starten.

Ein Mindestschutz sollte für jeden Rechner, der auch Fremden zugänglich ist, verwirklicht sein: Paßwortabfrage beim Booten!

Damit läßt sich der Rechner halbwegs absichern, für den Schutz bei laufendem Rechner sind allerdings ergänzende Maßnahmen notwendig, auf deren Darstellung hier verzichtet wird.

Mit der Paßwortkarte wird eine Eigenschaft des PC-BIOS genutzt, die beim Booten dafür sorgt, daß z.B. BIOS-Ergänzungen von Bildschirmkarten und Laufwerkscontrollern erkannt und geladen werden. Dazu genügt es, in den entsprechenden Adressbereich ein ROM einzublenden.

Die im ROM enthaltenen Programme werden zwangsläufig bei jedem Booten abgearbeitet. Hier ist das Programm so gestaltet, daß der Rechner nur dann weiterläuft, wenn das richtige Paßwort eingegeben wurde. An-

sonsten wird einfach eine Endlosschleife beibehalten.

Das Beste an der Karte ist, daß im Grunde beliebige Programme ins ROM gebracht werden können.

Die Schaltung ist mit einem EEPROM (auch E²PROM genannt) aufgebaut. Das ist im Grunde nichts weiter als ein ROM das im Rechner beschrieben werden kann. Der Unterschied zum RAM ist, daß Schreibzugriffe auf ein EEPROM viel langsamer sind und die Informationen nach dem Ausschalten erhalten bleiben. Damit sind Änderungen des Programmes ebenso wie des Paßwortes ohne Probleme möglich. Mittels Schalter wird Kontakt B11 an WE (Write Enable = Pin 27, im Platinenlayout als "Brücke" bezeichnet) gelegt und damit der Schreibmodus aktiviert, das Programm kann ins EEPROM geladen werden.

Über DIP-Schalter wird die Adresslage eingestellt. Hierbei ist wichtig, mit Programmen wie MSD von Microsoft oder Checkit festzustellen, welche Adressbereiche nicht belegt sind (Tabelle 1).

Das ist jedoch nur eine typische Adressbelegung und keineswegs bei jedem Rechner so. Die Adressen von

Hauptspeicher, VGA-Karte und BIOS ROM sind bei jedem Rechner wie oben angegeben, während Festplattenkontroller-BIOS und EMS-Page Frame keineswegs immer an den oben angegebenen Stellen liegen müssen.

Für die Paßwortkarte sind 8k Adressbereich notwendig, das entspricht 200h (z.B. CC00h-CDFFh). Wer die Schaltung etwas abwandeln will und ein größeres ROM/EEPROM benutzt (z.B. ein 27128 oder 27256) muß natürlich mit mehr Platz rechnen. Am besten benutzt man eine der folgenden Adressen (soweit nicht belegt): C800h, CC00h, CE00h, D000h, D800h. Um beim Booten eingeblendet zu werden muß die Adresse im Bereich Cxxx bis Dxxx liegen.

Die Adresse der Karte errechnet sich aus der Stellung der DIP-Schalter. Jeder Schalter steht für eine Hexadezimalzahl (Tabelle 2).

Schalter	Hex
1	80
2	100
3	200
4	400
5	800
6	1000
7	2000
8	4000

Tabelle 2

Addiert man die Zahlen der geschlossenen Schalter und zählt noch 8000 Hex dazu erhält man die gewünschte Segmentadresse, z.B.

$8000+4000+800+400 = CC00$, wenn die Schalter 4, 5 und 8 geschlossen sind.

Um das Rechnen zu ersparen sind in Tabelle 3 die Schalterstellungen für einige gängige Adressen aufgeführt.

Gängige Adressbelegung:

0000h bis 9FFFh Hauptspeicher (640k)
 A000h bis BFFFh VGA Video RAM
 C000h bis C7FFh VGA Video ROM
 C800h bis CC00h eventuell Festplattencontroller
 (nicht bei AT-BUS- oder MFM-Platten)
 CC00h bis CFFFh frei
 D000h bis DFFFh eventuell EMS-Page Frame
 E000h bis EFFFh frei
 F000h bis F7FFh Erweitertes BIOS ROM
 F800h bis FFFFh BIOS ROM

Tabelle 1

Vorsicht! Adresskonflikte können z.B. den Zugriff auf Festplatten verhindern oder stören. Also vorher ausgiebig testen.

Adresse	Schalter geschlossen
C800	8-5
CC00	8-5-4
CE00	8-5-4-3
D000	8-6
D800	8-6-5
E000	8-7
E800	8-7-5

Tabelle 3

Aufbauhinweise: Entweder mit einer Lochrasterplatine für PC-Anschluß nach Schaltplan verdrahten oder die Layoutvorlage aus PRISMA auf doppelseitige Rohlinge belichten und ätzen, anschließend Bohrungen einbringen.

Vor dem Einsetzen in den Rechner Platine durchmessen! Also Layout verfolgen und auf unzulässige Verbindungen der unbestückten Platine am Busstecker messen. Für Mutige: An einem alten PC Platine einstecken und testen ob's raucht.

Erst wenn dieser erste Test erfolgreich war Kondensatoren und IC-Sokkel einlöten.

Danach Test wie oben.

Bei Erfolg mit IC's bestücken, Adresslage einstellen und programmieren.

Wichtig! Weder die Autoren noch der CCD e.V. übernehmen eine wie auch immer geartete Garantie für das ordnungsgemäße Funktionieren der Schaltung. Insbesondere kann keinerlei Haftung für Folgeschäden übernommen werden.

Die Software zur Paßwortkarte

Die Paßwortkarte soll die Schwierigkeit umgehen, die jeder hat, wenn er versucht einen Rechner abzusichern. Alle Softwarelösungen scheitern daran, daß man sie einfach umgehen kann, wenn man von einer eigenen Diskette bootet. Die einzige softwaremäßige Möglichkeit besteht noch darin, die Datenbestände zu verschlüsseln. Hier hat aber ein Bug der Software oder sonstige Fehler verheerende Folgen: Der Datenbestand ist zerstört.

Ein Ausweg aus dem Dilemma ist ein ROM, daß beim Booten automatisch aufgerufen wird und damit nicht umgangen werden kann. Einige neuere BIOSse (z.B. AMI) sind schon mit Paßwortabfrage ausgerüstet, aber auch für die Rechner, die kein solches BIOS haben kann dieser Schutz einfach nachgerüstet werden, mit einer einfachen Steckkarte. Natürlich ist es auch dann noch möglich den Rechner zu starten, indem man den Rechner aufschraubt und die Karte entfernt, aber in diesem Fall hätte der Übeltäter auch gleich die Platte mitnehmen können. Außerdem ist leicht möglich den Rechner vor Öffnen zu schützen.

Die Software der Paßwortkarte besteht im Wesentlichen aus zwei Teilen. Zum einen aus dem Boot-Programm im EEPROM der Karte, zum anderen aus den Utilities zum "Einbrennen" des Boot-Programms in das EEPROM und zum Ändern des Paßwortes auf der Karte.

Das Boot-Programm ist vollständig in Assembler (MicroSoft Version 5.1) geschrieben. Nach dem Start des Programms löscht es den Bildschirm, gibt eine Copyright-Meldung aus und fragt das Paßwort ab. Das Paßwort darf maximal 128 Zeichen lang sein und wird durch die <RETURN> Taste beendet. Dadurch ist es für Fremde nicht möglich, die Länge des Paßwortes herauszufinden. Das Paßwort ist im EEPROM nicht gespeichert, sondern nur seine Länge und eine daraus errechnete 32-Bit Signatur. Damit ist es praktisch unmöglich das Paßwort aus dem EEPROM-Inhalt heraus zu erkennen.

Das Programm vergleicht die Länge und die Signatur des eingegebenen Paßwortes mit den gespeicherten Werten. Bei falscher Eingabe hat man zwei weitere Versuche, dann wird eine Meldung ausgegeben und der Rechner unter dauerndem Piepsen des Lautsprechers "aufgehängt".

Das Bildschirmlöschen am Anfang kann je nach Geschmack auch weggelassen werden. Dazu ist im Quelltext (passwrom.asm) bei start: die Zeile mit call Clear_scr zu entfernen und die Subroutine Clear_scr ebenfalls zu löschen.

Ebenso läßt sich die Anzahl der möglichen Versuche leicht ändern, Werte zwischen 1 und 0FFFFh sind möglich.

Dazu muß in der 2. Zeile nach start: der Wert geändert werden, mit dem das CX-Register bestückt wird. Das Programm füllt in dieser Version 512 Byte. Dadurch sind noch 7,5 kByte im dem 8 kByte großen EEPROM frei, die noch belegt werden können. Einer Vergrößerung des Programms oder dem Hinzufügen von weiteren Programmen steht also nichts im Wege.

Das BIOS bemerkt auch, wenn mehrere Programme in einem ROM hintereinander stehen. Die Programme müssen nur mit 55AAh und ihrer Längenkennung beginnen. Verändert man das Paßwortprogramm so weit, daß es größer als 512 Byte wird, muß man natürlich daran denken, eine andere Längenkennung einzutragen.

Das Programm zum Ändern des Paßwortes und Laden eines ganzen Programmes in das EEPROM wurde im mixed-language Verfahren programmiert. Das Hauptprogramm (passwort.c) ist in C (MicroSoft Version 6) geschrieben, die Module zur Berechnung der Signatur, Berechnung der ROM-Checksumme, und Programmieren des EEPROMs in Assembler (pwextern.asm).

Beim Aufruf des Programms muß die Adresse des EEPROMs in der C-üblichen hexadezimalen Notation angegeben werden, d.h. z.B.

```
passwort.exe 0xCC000000
```

wenn das EEPROM auf der Adresse CC00:0000 liegt. Die letzten vier Stellen müssen immer Nullen sein !

Das Programm prüft dann, ob an der angegebenen Adresse das Paßwortprogramm vorhanden ist und ruft dieses dann auf, ansonsten bricht es mit einer Fehlermeldung ab. (Bei der Erstbestückung des EEPROMs muß man den entsprechenden Teil des Quelltextes "lahmlegen", um das Paßwortprogramm laden zu können.) Nach der Eingabe des korrekten Paßwortes fährt das Programm fort.

Man hat jetzt 2 Optionen. Mit "1" kann man das Paßwort ändern, mit "2" ein neues Programm laden. In beiden Fällen wird ein neues Paßwort abgefragt und zwar zur Sicherheit doppelt. Wenn das neue Paßwort richtig eingegeben wurde, folgt die Aufforderung den Programmierschalter der Karte zu schließen. Nach erfolgreicher Programmierung folgt die Aufforderung den Schalter wieder zu öffnen. Bei der

Passwort.c

```

/**** Passwort.c *** Version 1.00 *** Microsoft-C 6.0 ****/
/* Programm zur Änderung des Passwortes der PC-Passwortsicherungskarte
   Copyright (C) 1990 by W.Dworak,A.Wolf
*/
#include <stdlib.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <bios.h>

#define FALSE 0
#define TRUE 1

void Fehler(char *, int);
int ReadCmdLine(int, char **, char far * *);
int VerifyRom(char far *);
void ChangePw(char far *);
void NewProg(char far *);
int ReadPw(char *, int *);
extern char CALC_CRC(char far *);
extern void CALLROM(char far *);
extern int CALC_SIG(char *, int);
extern int WRITE_PW(char far *, int, int);
extern int WRITE_CRC(char far *, char);
extern int WRITE_PRG(char far *);

void main(int argc, char **argv)
( char far *RomAddress;
  char Passwort[128], DummyChar;
  int PwSig, PwLaenge;
  int DummyInt;
  PC-Passwortsicherungskarte Version 1.0a\n";
  Copyright (C) 1991 W.Dworak, A.Wolf\n";
  Passwortänderungsprogramm\n";
  !! A C H T U N G !!\n";
  printf("Dieses Programm benötigt das aktuelle Passwort. Sollte Ihnen\n");
  printf("dieses Passwort NICHT bekannt sein, brechen Sie mit <ESC> ab.\n");
  printf("Andernfalls wird der Rechner blockiert!\n");
  printf("Zur Eingabe des aktuellen Passworts <RETURN> drücken.\n");
  DummyInt = getch();
  if (DummyInt != 13)
  ( if (DummyInt==0)
    getch();
    exit(1); )
  if (!ReadCmdLine(argc, argv, &RomAddress))
  Fehler("Falsche Parameter!\n",1);
  printf("ROM-Adresse: %P\n", RomAddress);
  if ((DummyInt = VerifyRom(RomAddress)) != 0)
  if (DummyInt == 1)
  Fehler("ROM an der angegebenen Adresse nicht gefunden!\n",1);
  else
  Fehler("ROM-Checksumme falsch!!\n",2);
  CALLROM(RomAddress+3);
  printf("\n1.....Passwort ändern\n");
  printf("2.....Programm neu laden (PASSWROM.BIN)\n");
  printf("<ESC>..Ende\n");
  printf("Bitte wählen ...");
  DummyChar = (char) getch();
  if ((DummyChar == '1') || (DummyChar == '2'))
  ( if (!ReadPw(Passwort, &PwLaenge))
    exit(1); )
  PwSig = CALC_SIG(Passwort, PwLaenge);
  printf("\nZum Umprogrammieren des EEPROMs bitte den ");
  printf("Programmierschalter\neinschalten!\n");
  printf("Beliebige Taste drücken ...");
  if (getch() == 0)
  if (DummyChar == '2')
  if (!WRITE_PRG(RomAddress))
  Fehler("\nKonnte Programm nicht schreiben!\n",2);
  if (!WRITE_PW(RomAddress, PwSig, PwLaenge))
  Fehler("\nKonnte Passwort nicht schreiben!\n",2);
  DummyChar = CALC_CRC(RomAddress);
  if (!WRITE_CRC(RomAddress, DummyChar))
  Fehler("\nKonnte Checksumme nicht schreiben!\n",2);
  printf("\nDas EEPROM wurde erfolgreich umprogrammiert.");
  printf("Beliebige Taste drücken ...");
  if (getch() == 0)
  getch();
  printf("\n");
  }
  exit(0);
)

int ReadCmdLine(int argc, char **argv, char far * RomAddress)
( char *DummyChar;
  if (argc != 2) return(FALSE);
  *RomAddress = (char far *) strtoul(++argv, &DummyChar, 0);
  return(TRUE); )

void Fehler(char *Meldung, int ErrCode)
( printf(Meldung);
  exit(ErrCode); )

int VerifyRom(char far *RomAddress)
( char * DummyStr;
  int Counter;
  DummyStr = "PWBRX01F\x0BC";
  for(Counter = 0; Counter < 6; Counter++)
  ( if (*(RomAddress + 0x1F5L + (long) Counter) != DummyStr[Counter])
    return(1); )
  if ((char *)RomAddress + 0x1FFL) != CALC_CRC(RomAddress))
  return(2);
  return(0); )

int ReadPw(char * Passwort, int * Laenge)
( int Index, Index2;
  char Passwort[128], Passwort2[128];
  printf("\nNeues Passwort eingeben: ");
  Index = 0;
  while ((Index < 128) && ((Passw[Index] = (char) getch()) != '\r'))
  Index++;
  Passwort[Index] = '\0';
  printf("\nPasswort wiederholen: ");
  Index2 = 0;
  while ((Index2 < 128) && ((Passw2[Index2] = (char) getch()) != '\r'))
  Index2++;
  Passwort[Index2] = '\0';
  if (Index != Index2)
  ( printf("Passwörter verschieden lang!\n");
    return(FALSE); )
  for (Index = 0; Index < Index2; Index++)
  ( if ((Passw[Index] != Passwort2[Index]) || (Passw[Index] == '\0'))
    Passwort[Index] = Passwort2[Index];
    *Laenge = Index2;
    return(TRUE); )
)

```

PxExtern.asm

```

;-----
; PWEXTERN.asm Externe Funktionen von PASSWORT.C
; Version 1.0a *** MASH 5.10 *** A.Wolf, W.Dworak
;-----
.MODEL SMALL
.CODE

PUBLIC _CALC_CRC
PUBLIC _CALLROM
PUBLIC _CALC_SIG
PUBLIC _WRITE_FW
PUBLIC _WRITE_CRC
WAHR EQU 1
FALSCH EQU 0

;-----
; extern char CALC_CRC(char far *)
; Berechnet die ROM-Checksumme
;-----
_ROM_ADR EQU [bp+4]

_CALC_CRC PROC
push bp
mov bp,sp
les ROM_ADR
mov cx,FFFF
xor ax,ax
add ax,ROM_ADR
inc bx
loop CALC_CRC1
mov cl,al
mov ax,0100h
sub ax,cx
pop bp
ret

_CALC_CRC ENDP

;-----
; extern void CALLROM(char far *)
; Ruft das Passwort-ROM auf (zur Passwortabfrage)
;-----
_ROM_ADR EQU [bp+4]

_CALLROM PROC
push bp
mov bp,sp
push si
push di
push ss
push ds
call DWord Ptr ROM_ADR
pop ds
pop ss
pop di
pop si
pop bp
ret

```

```

_CALLROM ENDP
;-----
; int CALC_SIG(char *, int)
; Berechnet die Signatur des Passwortes
;-----
_PASSW EQU [bp+4]
_LAENGE EQU [bp+6]

_CALC_SIG PROC
push bp
mov bp,sp
push si
xor dx,bx
mov cx,LAENGE
mov si,PASSW
cld

lods Byte Ptr [si]
call SIGNAT
loop Calc_s1
mov ax,cx
call SIGNAT
call SIGNAT
mov ax,bx
pop si
pop bp
ret

_CALC_SIG ENDP

PROC NEAR
push cx
mov cx,8

rol al,1
rcl bx,1
jnc short signat_2
xor bx,0A097h

loop signat_1
pop cx
ret

_SIGNAT ENDP

;-----
; extern int WRITE_FW(char far *, int, int)
; Schreibt das neue Passwort (Signatur+Länge) in das EEPROM
;-----
_ROM_ADR EQU [bp+4]
_SIGNTR EQU [bp+8]
_LAENGE EQU [bp+10]

_WRITE_FW PROC
push bp
mov bp,sp
push si

```

```

les bx,ROM_ADR
add bx,1FBh
mov si,8
mov cx,4
push cx
mov cx,0FFFFh
mov al,Byte Ptr ss:[bp+si]
mov Byte Ptr es:[bx],al
dec cx
jcxz Wrt_pw_err
cmp Byte Ptr es:[bx],al
jne write_pw2
pop cx
inc si
inc bx
loop write_pw1
mov ax,WAHR
jmp Wrt_pw_ret
pop cx
mov ax,FALSCH
pop si
pop bp
ret
ENDP

;-----
; extern int WRITE_CRC(char far *, char)
; Schreibt die Prüfsumme in das EEPROM
;-----

ROM_ADR EQU [bp+4]
CRC EQU [bp+8]
PROC
push bp
mov bp,sp
les bx,ROM_ADR
add bx,1FFh
mov al,CRC
mov Byte Ptr es:[bx],al
mov cx,0FFFFh
dec cx
jcxz Wrt_crc_err
cmp Byte Ptr es:[bx],al
jne write_crc1
mov ax,WAHR
jmp Wrt_crc_ret
mov ax,FALSCH
pop bp
Wrt_crc_err:
pop bp
Wrt_crc_ret:
ret
ENDP

;-----
; extern int WRITE_PRG(char far *)
; Schreibt das Programm PASSWROM.BIN in das EEPROM
;-----

ROM_ADR EQU [bp+4]
PROC
push bp

```

```

mov bp,sp
push si
push di
push ss
push ds
mov ax,@data
ds,ax
mov ax,3D00H
dx,OFFSET fname
int 21h
jc fehler
mov handle,ax
mov bx,ax
mov ah,3FH
mov cx,200H
dx,OFFSET puffer
int 21h
jc fehler
mov ah,3EH
int 21h
jc fehler
mov cx,200H
si,OFFSET puffer
les di,ROM_ADR
lies: lodsb
mov es:[di],al
mov dx,0FFFFh
dec dx
jz fehler
cmp es:[di],al
jne noch
inc di
loop lies
mov ax,WAHR
jmp return
fehler:
mov ax,FALSCH
pop ds
pop ss
pop di
pop si
pop bp
ret
_WRITE_PRG ENDP
.DATA
fname db 'PASSWROM.BIN',0
handle dw 0
.DATA?
puffer db 200h dup (?)
END

```

PasswRom.asm

```

PAGE 66,132
;-----
; PASSWROM.asm Passwortsicherungskarte Boot-ROM Programm
; Fragt das Passwort ab, und blockiert nach drei vergeblichen Versuchen
; den Rechner, ansonsten geht's zurück zum BIOS
; Copyright (c) 1991 by W.Dworak, A.Wolf
;-----

CODE
TRUE EQU 1
FALSE EQU 0
segment byte public
assume cs:CODE

PUBLIC
PASSW

org 0

passw proc far
db 055h ; ROM-Kennung 55AAh
db 0AAh ; ROM-Größe
db 1 ; Bildschirm löschen
; 3 Versuche
Clear_scr
mov cx,3
mov bx,offset titel
print
try_again:
call Test_CRC
cmp ax,TRUE
je back
dec cx
short hangup
mov bx,offset msg1
jmp try_again

back:
retf

hangup:
mov bx,offset msg2
print
dx,43h
mov al,0B6h
out dx,al
dec dx
xor al,al
out dx,al
mov al,08h
out dx,al
mov dx,61h
in al,dx
or al,00000011b
out dx,al
cli
jmp short hangup_1

passw endp

Clear_scr
near ax
push bx
push cx
dx,ah
mov ah,0Fh
; Video display ah=functn 0Fh
; get state, al=mode, bh=page
int 10h
near ax
push bx
push cx
dx,ah
mov ah,0EH
; Video display ah=functn 0EH
; write char al, teletype mode
int 10h
near ax
push bx
push cx
dx,ah
mov ah,0
; Video display ah=functn 0h
; get char al & attrib ah @curs
int 10h
mov bh,ah
mov ah,6
mov al,0
mov ch,0
mov cl,0
mov dh,18h
mov dl,4Fh
int 10h
; Video display ah=functn 06h
; scroll up, al=lines
mov ah,0Fh
int 10h
; Video display ah=functn 0Fh
; get state, al=mode, bh=page
mov ah,2
mov dx,0
int 10h
; Video display ah=functn 02h
; set cursor location in dx
pop dx
pop cx
pop bx
pop ax
retn
endp

proc push
ax
push bx
push cx
mov al,cs:[bx]
cmp al,0
je short print_3
call Putch
inc bx
jmp short print_2
pop cx
pop bx
pop ax
retn
endp

proc proc
push ax
push bx
push cx
mov ah,0Fh
; Video display ah=functn 0Fh
; get state, al=mode, bh=page
int 10h
pop ax
mov ah,0EH
mov bl,0
int 10h
; Video display ah=functn 0EH
; write char al, teletype mode
pop bx
pop ax
retn
endp

proc mov
near ax,0
Taste
endp

```

```

; Keyboard i/o ah=function 00h
; get keyboard char in al, ah=scan
16h int ; Keyboard i/o ah=function 00h
; get keyboard char in al, ah=scan
retn
endp

Taste
proc
push
mov ah,1
int 16h
pop ax
retn
endp

Tast_Dr
proc
push
xor bx,bx ; Zero register
xor cx,cx
proc
push
xor bx,bx ; Zero register
xor cx,cx
retn
endp

Tast_Dr
proc
push
xor bx,bx ; Zero register
xor cx,cx
retn
endp

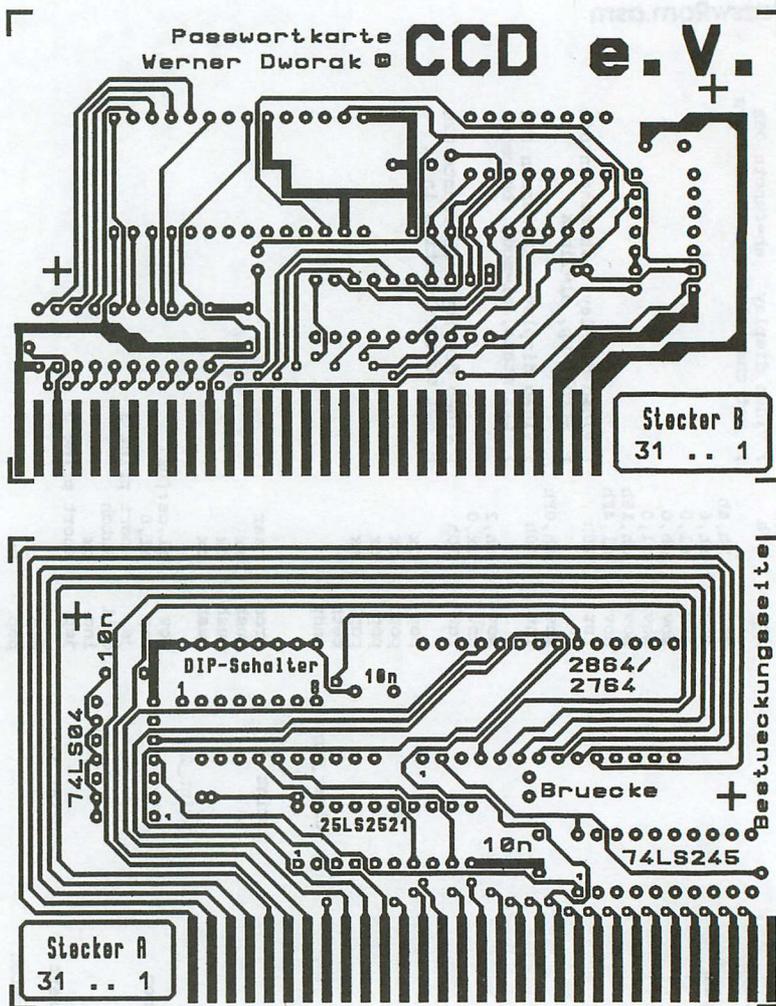
Test_CRC
proc
push
xor bx,bx ; Zero register
xor cx,cx
retn
endp

Test_c2:
proc
push
xor bx,bx ; Zero register
xor cx,cx
retn
endp

Test_c3:
proc
push
xor bx,bx ; Zero register
xor cx,cx
retn
endp

pw_err:
Test_back:
Test_CRC
CRC
crc_2:
crc_3:
CRC

```



```

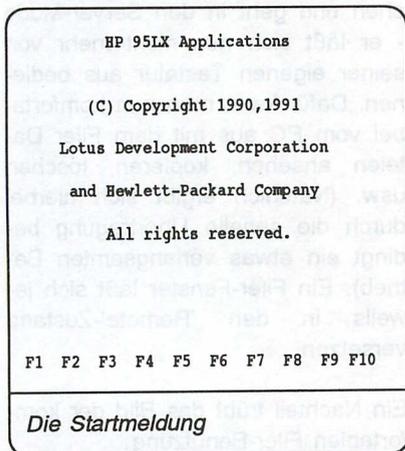
; Rotate thru carry
; Rotate thru carry
; Loop if cx > 0

```

HP95: Connectivity Pack

Ein Erfahrungsbericht von Gilles Kohl

Unter dem Namen 'Connectivity Pack' bietet HP zum Preis von DM 220.- ein Kabel und Software an, mit dem sich der heimische PC und der HP 95LX (u.a.) zwecks Datenaustausch zusammenschließen lassen. Was das Paket genauer leistet und welche Erfahrungen ich damit machen konnte ist Thema dieses Beitrags (mein erster in PRISMA).



Inhalt

Neben dem reinen Übertragungskabel (Länge: 1.5m) finden sich ein Handbüchlein, ein 9- auf 25 Pol-Adapter und Disketten im 5.25" (360K) und 3.5" (720K) Format im Paket.

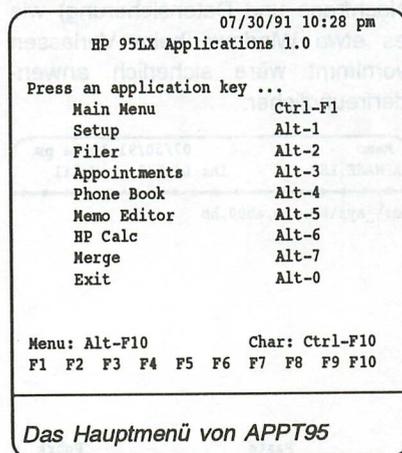
Voraussetzung für den Einsatz sind eine freie serielle Schnittstelle, MSDOS Version 3.1 oder höher und 512K freies RAM.

Das Kabel aus dem Connectivity Pack unterscheidet sich physikalisch vom HP-48er Kabel durch eine ca 3 cm lange Verdickung am 95er Ende (in Plastik gegossen und nicht aufzukriegen) - das 48er Kabel funktioniert jedoch - soweit ich feststellen konnte - genauso.

Die Software besteht im Prinzip aus

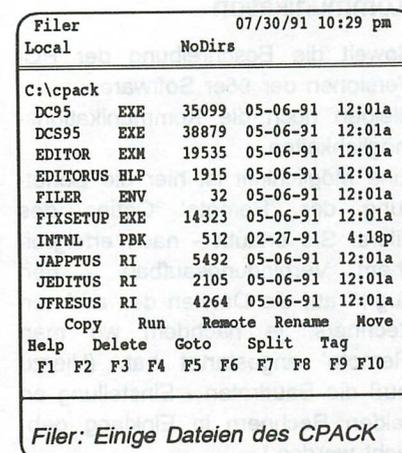
zwei unabhängigen Teilen: APP95 und DOS Connect.

Die Software: APP95



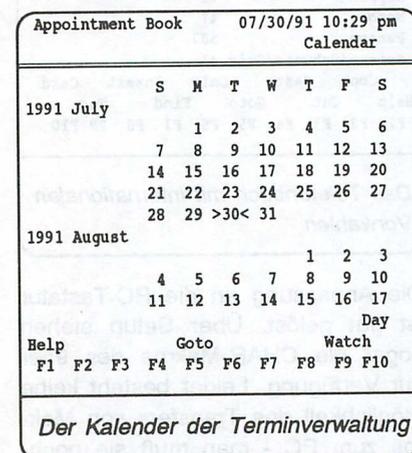
Bei APP95 handelt es sich im wesentlichen um die PC-Versionen der in den HP95 integrierten Anwendungen - mit ein paar Unterschieden. Zum einen ist Lotus 1-2-3 und das Datacomprogramm nicht enthalten, zum andern gibt es ein weiteres Programm: Merge.

Nach dem mitgelieferten Benutzerhandbuch zu schließen müsste auch ein Programm namens Translate vorhanden sein.



Merge erlaubt es zwei Terminkalender (*.ABK) oder auch zwei Telefonlisten (*.PBK) zu vereinigen. Dies ist gerade dann sinnvoll, wenn parallel am PC und am 95er geändert bzw. ergänzt wurde.

Translate soll die Übersetzung obiger Dateitypen von und aus den Formaten Lotus Metro, Borland Sidekick und PC Tools erlauben. Translate ist jedoch einfach nicht vorhanden. Im Benutzerhandbuch ist es vollständig beschrieben, auf den Disks jedoch nicht zu finden. In einer README-Datei wird diese Abwesenheit indirekt durch eine Auflistung der enthaltenen Applikationen bestätigt; auch hier fehlt Translate. Kein Wort der Erklärung allerdings - es wird nicht erwähnt.



Startet man APP95, so findet man sich in einem Hauptmenü wieder, welches ungefähr der 'Visitenkarte' des HP95 entspricht. Die blauen Anwendungstasten des HP95 sind auf ALT-<Ziffer> gelegt:

- ALT-1 Setup
- ALT-2 Filer
- ALT-3 Appointment Book
- ALT-4 Phone Book
- ALT-5 Memo Editor

- ALT-6 HP-Calc
- ALT-7 Merge
- (ALT-8 Translate, nach dem Handbuch)
- ALT-0 Exit.

Der optische Eindruck beim Start ist etwas enttäuschend - APP95 läuft mit 16x40 Zeichen im Textmodus in Bildschirmmitte zentriert.

Dementsprechend mickrig sieht es auch aus. Wer das HP95DEMO gesehen hat weiß, daß es auch anders (ansprechender!) geht. Lotus scheint sich Probleme mit Grafik ganz erspart zu haben - so kann HP-Calc auch, anders als die 95er-Version, keine Funktionsgraphen zeichnen. Ein weiterer Unterschied: die Alarme und der Timer des Appointment Book funktionieren nicht. (Der Timer funktioniert schon, 'klingelt' aber nicht.)

Phone Book		07/30/91 10:33 pm	
_1.pbk		Index	
Eire	353		
España	34		
France	33		
Italia	39		
Luxembourg	352		
México	52		
Nederland	31		
Nippon	81		
Norge	47		
Panamá	507		
Suisse/Schweiz/Sviz	41		
Copy	Paste	Only	Insert
Card			
Help	Cut	Goto	Find
Tag			
F1	F2	F3	F4
F5	F6	F7	F8
F9	F10		

Das Telefonbuch mit internationalen Vorwahlen

Die Anpassung an die PC-Tastatur ist gut gelöst. Über Setup stehen sogar die CHAR-Makros des 95er zur Verfügung. Leider besteht keine Möglichkeit des Transfers von Makros zum PC - man muß sie nochmals eingeben. Die Lage der 'Menü' Taste ist frei definierbar (ALT-F10 ist Voreinstellung), ebenso kann man wählen mit welcher Taste man APP95 verlassen kann. Anders als beim 95er kommt man mit CTRL-F1 ins Übersichtsbild, auch wenn noch Applikationen offen sind.

Etwas unglücklich ist die Farbwahl in Appointment Book, Tage mit Terminen werden (auf meiner VGA-Karte) hellblau angezeigt. Von

nichtmarkierten (dunkelblauen) schwer zu unterscheiden.

Genau wie beim HP95 lassen sich die einzelnen Anwendungen kreuz und quer aufrufen und behalten dabei exakt den vorherigen Zustand. Leider muß man auch PC-seitig alle Anwendungen schließen, bevor man DOS-Programme über den Filer aufrufen kann, und auch das Verlassen von APP95 geht über ALT-0 nur dann, wenn man alle anderen Anwendungen geschlossen hat. Ein implizites Schließen (mit Nachfrage und Datensicherung) wie es etwa Windows beim Verlassen vornimmt wäre sicherlich anwenderfreundlicher.

Memo		07/30/91 10:34 pm	
APNAME.LST		Ins Line:1 Col:1	
c:_sys\hb.exe,ab00,hb			
Paste		Pgbrk	
Help	Find	Mark	
F1	F2	F3	F4
F5	F6	F7	F8
F9	F10		

Der Memo-Editor

Wer obige Zeile in eine Datei namens C:_DAT\APNAME.LST reinschreibt CTRL-ALT-DELETE drückt und danach ALT-Setup drückt erlebt eine Überraschung!
(Dies funktioniert nur auf dem HP95)

Kommunikation

Soweit die Beschreibung der PC-Versionen der 95er Software.

Bleiben noch die Kommunikationsmöglichkeiten.

Eine Möglichkeit ist hier die Benutzung der 'Remote' Option des Filers. Sie erlaubt - nach erfolgreichem Verbindungsaufbau - den Zugriff auf die Dateien des anderen Rechners, je nachdem wo man 'Remote' angestartet hat. (Hierzu muß die Baudraten - Einstellung an beiden Rechnern in Einklang gebracht werden.)

Solve Catalog	
o	Use arrows to change columns; names on left equations on right.
o	Enter equation or name: select column and row, type entry, press ENTER.
o	Edit, DELETE, INSERT, or Calc an entry: highlight entry, press F2, F3, F4, or F9.
o	Search for text: Highlight column, Press F7 (Find), type text, press ENTER (or F5, F7). Press ESC to quit.
o	Copy entries: CTRL and F2, F3, F4.
o	Erase all equations and data: MENU E D
o	Equation files: Press MENU F.

Press ESC to exit Help. Help index
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10

Auch die Help-Funktion ist verfügbar

Beispiel:

auf beiden Rechnern (PC und 95er) startet man den Filer, drückt dann auf dem PC [F7] (split) und [F6] (remote). Der 95er wird angesprochen und geht in den Server-Mode - er läßt sich nun nicht mehr von seiner eigenen Tastatur aus bedienen. Dafür kann man nun komfortabel vom PC aus mit dem Filer Dateien ansehen, kopieren, löschen usw. (Natürlich ergibt sich hierbei durch die serielle Übertragung bedingt ein etwas verlangsamten Betrieb). Ein Filer-Fenster läßt sich jeweils in den 'Remote'-Zustand versetzen.

Ein Nachteil trübt das Bild der komfortablen Filer-Benutzung:

Es lassen sich PC-seitig nur COM1 oder COM2 benutzen. Wer (wie ich) die Maus auf COM1 und ein Modem auf COM2 angeschlossen hat muß umbauen oder umstecken, oder benutzt den Filer erst gar nicht zu Kommunikationszwecken. Eine COM3 oder COM4 wird nämlich - obwohl vorhanden - nicht unterstützt.

Glücklicherweise liefert HP die Alternative im Connectivity Pack gleich mit. Es ist diese Alternative - DOS Connect genannt - die das bislang insgesamt doch etwas trübe Bild etwas aufhellt. (DOS Connect ist übrigens nicht von Lotus sondern von Traveling Software)

DOS Connect besteht aus zwei Teilen, einem Server der auf dem 95er läuft, und einem residenten Programm, welches auf dem PC

gestartet wird.

DC95 (der PC-Teil) kann auch COM3 oder COM4, ja sogar gänzlich andere Basisadressen und Interrupts behandeln - Voreinstellung beim Aufruf ist COM1, anderes kann per Option angewählt werden. Ist erstmal eine Verbindung zwischen beiden Teilen aufgebaut, (115200 Baud) können die Laufwerke des 95er direkt (als Laufwerke G: etc ... beispielsweise) vom PC aus angesprochen werden.

```

Phone Book Merge      07/30/91 10:38 pm
Press F3 to select first input file

Input file 1:  INTNL.PBK
Input file 2:  PRIV.PBK
Output file:   MERGE.PBK

Conflict:      Query
Beep:          Enabled
Log file:      MERGE.LOG
                Start
Help   Files
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10

Das Programm Merge. Hier wird
grad aus INTNL.PBK und PRIV.PBK
die Datei MERGE.PBK zusammen-
gestellt.

```

DOS Connect läuft im Hintergrund und macht diese Umsetzung trans-

parent für alle Anwendungsprogramme. Die 95er Laufwerke haben dabei den Status von Netzwerklaufrufen - FORMAT, SYS und ähnliche Befehle sind daher nicht erlaubt. DOS Connect arbeitet problemlos mit Norton Commander und ähnlichen zusammen - nur Windows' FileManager versagte bei mir den Dienst.

Das Schöne beim Arbeiten über DOS Connect (im Vergleich zum Arbeiten über den Filer) ist, daß man seine am PC gewohnten Hilfsprogramme auf 95er Dateien loslassen kann. Dies ist auch mit den in APP95 enthaltenen Programmen möglich - es lassen sich von Phone Book oder Appointment Book (PC-Version) aus direkt die 95er-Dateien bearbeiten. Mit dem Filer allerdings kann man die Netzwerklaufrufe nicht ansprechen. Dies liegt daran, daß der Filer die serielle Schnittstelle, welche schon von DOS Connect benutzt wird, für die Remote-Funktion braucht.

Plusminus

Positiv:

+Fast vollständige Palette der 95er Utilities für den PC

+Problemlose Installation
+9/25 Pol Adapter
+Dual Media (5.25" und 3.5")
+brauchbares Handbuch
+DOS Connect

Negativ:

- hoher Preis
- Filer unterstützt nur COM1 und COM2
- Programm 'Transfer' kommentarlos gestrichen
- keine Grafikerunterstützung
- Filer kann bei Programmaufruf nicht auf Platte oder EMS auslagern - nur kleine Programme aufrufbar.
- Beep läßt sich nicht abschalten

Sollten sich Interessenten für weitere Informationen über den HP95 im CCD finden, so wäre ich bereit noch weitere Beiträge, z.B. undokumentierte Grafikfunktionen oder Dateiformate des Phone-Manager, bringen.

Über ein Feedback würde ich mich sehr freuen.

Gilles Kohl

Rudolfstraße 12
D7500 Karlsruhe

Fragestunde zum HP28S und HP48SX

Das große Programmierfieber ist beim HP28S oder HP48SX wohl noch nicht ausgebrochen.

Vielleicht liegt es ja daran, daß die Geräte noch nicht so verbreitet sind, vielleicht warten die Besitzer nur auf PRISMA statt selbst zu programmieren, aber vielleicht bestehen ja auch elementare Probleme beim Umgang mit HP28S oder HP48SX.

Für diesen Fall biete ich einen Kummerkasten in Pas-de-deux Teil 8 an.

Wer Fragen zu den Themen:

- ☞ Programme vom HP41 umschreiben;
- ☞ wie ersetze ich RCL IND oder STO ...
- ☞ Was mache ich mit synthetischer Programmierung (z.B. RCL d) und wo ist das ALPHA-Register ...
- ☞ Programmierung des HP28S
- ☞ wie geht rekursive Programmierung
- ☞ Programmierung des HP48SX

hat, soll mir (Ralf Pfeifer, Rubensstraße 5, W5000 Köln 50) schreiben, Postkarte genügt.

Die Fragen werden anonym abgedruckt, außerdem gibt es keine dummen Fragen.

Auf eine Postkarte kommen wahrscheinlich 20 Anwender, die sich nicht getraut haben zu fragen.

Ich habe allerdings keine Ahnung von SYSEVALs oder vom Betriebssystem, das überlasse ich lieber Peter Röhl (HP28S) und Georg Hoppen (HP48SX).

Ralf Pfeifer



Atari-Disketten des Jahres 1991

von Werner Müller

CCD_INFO.031

CHMELEON.119 Accessory zum Nach-
laden weiterer ACC
HP_DJ HP_DESKJET Utilities
MONO23 Diskettenmonitor
SYS_MON.105 Systemmonitor

CCD_INFO.032

LGS18B Alternative File Selektor
Box
NEWFONTS Alternative System-
fonts für Atari laden
(Mit Source im
Assembler von CCD-
Mitglied Müller)
RUFUS.TOS Terminal Programm
(GEM)
SAGROTAN Viruskiller Vers. 4.17
UNLZH Decomprimieren von
LZH-Archiven
VAULT201 Schnelles Backup-
programm für Atari-
Festplatten

CCD_INFO.033

AUTOSORT Hilfsprogramm zum
Sortieren von Dateien
im Autoordner
(Reguliert die Reihen-
folge des Starts)
CHEETAH3 Update des schnellen
Filekopierers
DFIND Filefindprogramm (Wo
war doch gleich....??)
DFORMAT Diskettenformatierpro-
gramm
DSKCH32A Diskettentestpro-
gramm
EPROMDSK.PD Programme in ein
Eprom?
IMAGELAB.TTP Bildverarbeitungspro-
gramm sw (in Deutsch)
MAXON2.DOC Maxon-Pascal Version
1.5 Bericht.
NVDI.DOC NVDI-Test
OMIKRON.DOC Omikron was neues?
PACKER.V13 Programmpacker
(u.a. Erstellen von
selbstextrahierenden
Archiven)

TOSPATCH.V13 Patchprogramm mit
Patches für Tos 1.4
CMDMENTS Was man beim Pro-
grammieren des Atari-
ST beachten sollte
(englisch)

CCD_INFO.034

AIM Bildverarbeitungssys-
tem aus den
Niederlanden sw und
Farbe
(Dokumentation in
Englisch)

CCD_INFO.035

CHECKHD Testprogramm für
Festplatten
DIASCHAU Ansehen von Bilder-
serien
DOUBLECLICK Ein Sack voll Utilities
GEM_SHOW Ansehen von Bildern
(Metafiles)
GEM_VIEW Ansehen von Bildern
(Pixelgraphik-Formate)
MORTIMER Omikron.Basic Pro-
gramm zum Aufrufen
von Mortimer
PCLCOMP Komprimieren von
PCL Dateien (HP-
Druckersprache)
ULTRA101 Diskettenpacker nötig
für *.ULT auf
CCD_DISK.038
UUCODE10 UUcode/UUdecode
ZOO.TTP Zoo(Packer) Version 2.1

CCD_INFO.036

ANALMETA Metafile Analyse Pro-
gramm
IDEALIST.3_0 ASCII-Druckprogramm
(auch für Deskjet)
LZH201E Packer neue Versio-
nen (schon wieder)
SCHREIBK.URS Schreibmaschinen-
Lernprogramm
(aus Copyright Grün-
den ohne Lektionen)
(Erst die Tastatur,
dann den Rechner

beherrschen)
VKILL384 Viruskiller aus den
USA neuste Version
WINXV13 Viele viele GEM-
Fenster

CCD_INFO.037

COMPO Entschlüsseln von
That's write Druk-
kertreibern
(von CCD-Mitglied
Müller)
EYES Wo ist die Maus?
GILBERT von CCD-Mitglied Gil-
bert
KRUEGER von CCD-Mitglied
Krüger
MAPMEM Ausgeben der Spei-
cheraufteilung
VIDEOTXT Test eines Video-
textdecoders für den
Atari-ST

CCD_INFO.038

COLACALC Ein kleiner Rechner in
der unteren Bild-
schirmzeile
PRINTER Druckerprogramm/
Spooler
SQUEEZIM Imagedateien kompri-
mieren
ULTILITA.ULT utilita Diskette in einer
Datei...: (ingesandt
von Herrn Krüger)

Anmerkung: Die Sharewarepro-
gramme SYSMON, CAMELEON
und RUFUS sind für jedermann von
mir gegen SAFU mit Diskette zu
erhalten.

Ausblick 1992:

Entgegen meinen Befürchtungen im
Herbst 1991 gibt es doch noch
"neue" Programme, die es rechtfertigen,
Atari-Disketten mindestens
noch bis Ende 1992 herauszugeben.
Falls sich aber die Lage auf
dem PD-Disketten-Markt (der Anteil
von Schrott oder von Wiederholun-

Fortsetzung auf Seite 61

Pas de deux

Teil VII

Tips & Tricks für HP28/48SX von Ralf Pfeifer

Zuerst ein paar Worte zum Speichermanagement.

Hier haben 28S und 48SX einige Besonderheiten, z.B. die "Garbage Collection" eine Art interner Hausputz. Während die Rechner arbeiten, produzieren sie immer wieder Datenmüll, den sie aber vor dem Anwender verbergen.

Dieser Müll befindet sich nur im freien RAM (dessen Größe zeigt die eingebaute Funktion MEM an), und auch nur dort findet der Hausputz statt. Normalerweise erkennen die Rechner selbst, wann es Zeit für eine solche Aktion ist, der Anwender merkt es z.B. an folgenden Effekten:

Ein Programm soll zählen und diese Zahlen mit DISP anzeigen. Dabei hält der Rechner zwischendurch an als hätte ihn eine plötzliche Lähmung befallen.

Unangenehm wird dieser Effekt, wenn ein Programm piepsen soll, und zwischen den Piepstönen keine Pause sein darf - die Rechner schaffen es meist trotzdem hier einen Hausputz zu starten. In meinen Morseprogrammen (PRISMA 1/90 für 28S und 4/90 für 48SX) habe ich MEM benutzt, da dieser Befehl die "Garbage Collection" auslöst. Bei der Eingabe über die Tastatur sind Verzögerungen praktisch bedeutungslos, es sei denn, man hat dem 48SX gerade eine neue 128K-Karte spendiert und eingebunden (Befehl MERGE, Abhilfe s. Artikel GRAPHISTO 48).

Eine weitere Eigenart betrifft das Behandeln von Listen (s. auch Matthias Rabe, "Komfortable Listenverarbeitung" in PRISMA 4/90). In seinem Buch "Principles and Programming of the 28C/S" beschreibt Wik-

kes einen einfachen Test:

Die Last-Funktionen ARG (28S: LAST), COMMAND und STACK (28S: UNDO) abschalten, in Ebene 1: folgendes Programm setzen:

```
« 1 50 FOR A A NEXT 50
  →LIST »
```

Dann startet man dieses mit EVAL und holt sich ein Element mit 50 GET so daß nur die Zahl 50 auf dem Stack liegt. Mißt man mit MEM den freien Speicher, führt SWAP DROP aus, dann zeigt ein erneutes MEM die enorme Differenz von 447,5 Bytes, obwohl die Liste ja weder von den Funktionen LAST, COMMAND (CMD) oder STACK (UNDO) gesichert wurde.

Des Rätsels Lösung:

Intern speichert der Rechner trotz GET die komplette Liste. Das Element, welches wir in Ebene 1: sehen, ist eigentlich nur ein Zeiger auf diese Liste. Steht die Liste in irgendeiner Variablen, braucht der Rechner keinen zusätzlichen Platz. Dieser Effekt macht sich nur bemerkbar, wenn die Liste durch GET, GETI oder LIST→ verschwindet und bleibt, bis alle Elemente verschwunden, bearbeitet (Zahlen z.B. quadrieren) oder in eine andere Liste gespeichert wurde.

Die einzige Ausnahme scheinen Objekte mit 2,5 Bytes Länge zu sein, denn würde man die oben angegebene Tastenfolge mit 1...9 GET versuchen, verschwände die Liste sofort mit dem GET aus dem RAM. In den übrigen Fällen sollte man (nennen wir es Isolationsmethode)

```
1 →LIST LIST→ DROP
```

ausführen, wenn man den Speicherplatz braucht. Zusätzlich bietet der 48SX noch den Befehl NEWOB

Wie auch →LIST fertigt er von jedem Objekt eine vollständige Kopie an. Bearbeitet man aber alle Elemente einer Liste, so kann man mit dem heimlich gespeicherten Original leben. Kritisch wird es allerdings, wenn Programme die Listen immer wieder auflösen und dann neu zusammenfassen.

Wickes hat dazu in seinem Buch das Programm SORT und GSORT (Methode QuickSort) vorgestellt. Ähnliches gilt natürlich auch für die Programme MSORT (MergeSort) QuickSort und QuickerSort von M. Rabe aus seinem Artikel "Sortieren" in 5-6/90, weshalb er das Unterprogramm HD (aus seinem Artikel "Listenverarbeitung" in 4/90) verwendet.

Im Stack verwenden beide Rechner folgende Strategie, um Speicherplatz zu sparen:

Holt man ein Objekt mit RCL aus einer Variablen oder dupliziert eines im Stack mit DUP, zeigt MEM nur ein paar Bytes Verlust an, denn der Rechner merkt sich im Stack nur, wo er das eben zurückgeholt Objekt findet. Auch das kann man mit der Isolationsmethode verhindern.

Vorsicht Bug:

Zunächst gibt man

```
{ 2 } und
2 'A' STO A { } +
ein.
```

Es sieht so aus, als ob zwei gleiche Listen in den Ebenen 1+2 stehen. SAME, == und "Ungleich" sehen das nicht so !!

Mißt man jetzt noch deren Platzbedarf (48SX: Befehl BYTES; 28S: Programm LNG aus Pas-de-deux 1, 3/90), so erhält man einmal 7,5 und dann 15,5 Bytes.

Das "Programmers Reference Manual" beschreibt diesen Unterschied beim Befehl BYTES so:

Im ersten Fall enthält die Liste die Zahl 2, die nur 2,5 Bytes braucht, im zweiten Fall kam die 2 aus der Variablen A.

Sinngemäß erläutert, speichert der Rechner hier nicht "Das ist eine Zwei", sondern "Das kommt aus der Variablen A", und diese Referenz kostet 10,5 Bytes.

Das würde allerdings total dem Verhalten widersprechen, welches wir von der Isolationsmethode erwartet haben, nämlich daß das Einfügen eines Objekts in eine Liste immer eine vollständige Kopie dieses Objekts erzeugt. Nach meinen Beobachtungen beschränkt sich dieses Verhalten aber auf die ganzen Zahlen -9...9 und die fünf numerischen Konstanten (e, i, ...), wenn diese in globale Variable gespeichert werden.

Deshalb bin ich der Ansicht:

Der Bug tritt nur bei Zahlen auf, die der Rechner als 2,5 Byte-Objekte speichert. Alle anderen reellen Zahlen brauchen nämlich 10,5 Bytes Platz.

Mißt man den Platzbedarf von A mit

2 'A' STO 'A' BYTES

dann müßte die 2 mit ihren 2,5 Bytes weniger Platz brauchen als z.B.

55 'A' STO 'A' BYTES

weil die 55 ganze 10,5 Bytes verbraucht, aber in beiden Fällen stellt BYTES 16 Bytes fest.

Vermutung daher:

Auch die Zahlen, die sonst nur 2,5 Bytes verbrauchen, werden mit 10,5 Bytes abgespeichert.

Nach dem Rückruf (hier aus A) erfolgt aber eine Umwandlung für die Tests nur dann, wenn die Zahlen frei im Stack stehen, nicht aber für Zahlen in Listen, so daß das hier SAME tatsächlich keine Gleichheit feststellen kann.

Das Verhalten von == und "Ungleich" stellt allerdings noch immer einen Bug dar, denn diese Funktionen erkennen andererseits z.B. die

Gleichheit von 2 und (2;0) an.

Daß die Rechner 2,5-Byte-Zahlen in die normale reelle Form umwandeln kennt man aus den Arrays, dort ist es notwendig, weil diese Objekte auf schnellstmögliche mathematische Bearbeitung getrimmt wurden.

Warum sucht der Rechner aus Listen die Daten langsamer heraus als aus Arrays oder Strings ?

Ganz klar: Soll der Rechner die Nr. 25 aus einem String oder einem Array herausuchen, dann kann er sofort ausrechnen, wo er die Nummer 25 findet. Listen können dagegen beliebige Objekte enthalten.

Der Rechner muß daher alle 24 vorangehenden Objekte auf deren Länge untersuchen, bevor er weiß, wo Nr. 25 beginnt.

Wenn ein selbstgemachtes Programm erst einmal läuft, dann sollte man versuchen, Listen, die Zahlen oder Zeichen enthalten, durch geeignete Strings oder Arrays zu ersetzen.

Neben ihren vielen Funktionen bieten 28S und 48SX auch eine Reihe von Anwenderflags an. Die Frage erhebt sich, ob man diese überhaupt gebrauchen kann.

Im 41er waren sie ja nützliche Gesellen, da testet man an einer Stelle etwas, braucht das Ergebnis erst an einer anderen Stelle. Das Programm löscht dazu das Flag, testet irgendwas, und setzt dann ggf. das Flag:

CF 03

X=Y?

SF 03

Nur falls die Inhalte in Ebene 1: und 2: gleich waren, setzte das Programm Flag 0.

An späterer Stelle kann man das Flag testen und eine Entscheidung treffen.

Auf dem HP-28/48 führt das jedoch zu unnötigem Aufwand, man müßte hier

SAME 3 IF SWAP THEN

SF ELSE CF END

oder auch

3 CF IF SAME THEN

3 SF END

programmieren.

Stattdessen empfehle ich für den Informationsaustausch innerhalb eines Programms einfach das Testergebnis (wie alle Tests verbraucht SAME den Inhalt von Ebene 1: und 2: und schreibt dafür eine 0 oder 1 in Ebene 1:) im Stack oder einer lokalen Variablen aufzubewahren.

Eine solche Anwendung zeigen die Programme BOGEN und KR aus Pas-de-deux-3 (5-6/90). Hier prüfen die Programme zunächst mit IFERR, ob die Variable in Ebene 1: bereits existiert. Falls ja schreiben die Programme deren Inhalt und Namen sowie eine 1 in den Stack. Falls die Variable jedoch nicht existiert schreiben beide Programme nur eine 0 in den Stack.

Das IF...THEN am Schluß prüft dann, ob eine 1 da war, und nun die Variable wiederhergestellt werden muß, oder es findet die 0 und tut garnichts.

Brauchbar hingegen bleiben die Flags beim Informationsaustausch zwischen Programmen, oder wenn der Anwender hier bestimmte Statusinformationen für verschiedene Programme festlegen möchte (der Rechner tut sowas auch, z.B. um festzulegen, ob Winkel im DEG, RAD oder GRAD-Modus verarbeitet werden).

Manche Funktionen verlangen, daß Eingaben in einem bestimmten Winkelformat erfolgen.

Als Beispiel soll hier die Funktion "involut x" dienen, die für die Berechnung von Zahnradern mit Evolventenprofil Verwendung findet, damit die Zähnen nicht infolge von Karies (Pittingbildung) oder Parodontose (mangelnde Zahnfußfestigkeit) erkranken. Die Funktion lautet:

$$\text{inv}(x) = \tan(x) - x;$$

Diese Gleichung gilt nur, wenn man x im Bogenmaß einsetzt.

Als Programm sieht das so aus:

« TAN LASTARG - »

In der Einstellung DEG berechnen

die Rechner zwar den Tangens richtig ziehen aber x nicht im Bogenmaß sondern in Grad ab.

Mein Vorschlag:

« **TAN LAST -1 ACOS / π * »**

Haben wir RAD gewählt, ergibt -1 ACOS gerade π , und π durch π ergibt Eins. Hätten wir DEG gewählt, ergäbe -1 ACOS genau 180° , dadurch geteilt und mit π multipliziert, bekommen wir auf wunderbare Weise ganz automatisch den Umrechnungsfaktor von Grad auf Radian, nämlich $\pi/180^\circ$. Entsprechendes ergibt sich auch bei der Einstellung von GRAD (Neugrad, Gon).

28S und 48SX bieten eine Reihe von **Schleifenstrukturen** an, hier eine kurze Betrachtung:

Alle Schleifenstrukturen werden immer mindestens einmal durchlaufen, egal was die Testbedingungen sagen.

Egal ob FOR, START, DO bis auf WHILE.

Diese bietet als einzige die Möglichkeit einen Programmteil wirklich nur dann auszuführen, wenn die Testbedingungen erfüllt waren. FOR und START laufen dagegen immer mindestens einmal durch !

Für die IF... Strukturen ist die Lage von IF im Programm egal.

Gleiches gilt für UNTIL in DO...; Eigentlich könnte man ohne diese Befehle auskommen, wenn der Rechner nicht immer kleinliche Syntax-Error produzierte. Beim 28S sollte zwischen IF und THEN genau ein Befehl stehen, dann spart der Rechner 5 Bytes ein.

Die Frage, die die ganze Rechnergeneration schon seit Monaten bewegt, nähert sich der Aufklärung:

Wie dupliziere ich am schnellsten ?

Der Rechner kennt die Befehle DUP, DUP2 und DUPN. Ein Programm, welches 3 DUPN statt DUP DUP2 benutzt, verschwendet zwar keinen Platz, dafür aber Zeit. Auch DUP2 DUP2 ist schneller als 4 DUPN.

Hat man ein Objekt in Ebene 1: und möchte, daß es in den Ebenen

1: bis n: steht, dann empfiehlt sich das Programm NDUP.

Es ist zwar nicht kurz, aber sehr schnell und arbeitet wie eine Standardfunktion.

Nur positive Eingaben ($\geq 1,5$) führen überhaupt zu einer Aktion (PICK und ROLL machen ja auch nichts bei negativen Eingaben). Allerdings gibt es für reelle und komplexe Zahlen einen noch einfacheren Weg, wenn man die Funktion CON einsetzt.

Beispiel: Die Zahl 5,2 (steht bereits in Ebene 1:) soll in die Ebenen 1-8 kopiert werden.

Man führt

8 1 →LIST SWAP CON ARRAY→ DROP

aus.

Als kleiner Hinweis der mit der Arbeitsweise von NDUP zusammenhängt noch folgender Tip:

Wie bringe ich eine FOR Schleife dazu, für den Schleifenzähler (hier j) zwischen 1 und n nur Zweierpotenzen (1, 2, 4, 8, 16...) zu erzeugen ?

Na so:

1 n FOR j ... j STEP

Eine Anwendung von NDUP stellt die Betriebssystemerweiterung Con dar.

Während das eingebaute CON in Ebene 2: nur eine Liste mit einem oder zwei Elementen zuläßt, um Vektoren und Matrizen mit einem konstanten Wert zu erzeugen, akzeptiert Con zusätzlich reelle Zahlen in Ebene 2:.

Dann erzeugt es eine Liste mit Konstanten. In Ebene 1: erwartet CON eine reelle oder komplexe Zahl, wenn Con eine Liste erzeugen soll, darf in Ebene 1: sogar jedes beliebige Objekt stehen.

Im folgenden Programmierbeispiel geht es um römische Zahlen.

Allseits beliebt, ob zur Numerierung oder um Filmen die richtige Jahreszahl aufzudrücken, gibt es diese sieben Zeichen:

M = 1000 D = 500

C = 100 L = 50

X = 10 V = 5

I = 1;

Außerdem gibt es noch die Regel,

daß die Zeichen I, X und C den nächsthöheren vorangestellt, und dann von diesen abgezogen werden dürfen, so bedeutet also

CM = 900 CD = 400

XC = 90 XL = 40

IX = 9 IV = 4

Unter den vielen z.T. widersprüchlichen Regeln habe ich diese ausgewählt, weil sie als die klassischen gelten.

Manche Bücher wollen wissen, daß z.B. XM 10000 bedeutet, andere Quellen meinen, daß man seit dem späten Mittelalter gerade dieses Zeichen als 990 deutet, und überhaupt alle kleineren Zeichen vor die größeren gesetzt werden dürfen, so daß IL = 49 (statt XLIX), VM = 995 (statt CMXCV) bedeutet, und in Zweifelsfällen die kürzere gewählt werden soll.

Da mir solche Schreibweisen im täglichen Leben aber bisher nicht aufgefallen sind, gebe ich dafür nur die Programme →ROM3 und →ARAB3 (dieses Programm übersetzt beide hier angegebenen Schreibweisen !) an, deren Bedienung den folgenden Programmen entspricht.

Man kann sich nun Gedanken um einen geeigneten Algorithmus machen, ich verwende einfach folgenden:

☞ Die sieben einzelnen und sechs zusammengesetzten Zeichen nehme ich als 13 selbstständige Zeichen.

☞ Auf der einen Seite habe ich meine eingegebene Zahl, auf der anderen einen zunächst leeren String.

☞ Falls meine Zahl größer als 1000 ist, versuche ich 1000 abzuziehen und addiere zum String ein "M".

☞ Das wiederhole ich solange, bis meine Zahl kleiner als 1000 ist. Falls es geht, versuche ich 900 abzuziehen, und falls mir dies gelingt, addiere ich "CM" zum String.

Gleiches versuche ich dann noch mit 500, 100, 90 u.s.w.

Zur Programmierung nehme ich die Methode Listenauswahl, die ich schon mehrfach vorstellte:

In einer Liste befinden sich die Zahlen 1000, 900, 500... und in der anderen die Strings "M", "CM", "D"...

Die Inhalte beider Listen werden einander durch das Programm zugeordnet. Beginnen wir mit dem Programm →ROM, welches in Ebene 1: eine ganzzahlige positive Zahl erwartet:

```
« "" SWAP
```

Den leeren String füllt das Programm später mit den römischen Zeichen auf.

SWAP bringt dann die eingegebene Zahl in Ebene 1: zurück. Da wir 13 Zeichen definieren, muß die folgende FOR-Schleife von 1 bis 13 laufen und bekommt den Zähler j:

```
1 13 FOR j
```

Als innere Schleife wählen wir die WHILE...REPEAT Struktur. Dazu legen wir dann auch die Liste mit den zugelassenen Zahlen auf den Stack:

```
    WHILE { 1000
900 500 400 100 90
50 40 10 9 5 4 1 }
```

Eine von diesen Zahlen holen wir mit...

```
j GET
```

und anschließend testet

```
DUP2 >
```

ob unsere eingegebene Zahl größer ist als die, welche wir mit j GET aus der Liste holten. Falls ja geht es gleich nach

```
REPEAT
```

weiter, denn zunächst müssen wir unsere eingegebene Zahl mit

```
- SWAP
```

verkleinern und das entsprechende Zeichen aus der Liste mit den Strings holen:

```
{ "M" "CM" "D" "CD" "C"
"XC" "L" "XL" "X" "IX"
"V" "IV" "I" }
```

Legen wir dazu auf den Stack und extrahieren mit

```
j GET
```

das geeignete römische Zeichen um es mit

```
+ SWAP
```

an den in Zeile 1 begonnenen String anzuhängen. Das war's auch

schon, nach dem END muß noch ein DROP folgen, weil diese Stelle ja vom REPEAT aus angesprungen wird.

```
END DROP
```

Die WHILE...REPEAT-Schleife kann sich bei den Ziffern I, X und C bis zu dreimal wiederholen, bei M sogar beliebig häufig. Jetzt noch der Abschluß der FOR Schleife

```
NEXT DROP »
```

- fertig!

Die Rückübersetzung römischer in arabische Zahlen geht genauso:

Wir prüfen den eingegebenen String jeweils, ob am Anfang einer unserer bekannten 13 Strings steht. Dabei hilft uns die Funktion POS, wenn der eingegebene String und z.B. "XC" POS eine 1 ergibt, dann schneidet →ARAB das vordere Ende unserer Eingabe ab und addiert dafür 90 zum zukünftigen Ergebnis. Der Anfang von →ARAB gleicht dem von →RÖM, statt eines leeren Strings legen wir jetzt allerdings eine Null auf den Stack:

```
« 0 SWAP
```

Es folgt wieder die FOR Schleife

```
1 13 FOR j
```

dann folgt nach WHILE die Liste mit den zugelassenen Strings in Ebene 1: ab

```
WHILE { "M" "CM"
"D" "CD" "C" "XC"
"L" "XL" "X" "IX"
"V" "IV" "I" }
```

holen uns dann mit

```
j GET
```

das vom Zähler spezifizierte Zeichen und testen, ob es an Position 1 steht:

```
DUP2 POS 1 ==
```

Falls ja geht es nach REPEAT weiter,

```
REPEAT SIZE 1 +
```

```
OVER SIZE SUB SWAP
```

wobei zunächst ein oder zwei Zeichen unseres eingegebenen Strings abgeschnitten werden, bevor das Programm zunächst deren Gegenwert aus folgender Liste

```
{ 1000 900 500 400
```

```
100 90 50 40 10 9 5
```

```
4 1 } j GET + SWAP
```

mit dem Zähler j auswählt, holt und addiert. Die letzten beiden Zeilen entsprechen wieder dem Programm →ROM:

```
END DROP
```

```
NEXT DROP »
```

Daß die Listings von →ROM2 und →ARAB2 kürzer und schnellere Programme beschreiben sollte diese ausführliche Vorstellung nicht entwerfen. Die optimierten Programme berücksichtigen den Grundsatz, daß der Zugriff auf Strings und Arrays fast immer schneller und platzsparender erfolgen kann als bei Listen.

In Pas-de-deux 2 habe ich einige Sortierprogramme vorgestellt. Als Verfahren diente der Bubble-Sort, welches HP im Handbuch für 28 und 48 verwendet.

Den Algorithmus kann man sich so vorstellen:

Man nehme einen Satz Bleistifte unterschiedlicher Länge, stoße sie (mit dem hinteren Ende) auf und ziehe den Längsten heraus. Das Herausziehen wiederhole man so oft, bis man keinen Bleistift mehr hat, und lege den jeweils gezogenen Stift links neben die vorher Entfernten.

Damals habe ich mich sehr über den Geschwindigkeitszuwachs gefreut, bis mich der Artikel "Sortieren" von Matthias Rabe (PRISMA 5-6/90) aus allen Wolken holte.

Die Rechenzeit bei Bubblesort wächst nämlich quadratisch mit der Anzahl der sortierten Objekte:

Doppelte Anzahl bedeutet vierfache Rechenzeit, dreifache Anzahl ergibt neunfache Rechenzeit !

Seine ausgezeichnete Beschreibung reizte mich aber bald, meine Sortierprogramme zu verbessern. Seiner Beschreibung kann man entnehmen, daß ISORT in ungünstigen Fällen genauso langsam wie ein Bubblesort ist. Die QuickSort und QuickerSort-Algorithmen können so auch zum QuarkSort verkommen,

wenn man die Daten nur ungünstig genug anordnet.

Beweiswütige Mathematiker haben gezeigt, daß ein Sortieralgorithmus, so schlau er auch sein mag, bei (all-)gemein gewählten Mengen etwa $a \cdot n \cdot \text{lb}(n)$ Schritte braucht. Dabei ist $\text{lb}(n)$ der binäre oder Zweierlogarithmus, es gilt

$$\text{lb}(n) = \ln(n)/\ln(2) = \log(n)/\log(2)$$

und a eine beliebige Konstante.

Ein Verfahren, das jede beliebige Datenmenge in nur $a \cdot n$ Schritten bearbeitet, kann es daher nicht geben. Verfahren, die $a \cdot n \cdot \text{lb}(n)$ als Höchstgrenze für ihre Arbeit nehmen, hat Matthias aber auch vorgestellt. Den MergeSort-Algorithmus habe ich nicht vollendet, da Vorversuche zu viel zu aufwendigen Programmen (min. dreifach verschachtelte Schleifenstruktur) führten.

Übrig blieben die "sortieren durch binäres Einfügen"-Algorithmen, welche Matthias mit BISORT und BITSORT bezeichnete.

Nocheinmal kurz das Wesen des binären Einfügens:

Nehmen wir der Einfachheit wegen eine Liste, die die Zahlen 10, 20, 30 ...290 enthält, in die wir die Zahl 93 einfügen wollen.

Zunächst nehmen wir zwei Positionen an, zwischen denen 93 eingefügt werden könnte: Hier wäre es zwischen den Positionen 0 (vor der 10/Pos. 1) und Pos. 30 (nach der 290/Pos. 29).

Im ersten Schritt vergleichen wir 93 mit der Zahl auf Position $(0+30)/2=15$, dem mittleren Element der Liste, und dort finden wir die 150.

Mit einem einfachen Test können wir bereits die Hälfte aller Elemente ausschließen, denn weil 93 kleiner als 150 ist, dürfen wir es zwischen den Positionen 15 und 30 sicher nicht einfügen.

Nun probieren wir den Vergleich von 93 mit dem Element in der Mitte der restlichen Elemente auf Position $(0+15)/2=7$ (Vereinbarung: bei gebrochenen Zahlen runden wir ab!), das ist die 70.

Durch den Vergleich mit 93 stellen

wir fest, daß 93 offenbar zwischen Pos. 7 bis 15 (also zwischen 70 und 150) eingefügt werden muß.

Im nächsten Schritt probieren wir es dann mit Position $(7+15)/2=11$ u.s.w. solange, bis die beiden Positionen sich nur noch um 1 unterscheiden, da muß dann 93 rein. Will man also in n Elemente ein neues Einfügen, dann reichen also $\text{lb}(n)$ Vergleiche aus.

Ein Programm, welches auf dem 28S/48SX schnell läuft sollte aber zwei weitere Bedingungen erfüllen:

Es sollte nur im Stack sortieren, weil dort die schnellsten Möglichkeiten zum Vergleich und Einfügen der Elemente bestehen.

Außerdem sollte es nicht rekursiv sein, da dies zum einen zusätzlich Zeit kostet und außerdem bei jedem Selbstaufruf erneut Speicherplatz reserviert.

Den so definierten Algorithmus repräsentiert das Programm BISORT.

Das Programm sortiert in max. $n \cdot \text{lb}(n)$ Schritten jede Datenmenge, leider bearbeitet es sortierte Datensätze nicht schneller als beliebig sortierte Daten.

Abhilfe verspricht da ein Test, ob man überhaupt dieses Einfügeverfahren anwenden muß.

Man bekommt allerdings für n Elemente $n \cdot \text{lb}(n)$ Tests hinzu, ohne jedoch eine Erfolgsgarantie zu erhalten.

Das Programm BITSORT repräsentiert diesen Algorithmus.

Nach einigem Grübeln kam ich dann auf folgenden Algorithmus, von dem ich allerdings nicht weiß, ob ihn schon jemand beschrieben hat (Bestimmt! - sachdienliche Hinweise bitte an meine Adresse).

Beim einfachen binären Einfügen nehmen wir zuerst das Element aus der Mitte der Liste. Diesen und nur diesen ersten Schritt variiert mein Programm:

Es vergleicht das neue einzufügende Element mit dem zuletzt eingefügten, statt mit dem mittleren.

Gegenüber BITSORT ergeben sich bei meinem PSORT folgende Vorteile:

1. Im Programm enthält BITSORT zwei, PSORT nur einen Test, was die Anpassung des Programms an andere Tests (z.B. zum Sortieren von Daten, Uhrzeiten u.s.w.) erleichtert.

2. Nicht nur sortierte, sondern auch rückwärts sortierte Datensätze können in nur $n-1$ Schritten bearbeitet werden. Ausnahme:

Tauchen im rückwärts sortierten Datensatz mehrfach gleiche Elemente auf, so werden diese nur beim ersten Mal in einem Schritt eingefügt. Die übrigen brauchen jeweils $\text{lb}(n)$ Schritte.

Nimmt man eine Liste, die zunächst das größte, dann das kleinste, dann das zweitgrößte, dann das zweitkleinste u.s.w. Element enthält, dann hat man den "schlimmsten" Fall für PSORT gefunden, aber das Programm bleibt immer unter der $n \cdot \text{lb}(n)$ Grenze. Man kann sich überlegen, daß der schlimmste Fall für PSORT nicht mehr Tests erfordert kann, als der schlimmste Fall für BITSORT.

Die Bedienung der Programme BISORT, BITSORT und PSORT geht so:

In Ebene 1: erwartet das Programm die Anzahl n der zu sortierenden Objekte welche sich dann in Ebene 2: bis $n+1$ aufhalten sollten.

Bei der Optimierung dieser drei Programme fand ich noch einige interessante Details:

Der Rechner arbeitet schneller, wenn man die Funktion PICK mit ganzzahligen Werten füttert, so daß die IP-Befehle (man könnte auch CEIL verwenden, aber dieser Befehl braucht doppelt so lange wie IP) in diesen Programmen kein Luxus sind.

BISORT und BITSORT speichern die Grenzen, zwischen denen das gerade bearbeitete Element eingefügt werden soll, in Ebene 1:+2. Wählt man deren Reihenfolge so, daß die kleinere Zahl in Ebene 2: steht, läuft das Programm schneller als umgekehrt. Außerdem runden

(!) die Stackbefehle wie PICK, ROLL u.s.w. ihre Argumente, 2,5 PICK entspricht also 3 PICK, -5 PICK entspricht 0 PICK (=nix PICK), was einen bedeutender Unterschied zum HP-41C bedeutet, da dieser (für indirekte Aufrufe von Flags oder Datenregistern) immer nur den positiven, ganzzahligen Anteil benutzt.

Natürlich muß man zuweilen auch Listen oder Arrays sortieren. Dazu habe ich das Programm SORT aus Pas-de-deux 2 (4/90) überarbeitet und auf den PSORT-Algorithmus umgerüstet.

SORT akzeptiert folgende Eingaben:

1. Eine Liste in Ebene 1:, die entweder nur reelle, oder Binärzahlen oder Strings enthält.
2. Eine Matrix oder Vektor, welche reelle Zahlen enthalten.
3. Einen 'Namen', der eine Liste oder ein Array enthält.
4. In Ebene 1: die Anzahl der zu sortierenden reellen oder binären Zahlen oder Strings und in Ebene 2: bis n+1 diese Objekte.

Man kann die Programme SORT und PSORT weiter variieren, um sie speziellen Problemen anzupassen.

Dazu muß man die Testbedingung zwischen IF und THEN in Zeile 23 (SORT) bzw. Zeile 6 (PSORT) entsprechend verändern, oder, um Mathias Rabe zu zitieren:

"...Dadurch kann für alle erdenklichen zu sortierenden Daten eine Testroutine geschrieben werden, ohne daß man sich über den Sortieralgorithmus noch Gedanken machen muß...".

Man kann Daten (nur 48SX; Datumsformat MM,TTJJJJ oder TT,MMJJJJ muß mit Flag -42 entsprechend den gegebenen Daten eingestellt werden !) vergleichen, indem man die Testbedingung

DDAYS 0 <

wenn die Daten in aufsteigender,

oder...

DDAYS 0 ≥

wenn in absteigender Reihenfolge sortiert werden soll. Uhrzeiten lassen sich mit folgender Testbeding-

ung aufsteigend sortieren:

HMS- 0 <

oder absteigend so:

HMS- 0 ≥

Probleme mit der Laufzeit entstehen, wenn man einen sehr aufwendigen Vergleichstest braucht.

Nehmen wir folgendes Beispiel:

Auf dem Stack liegen eine Reihe von Listen, die (in dieser Reihenfolge) Name, Vorname, Wohnort, Telefon und Geburtstag verschiedener Personen enthalten.

Diese Listen sollen nun in eine alphabetische Reihenfolge nach Dudenregel gebracht werden.

Dazu muß man zunächst den Nachnamen holen, (z.B. mit DUP 1 GET) dann ein Space anhängen (" ") und zu diesem String den Vornamen addieren (OVER 2 GET +). Jetzt kommt die Umsetzung, denn der 28S/48SX sortiert nach ASCII-Codes (s. Pas-de-deux 2), nach Dudenregel haben Groß- und Kleinbuchstaben jedoch die gleiche Stellung im Alphabet und ä/ö/ü sortiert man wie a/o/u. Läßt man das bei jedem Test durchführen, so muß alleine diese Umsetzung $2*n*b(n)$ durchgeführt werden.

Mit einem kleinen Trick kann man diese für einfache Probleme auf $n*(1+b(n))$ Umsetzungen reduzieren, so wie beim später vorgestellten Programm SORT.

Aber bei komplexen Problemen empfehle ich DUOSORT, denn dort braucht man für jedes Element nur einmal eine Umsetzung durchführen, wobei man das Original (im Beispiel war es eine Liste) kopiert und einen String errechnet, den der Rechner mit den einfachen Vergleichsfunktionen bearbeiten kann. Genaugenommen erfinden wir einen Ersatz für unsere Liste, der sich einfach testen läßt (Beispiel aus dem Leben: Waren erhalten Preise, und die kann man vergleichen).

In Zeile 4 von DUOSORT steht *funktion*, hier kann man ein paar Programmschritte oder einen Unterprogrammaufruf einfügen, welche die Umsetzung erledigen. Diese *funktion* darf den Inhalt von Ebene

1: verbrauchen, und muß ein für die eingebauten Funktionen < vergleichbares Objekt erzeugen.

Als Beispiel kann man hier die Programme DUDEN oder DAT aufrufen.

DUDEN wandelt einen String, der Groß- und Kleinbuchstaben enthält in einen anderen um, bei dem alle Großbuchstaben durch kleine, und ä/ö/ü durch a/o/u ersetzt werden. Für die Umlaute nutzt das Programm eine Eigenart des Zeichensatzes, welche nur auf dem 48SX besteht. Das der 28S im LCD kleine Umlaute nur als Würfel darstellt, reicht es, wenn man für ihn nur die Zeilen 1-4 aus dem Listing von DUDEN abtippt.

Um Sonderzeichen auf der Tastatur des 28S verfügbar zu machen, kann man das CUSTOM-Menü verwenden, für den Doppelpunkt geht das so:

58 CHR { } + MENU

Diese Zeichen fügen sich allerdings in Strings nur mit einem zusätzlichen SPACE links und rechts ein, verzichten aber im Alpha-Eingabemodus auf die "".

DAT hingegen erwartet in Ebene 1: eine Liste, die an erster Stelle Datum und an zweiter Stelle eine Uhrzeit enthält. Solche Listen erhält man auf dem 48SX über RCLALARM. Damit man diese sortieren kann, bestimmt der Rechner den Abstand zu einem beliebigen Datum (vor dem 15.10.1582 geht nicht; hier: 3. Okt. 1990) und addiert dann die mit HMS→ umgewandelte und durch 24 geteilte Uhrzeit.

Die so berechnete Zahl kann man sofort sortieren. Zur 28S-Version gehört das Unterprogramm CJ, welches ich in 4/89 ("Der Zinsling") dokumentiert habe.

DUOSORT erwartet die Anzahl der zu sortierenden Objekte in Ebene 1: und diese selbst in den Ebenen 2 bis n+1 (also wie bei PSORT, BLSORT oder BITSORT). Das Programm macht zwar nur n Umsetzungen, braucht dafür aber $n*n(n)$ Sortier- und weitere n Einfügeschritte, so daß es sich nur für komplizierte Tests eignet.

Eine weitere Anwendung, ist SORT (28S: →STR nach OVER in Zeile 2 und ein CLMF ganz am Ende einfügen), das Programm welches die Statistikmatrix DAT wieder in Ordnung bringt.

Dazu nimmt es sich aus PAR die aktuelle Spalte der x-Werte und sortiert nach diesen die Zeilen von DAT. Ausgestattet mit den Eigenschaften von PSORT ordnet das Programm die Statistikmatrix besonders schnell, wenn an die sortierten Daten neue Werte angehängt wurden. Das Programm funktioniert allerdings nicht, wenn DAT nur einen Spaltenvektor enthält. - holt dann statt einem Vektor mit einem Element nur das Element zurück, was zwar praktisch aber für eine programmierte Anwendung inkonsequent ist.

Zur Erinnerung: SORT hatte ich in Pas-de-deux 2 vorgestellt, damit man aus der Statistikmatrix geordnete Wertetabellen herausholen kann.

Das Programm INSERT fügt ein Element in Ebene 1: in das Objekt in Ebene 2: ein.

Dabei darf in Ebene 2: eine Liste, ein Vektor, der Name einer Variablen, die eine Liste oder einen Vektor enthält, oder eine positive Zahl n stehen, die angibt, daß der Inhalt von Ebene 1: zwischen die Elemente von Ebene 3 bis n+2 eingefügt werden soll.

INSERT verlangt, daß die Daten, zwischen welche eingefügt werden soll, entweder aufwärts oder abwärts sortiert vorliegen. Matrizen konvertiert das Programm automatisch in Vektoren.

Zugelassene Elemente sind reelle und binäre Zahlen, sowie Strings, die nach ASCII-Codes sortiert werden.

In Pas-de-deux 6 habe ich das Programm INTER zur Tabelleninterpolation vorgestellt. Dieses Programm arbeitet nach meinen bisherigen Erkenntnissen einwandfrei, hat allerdings 50 Bytes zuviel. Das -natürlich kostenlose- Update liegt diesem

Artikel ebenfalls bei (Bedienung unverändert), allerdings verbrät das Programm nach wie vor den größten Teil seiner Zeit mit der Newtoninterpolation, während das Abspecken nur beim Suchalgorithmus Erfolg hatte (die Erfahrungen aus den Sortierprogrammen konnte ich vielfältig umsetzen !).

Eine weitere Ergänzung zu den Polynomprogrammen in Pas-de-deux 3 habe ich hier:

Für den 28S veröffentlichte ich bereits in PRISMA 4/88 und 4/89 Programme, die Polynome 2.-4. Grades nach den Formeln von Cardano (der hat außerdem im 16. Jh. die kardanische Aufhängung erfunden) und Ferrari (der hatte leider nichts mit Enzo oder Fiat zu tun, war dafür aber Cardanos Schüler) knacken.

Leider können sie mit der in Pas-de-deux 3 vorgestellten Vektordarstellung nichts anfangen.

Das ändert sich mit dem Programm NGL, denn dieses kann Polynome in Vektoren verarbeiten, löst Gleichungen 1. und 2. Grades mit reellen oder komplexen Koeffizienten sowie Gleichungen 3. und 4. Grades mit reellen Koeffizienten.

In Ebene 1: erwartet NGL dazu das Polynom als Vektor.

Zur Erinnerung:

Der Vektor z.B. eines quadratischen Polynoms enthält an erster Stelle den Koeffizienten von x^2 , an zweiter Stelle den von x und an dritter Stelle den des absoluten Gliedes, also so, wie man es üblicherweise schreibt. Die erste Zahl (der Koeffizient der höchsten Potenz) darf nicht Null sein.

Warum erscheint ein String beim Ausdruck kürzer, wenn man ihn mit **1 DISP PRLCD** statt **PR1** druckt ?

Im LCD haben alle Zeichen mindestens eine Leerspalte Abstand zu ihren Nachbarn, auf dem Drucker aber mindestens zwei. Breite Zei-

chen, wie z.B. A, erscheinen auf dem Drucker jeweils mit einer Leerspalte rechts und links, auf dem LCD bekommen sie nur eine Leerspalte links (oder nur rechts ?).

Da ein Zeichen im LCD immer genau 6 Spalten hat, während es auf dem Drucker eine siebte bekommt, könnte man, zumindest auf dem 48SX, Texte auch als GROBs (mit 6 Spalten pro Zeichen) ausdrucken und so bis zu 27 (statt 24) Zeichen auf den Drucker (166 Spalten Druckbreite) bekommen.

Wahrscheinlich alle, die den IR-Drucker haben, und manche 48SX-Besitzer benutzen Akkus.

Hersteller Panasonic behauptet, daß ein Akku als defekt gilt, wenn er nur noch 80% seiner Nennkapazität hat, und das ist nach etwa 1000 Ladezyklen zu erwarten.

Da eine schonende Ladung etwa 14-15 Std. dauert, braucht man ohnehin einen zweiten Akkusatz, aber wo bewahrt man den auf ?

In Elektronik-Läden findet man Halter für Batterien/Akkus fast jeder Größe, damit man diese in Geräten befestigen und anschließen kann. Ein solches Teil besitzt innen meist vier Fächer für die Akkus und außen zwei Anschlüsse (wie ein 9V-Block) von denen man einen vorsichtig entfernen sollte, damit hier kein Kurzschluß passiert.

Diese praktische Lösung hält alle Akkus fest und kostet etwa 1-2 DM pro Akkusatz.

Am Schluß noch eine kleine Danksagung:

Dieser Artikel entstand auf meinem ersten eigenen Computer. Ohne die Hilfe unserer Regionalgruppe Rheinland/Ruhrgebiet, ohne Jörn und die Computergöttin Martina würde es diesen Artikel immer noch nicht geben. Gegenüber einem 28S oder 48SX ist ein 386SX doch noch ein ziemlich undressiertes Gerät.

Ralf Pfeifer
Rubensstr. 5
5000 Köln 50

```

PFEIFER THU 22.08.91 10:44:06 UMR
48SX → ROM
< " " SWAP 1 13
002 FOR j
    WHILE ( 1000
004 900 500 400 100 90
50 40 10 9 5 4 1 )
006 j GET DUP2 >
008 "M" "C" "D" "CD" "C"
"X" "L" "XL" "X"
"Y" "I" "IV" "I" }
010 "I" } GET + SWAP
012 END DROP
014 > Bytes : 285
Checksum : # E49Dh
    
```

224 →ROM
1: "CCXXIV"

```

PFEIFER THU 22.08.91 10:46:06 UMR
48SX → ARAB
< 0 SWAP 1 13
002 FOR j
    WHILE ( "M"
004 "C" "D" "CD" "C"
"X" "L" "XL" "X"
"Y" "I" "IV" "I" }
006 j GET DUP2 POS 1 ==
008 REPEAT SIZE 1 +
010 OVER SIZE SUB SWAP
1000 900 500 400
100 90 50 40 10 9 5
012 4 1 } GET + SWAP
014 END DROP
016 > Bytes : 300
Checksum : # F428h
    
```

"CCXXIV" →ARAB
1: 224

```

PFEIFER THU 22.08.91 10:49:46 UMR
48SX → ROM2
< SIGN 4 5 9 10 40
50 90 100 400 500
900 1000 " " LASTARG
004 0 12
006 FOR j ROT
    WHILE DUP2 >
008 REPEAT ROT
"MCMD CDC XCL XLX I
XV IVI"
010 j 2 * LASTARG MOD
OVER + SUB + SWAP
012 ROT OVER - SWAP
END DROP
014 > Bytes : 238
Checksum : # 80D7h
    
```

```

PFEIFER THU 22.08.91 10:10:47 UMR
48SX → ARAB2
< DROP 1 4 5 9 10
40 50 90 100 400
500 900 1000 0
004 LASTARG 0 12
006 FOR j
    WHILE
008 "MCMD CDC XCL XLX I
XV IVI"
j 2 * LASTARG MOD
010 OVER + SUB DUP2 POS
1 SAME
012 REPEAT SIZE 1 +
OVER SIZE SUB ROT
014 ROT OVER + ROT
END DROP ROT
016 DROP
018 > Bytes : 255,5
Checksum : # AB0Dh
    
```

```

PFEIFER THU 22.08.91 10:10:47 UMR
48SX → ROM3
< " " SWAP 1 7
002 FOR j 8 j DUP 2
MOD + +
004 FOR k
    WHILE ( 1000
006 500 100 50 10 5 1 0
} DUP j GET SWAP k
008 GET - DUP2 >
REPEAT - SWAP
010 "MDCLXVI" DUP k DUP
SUB SWAP j DUP k DUP
012 + + SWAP
END DROP -1
014 STEP
016 > Bytes : 223,5
Checksum : # 90D6h
    
```

495 →ROM3
1: "VD"

```

PFEIFER THU 22.08.91 10:17:56 UMR
48SX → ARAB3
< 0 SWAP 1000 → k
002 < 1 OVER SIZE
START 1 1 SUB
004 LASTARG + OVER SIZE
SUB ROT ROT ( 1000
500 100 50 10 5 1 1
006 "MDCLXVI" ROT POS
008 GET SWAP
IF OVER k >
010 THEN k 2 ≠ -
END OVER 'k'
012 STO + SWAP
014 > Bytes : 215,5
Checksum : # 1F00h
    
```

"VD" →ARAB3
1: 495

```

PFEIFER THU 22.08.91 10:19:30 UMR
48SX → BISORT
< 2 SWAP
002 FOR j j ROLL 4
LASTARG 4 +
004 DO DUP2 + 2 /
IP SWAP OVER PICK 5
006 PICK
IF >
008 THEN ROT
END DROP
010 UNTIL DUP2 - -1
012 SAME
END DROP 3 -
014 ROLLD
016 > Bytes : 109
Checksum : # C54Fh
    
```

```

PFEIFER THU 22.08.91 10:19:52 UMR
48SX → BITSORT
< 2 SWAP
002 FOR j j PICK
LASTARG PICK
004 IF >
    THEN j ROLL 4
006 LASTARG 3 +
DO DUP2 + 2 /
008 IP SWAP OVER PICK 5
PICK
IF >
010 THEN ROT
END DROP
012 -1 > UNTIL DUP2 -
014 -1 > END DROP 3 -
016 ROLLD
018 > Bytes : 136
Checksum : # 681Dh
    
```

```

PFEIFER THU 22.08.91 10:20:34 UMR
48SX → PSORT
< 1 + 5 3 ROT
002 FOR j j ROLL
LASTARG 3 + ROT 4
004 DO OVER PICK 5
PICK
IF <
006 THEN ROT
END DROP DUP2
008 + 2 / IP SWAP
UNTIL DUP2 SAME
010 END 4 ROLL OVER
012 ROLLD DROP2
NEXT DROP
014 > Bytes : 119
Checksum : # A940h
    
```

```

PFEIFER THU 22.08.91 10:23:45 UMR
48SX → ZSORT
< SPAR 1 GET
"xcol: " OVER + 1
002 DISP NZ → x n
004 < x - 6 7 NZ 6 +
FOR j x - DUP x
006 GET ROT j SWAP 5
DO OVER PICK
008 x GET 5 PICK
IF <
010 THEN ROT
END DROP
012 DUP2 + 2 / IP SWAP
UNTIL DUP2
014 SAME
END 5 ROLL
016 OVER ROLLD DROP2
SWAP DROP
018 NEXT DROP 1 n
START Σ+
020 NEXT
022 > Bytes : 205
Checksum : # 32D2h
    
```

```

PFEIFER THU 22.08.91 10:46:19 UMR
48SX → SORT
<
002 IF DUP TYPE
    THEN DUP
    IFERR RCL
    THEN SIZE → s
006 *
IFERR LIST →
    THEN ARRAY →
008 LIST → DUP2 DROPN *
END SORT s
010 IFERR →LIST
    THEN →ARRAY
END
012 ELSE SORT SWAP
014 >
016 STO
END
018 ELSE 1 + 5 3 ROT
    FOR j j ROLL
020 LASTARG 3 + ROT 4
DO OVER PICK
022 5 PICK
IF <
024 THEN ROT
END DROP
026 DUP2 + 2 / IP SWAP
    UNTIL DUP2
028 SAME
    END 4 ROLL
030 OVER ROLLD DROP2
    NEXT DROP
032 > Bytes : 248
034 Checksum : # 7A52h
    
```

```

1: ( #B3h #AFh #CBh
#E2h #7Fh #EFh )
SORT
1: ( #2Fh #7Fh #AFh
#B3h #CBh #E2h )
    
```

```

PFEIFER THU 22.08.91 10:40:23 UMR
48SX → DUOSORT
< 0 n
< 5 1 n
002 FOR j j 2 *
ROLL DUP funktion
004 ROT j 5 + SWAP 5
WHILE DUP2 ≠
006 REPEAT OVER
PICK 5 PICK
IF <
010 THEN ROT
END DROP
012 DUP2 + 2 / IP SWAP
    END 4 ROLL
014 OVER ROLLD DROP j +
3 - ROT SWAP ROLLD
016 NEXT DROP n
DROPN
018 > Bytes : 183
Checksum : # 3E50h
    
```

```

PFEIFER THU 22.08.91 10:40:53 UMR
48SX → DUDEN
< DUP SIZE " "
002 DO DUP +
UNTIL DUP2 SIZE ≤
004 END 1 ROT SUB OR
1 3
006 FOR j
    WHILE DUP "ääö"
008 j DUP SUB POS
REPEAT LASTARG
010 "aaou" j DUP SUB
REPL
012 END
014 > Bytes : 123
Checksum : # 12B3h
    
```

Beispiel zu DUOSORT mit
DUDEN als Unterprogramm:

```

6: "Köln"
5: "BaChen"
4: "Münster"
3: "MÜNCHEN"
2: "mUtLaNgEn"
1: "müch"
6 DUOSORT
6: "BaChen"
5: "Köln"
4: "müch"
3: "MÜNCHEN"
2: "Münster"
1: "mUtLaNgEn"
    
```

```

PFEIFER THU 22.08.91 10:55:03 UMR
48SX → DAT
< 3,10199 OVER 1
002 GET DDAYS SWAP 2
GET HMS → 24 / +
004 > Bytes : 56
Checksum : # 749Bh
    
```

```

28S → DAT
< DUP 1 GET IP 1
002 LAST FO %? IP LAST
10000 * CJ LAST IFTE
004 SWAP 2 GET HMS → 24 /
+
006 > Bytes : 84
Checksum : # 4C69h
    
```

```

28S → CJ
< SWAP 2,85 - 12 / +
002 DUP 367 * IP OVER IP
1,75 * - ROT + IP
004 SWAP 1 % IP ,75 * -
IP 1721115 +
006 > Bytes : 128
Checksum : # 4B9Fh
    
```

```

28S → ZINTER
< RCL EVAL SIZE 1
002 GET COL? → s n x y
< RCL ( 1 x } GET
004 RCL ( n x } GET
OVER s < OVER s >
006 IF XOR
    THEN DROP2
008 "Nicht in Tabelle" 1
DISP ABORT
010 END > n 1
IF ROT
    THEN SWAP
END
012 DO DUP2 + 2 / IP
SWAP OVER x 2 →LIST
016 RCL SWAP GET s
IF >
    THEN ROT
END DROP
020 UNTIL DUP2 - ABS
1 SAME
022 END MAX 2 - 1
MAX n 3 - MIN DUP 3
024 +
    FOR j RCL ( j x
026 } GET RCL ( j y }
GET
028 NEXT 8 ROLL 7
ROLL 6 ROLL 5 ROLL 4
030 →ARRY 0 x } STO 4
→ARRY 0 1 1 3
032 FOR j DUP 4 PICK
1 GET ≠ ROT + SWAP s
034 x j GET - ≠ ROT 1 4
j -
036 FOR k k DUP2
GET LAST 1 + GET - x
038 k GET LAST j + GET -
/ PUT
040 NEXT ROT ROT
NEXT ROT 1 GET
042 > * + CLMF
> Bytes : 534
044 Checksum : # 7F0Eh
    
```

```

PFEIFER THU 22.08.91 11:02:56 UMR
48SX → ZINTER
< COL ? → s x y
002 < RCL ( 1 x }
GET RCL ( NZ x }
004 GET
IF OVER s <
006 OVER s > XOR
    THEN DROP2 1539
008 DOERE
END
010 IF > NZ 1 ROT
    THEN SWAP
END
012 DO DUP2 + 2 /
IP SWAP OVER x 2
→LIST RCL SWAP GET
016 IF >
    THEN ROT
END DROP
018 UNTIL DUP2 -
1 SAME
020 ABS
END MAX 2 - 1
MAX NZ 3 - MIN DUP
3 +
022 >
    FOR j RCL ( j y
024 x } GET RCL ( j x
} GET
026 NEXT 8 ROLL 7
ROLL 6 ROLL 5 ROLL
4 →ARRY 1 x } STO 4
030 →ARRY 0 1 1 3
032 FOR j DUP 4
PICK 1 GET ≠ ROT +
SWAP s x j GET - *
034 ROT 1 4 j -
    FOR k k DUP2
036 GET LASTARG 1 + GET
- x k GET LASTARG j
+ GET - / PUT
038 NEXT ROT ROT
040 NEXT ROT 1 GET
> * +
042 > Bytes : 493
Checksum : # 3FEBh
    
```

```

283 INSERT
< OVER
002 IF TYPE
THEN OVER
004 THEN RCL
THEN TYPE 5
006 THEN SAME
THEN + LIST
008 DUP → 1
010 INSERT 1 → LIST
012 ELSE SWAP
ARRY → LIST → DUP2
014 DROPN * DUP → a
016 ROLL INSERT a 1 +
018 END
020 SWAP STO
END
022 ELSE 4 ROT 5 + 4
PICK OVER 1 - PICK
024 IF <
THEN SWAP
END
026 DO DUP2 + 2 / IP
028 SWAP OVER PICK 5
PICK
030 IF >
THEN ROT
END DROP
032 UNTIL DUP2 - ABS
034 1 SAME
END MAX 4 -
036 ROLLD
END
038 > Bytes : 316,5
Checksum : # C9D2h
1: [ 0 2 4 6 8 ]
1: [ 0 2 4 6 7 8 ]
.PFEIFER THU 22.08.91 11:41:20 UMR
48SX NGL
<
002 IF DUP SIZE 1 GET
OVER TYPE 3 * SWAP
004 > AND
THEN 1281 DOERR
END DUP 1 GET /
ARRY → LIST →
008 < SWAP - 2 / DUP
SQ ROT - 1 +
010 LASTARG - IM
LASTARG DUP RE IFTE
012 SWAP IM LASTARG DUP
RE IFTE
014 > ROT ROT → a b
016 < a DUP SQ DUP
+ b 9 * - * 27 / +
018 2 / b 3 * a SQ - 9
/ OVER SQ OVER 3 ^
020 +
IF DUP ARG
THEN DROP NEG
022 SWAP OVER 1,5 ^
NEG ACOS 3 / COS
024 SWAP 1 * DUP +
ELSE SWAP
026 DROP 1 - LASTARG +
3 XROOT SWAP 3
XROOT + NEG
030 END a 3 / -
032 LASTARG 2 * + DUP2
* b +
> n x y
< NEG x
y EVAL x
038 > 4 ROLLD → b c
< 4 / DUP SQ
040 6 * b - OVER DUP
DUP b OVER SQ 3 * -
042 * c - * 5 ROLL +
SWAP DUP + DUP SQ b
044 - * c +
IF DUP
THEN → r q
< 2 / r NEG
048 DUP2 * a SQ 8 / - y
EVAL DROP2 DUP SQ r
050 - 1 q OVER / 2 / ROT
052 ROT - LAST + 3 PICK
NEG
054 EVAL
END SWAP x
056 EVAL 5 PICK - 5
ROLLD 4 PICK - 4
ROLLD x EVAL 3 PICK
058 - SWAP ROT -
> n GET EVAL
060 EVAL n → LIST
> SWAP DROP
062 > Bytes : 807
Checksum : # F4EFh
285 SORT
< XPAR 1 GET
002 "XCOL: " OVER → STR +
1 DISP NZ → x n
004 < 2 - 6 7 NZ 6 +
FOR j 5 - DUP x
006 GET ROT j SWAP 5
008 GET 5 PICK
IF <
THEN ROT
END DROP
010 DUP2 + 2 / IP SWAP
UNTIL DUP2
014 SAME
END 5 ROLL
016 OVER ROLLD DROP2
SWAP DROP
018 NEXT DROP 1 n
START Σ+
NEXT CLMF
020 > Bytes : 210
Checksum : # D357h
.PFEIFER THU 22.08.91 12:45:33 UMR
48SX NDUP
< DUP → n
002 <
IF 1,5 ≥
THEN 1 n 2 /
FOR j j DUPN
006 LASTARG
STEP n
008 LASTARG 2 * - DUPN
END
010 > Bytes : 98
012 Checksum : # 721Ch

```

```

Nullstellen gesucht:
f(x)=2x^3-6x^2+2x-6
1: [ 2 -6 2 -6 ]
1: ( 3 (0;-1) (0;1) )
285 NGL
< DUP SIZE 1 GET
002 OVER TYPE 3 * OVER 3
> AND SWAP 5 >
004 IF OR
THEN INV
END DUP 1 GET /
006 ARRY → LIST →
008 < SWAP - 2 / DUP SQ
ROT - 1 + LAST -
010 >
ROT ROT → a b
012 < a DUP SQ DUP +
b 9 * - * 27 / + 2 /
014 b 3 * a SQ - 9 /
OVER SQ OVER 3 ^ +
016 DUP
IF ARG
THEN DROP NEG
018 DUP 1,5 ^ ROT SWAP /
NEG ACOS 3 / COS
SWAP 1 * DUP +
022 ELSE SWAP DROP
1 - LAST + 1 2
START SIGN
024 LAST ABS 3 INV ^ *
SWAP
NEXT + NEG
END a 3 / -
028 LAST 2 * + DUP2 * b
030 +
> n x y
< NEG x
y EVAL x
034 >
4 ROLLD → b c
< 4 / DUP SQ 6
038 * b - OVER DUP DUP b
OVER SQ 3 * - * c -
040 * 5 ROLL + SWAP DUP
+ DUP SQ b - * c +
042 >
IF DUP
THEN → r q
< 2 / r NEG
046 DUP2 * a SQ 8 / - y
EVAL DROP2 DUP SQ r
048 - 1 q OVER / 2 / ROT
ROT - LAST + 3 PICK
050 NEG
>
ELSE ROT ROT x
052 EVAL
END SWAP x
054 EVAL 5 PICK - 5
ROLLD 4 PICK - 4
ROLLD x EVAL 3 PICK
058 - SWAP ROT -
> n GET EVAL
060 EVAL n → LIST
> SWAP DROP
062 > Bytes : 807
Checksum : # F4EFh
285 SORT
< XPAR 1 GET
002 "XCOL: " OVER → STR +
1 DISP NZ → x n
004 < 2 - 6 7 NZ 6 +
FOR j 5 - DUP x
006 GET ROT j SWAP 5
008 GET 5 PICK
IF <
THEN ROT
END DROP
010 DUP2 + 2 / IP SWAP
UNTIL DUP2
014 SAME
END 5 ROLL
016 OVER ROLLD DROP2
SWAP DROP
018 NEXT DROP 1 n
START Σ+
NEXT CLMF
020 > Bytes : 210
Checksum : # D357h
.PFEIFER THU 22.08.91 12:45:33 UMR
48SX NDUP
< DUP → n
002 <
IF 1,5 ≥
THEN 1 n 2 /
FOR j j DUPN
006 LASTARG
STEP n
008 LASTARG 2 * - DUPN
END
010 > Bytes : 98
012 Checksum : # 721Ch

```

```

# 105Fh 3 NDUP
3: # 105Fh
2: # 105Fh
1: # 105Fh
.PFEIFER THU 22.08.91 12:49:32 UMR
48SX Con
002 IF OVER TYPE
THEN CON
ELSE SWAP DUP → n
< NDUP n → LIST
006 >
END
008 > Bytes : 64
Checksum : # E05h
Die vorangegangenen
Programme gehören
zum Artikel
Pas de deux Teil 7
***
Die nachfolgenden
Programme gehören
zum Artikel
GRAPHISTO - 48
.PFEIFER THU 22.08.91 12:34:16 UMR
48SX ALMBSP
< RCLALARM LASTARG
002 DELALARM LIST → ROLL
28 DATE+
004 IF DUP 8 <
THEN 7 DATE+
END 4 ROLLD 4
006 → LIST STOALARM DROP
"Ortsgruppentreffen"
010 DUP 1 DISP -9 9
START 1 WAIT 2000
012 , 1 BEEP 1000 , 1
BEEP
014 IF KEY
THEN DROP2 MAXR
ELSE 1
END
016 STEP
018 > Bytes : 193
020 Checksum : # A78Fh
.PFEIFER THU 22.08.91 12:35:43 UMR
48SX SQPL
< # a7h # 41h PDIM
002 PPAR' EVAL LASTARG
OVER LIST → DUP2
004 DROPN DUP ROT - C→R
ROT C→R DUP R→C
006 SWAP DUP R→C 7 ROLL
EVAL SWAP 8 / CEIL
008 > x y v u s z
< CR 1 s
FOR a s a - "a"
OVER + " " DUP
+ DUP + + SWAP 1 +
" " + 26 OVER SIZE
012 - SWAP REPL PR1 u a
DUP 1 - R→C s / x *
014 - C→R 1 z
FOR b DUP2 v
b DUP 1 - R→C z / y
* - C→R ROT SWAP
R→C PMAX R→C PMIN
020 DRAW DRAX
IF a b + 2
022 MOD
THEN PPAR
LIST → DUP2 DROPN -3
RND PMAX -3 RND
024 PMIN LABEL
END PICT (
# 0h # 1h ) ( # A5h
# 40h ) SUB PR1
030 DROP ERASE
NEXT DROP2
032 PR1 DROP
NEXT STO
034 > PICT PURGE
> Bytes : 518
Checksum : # 687Bh
.PFEIFER THU 22.08.91 12:36:00 UMR
48SX GNOT
< 3 DUPN SUB NEG
002 SWAP DROP REPL
> Bytes : 27,5
004 Checksum : # C9A6h

```

```

.PFEIFER THU 22.08.91 12:29:34 UMR
48SX GLOT
< EVAL # A5h ROT 8
* R→B # A7h OVER
PDIM 1 - 2 → LIST
002 PPAR SWAP OVER
LIST → DUP2 DROPN
004 DUP2 OVER - RE 7
PICK / SWAP RE ROT
006 IM R→C → 1 u x o
< 1 ROT
FOR j "a" j 1 -
+ " " DUP + DUP
+ + j → " " + 26
010 OVER SIZE - SWAP
REPL PR1 j DUP 1 -
R→C x * C→R u + DUP
012 PMIN SWAP o + DUP
PMAX DRAW DRAX -3
014 RND PMAX -3 RND
PMIN LABEL PICT RCL
( # 0h # 0h ) 1 SUB
016 PR1 DROP CR PR1
018 DROP ERASE
NEXT PPAR' STO
024 > PICT PURGE
> Bytes : 384,5
026 Checksum : # 5232h
.PFEIFER THU 22.08.91 12:36:18 UMR
48SX GPACK
< DUP SIZE DROP B→R
002 8 / CEIL 2 * → b
< → STR 1 3
004 " " POS SUB LASTARG
006 + OVER SIZE SUB
NEXT b "0"
008 DO DUP + DUP2
UNTIL SIZE <
END 1 ROT SUB
010 SWAP
WHILE DUP2 SWAP
012 POS 1 SAME
REPEAT b 1 +
014 OVER SIZE SUB
END
WHILE DUP2 DUP
016 SIZE DUP b 1 +
SWAP SUB SAME
018 REPEAT 1 OVER
SIZE b - SUB
END ROT ROT
020 DROP2 DUP SIZE b /
" " + SWAP + + +
022 > STR → DUP DUP
SIZE # 1h OVER
BLANK ROT B→R 0
024 OVER 1 - → h o b j
k
<
DO j R→B # 0h 2
032 → LIST o REPL
UNTIL DUP2 *
034 'j' INCR b ≥ OR
END DROP DUP
DO k R→B # 0h 2
036 → LIST o REPL
UNTIL DUP2 *
038 'k' DECR j ≤ OR
END DROP j 1 -
R→B # 0h 2 → LIST k
1 + R→B h 2 → LIST
040 SUB
042 > Bytes : 531,5
044 Checksum : # D8CCh
.PFEIFER THU 22.08.91 12:32:50 UMR
48SX IDISP
<
IF DUP 0 ≤
THEN LCD → NEG
002 → LCD NEG DISP LCD →
NEG → LCD
ELSE DISP
END
006 > Bytes : 55
008 Checksum : # B7F4h
.PFEIFER THU 22.08.91 12:32:12 UMR
48SX GAND
< NEG ROT NEG ROT
002 ROT GOR NEG
> Bytes : 27,5
004 Checksum : # EC38h
.PFEIFER THU 22.08.91 12:32:53 UMR
48SX BALLAST
< 0: LEER DUP
002 PURGE MEM 12000 - 8
/ IP
IF DUP 9 >
004 THEN 1 → LIST 0
CON SWAP STO
ELSE DROP2
006 END
> Bytes : 90
010 Checksum : # BECh

```

Graphisto

Der Teufel steckt im Detail Grafikobjekte auf dem HP48 von Ralf Pfeifer

Die Grafikobjekte (kurz GROB) des 48SX erscheinen mir ebenso neu wie unausgegoren.

Das fängt schon mit den Funktionen an:

AND, OR, XOR und NOT wirken nicht auf GROBs, weshalb sich sicher mancher schon gefragt hat, wie man ein GROB invertiert, also in Negativdarstellung umschaltet.

Die Lösung:

Die Funktion NEG machts, was das Handbuch aber verschweigt.

Die Funktion + überlagert darüberhinaus zwei gleichgroße GROBs mit einem logischen OR, doch das war's schon mit der eingebauten Hilfe.

Das Rätsel des Platzbedarfs bei GROBs konnte ich inzwischen auch lösen:

Steht in der Anzeige GROB b*h (b=Breite, h=Höhe in Pixeln), dann braucht das Objekt $10 + \text{CEIL}(b/8) * h$ Bytes (CEIL ist eine eingebaute Funktion).

Aus dieser Formel erkennt man, daß Grafikobjekte mit z.B. 131 Pixel Breite genausoviel Platz brauchen, wie ein Objekt mit 136 oder 129 Spalten. Der 48SX speichert ein GROB nämlich zeilenweise und faßt je 8 Pixel immer zu einem Byte zusammen. Falls das nicht ganz aufgeht, läßt er im letzten Byte jeder Zeile einfach ein paar Bits ungenutzt. Also kann man bei einem GROB die Breite immer so wählen, daß sie sich durch 8 teilen läßt.

Diese Regel gilt auch bei der Umwandlung eines GROB mit →STR. Hier bilden jeweils $2 * \text{CEIL}(b/8)$ Hex-Zeichen eine Zeile, wobei die Hex-Zeichen von links nach rechts die Pixel im GROB von links nach rechts beschreiben.

Ärger gibt es auch, wenn ein GROB

mit →LCD angezeigt werden soll, da die Funktion →LCD nämlich die Menüzeile übernimmt, so daß von einem GROB höchstens die Höhe von 56 Zeilen angezeigt wird.

Abhilfe: GROB mit

PICT STO GRAPH

anzeigen, wobei allerdings das Fadenkreuz der interaktiven Grafikumgebung auftaucht (so kann man dann auch durch größere GROBs rollen) oder mit

**PICT STO { #0 #0 }
PVIEW 7 FREEZE.**

Wer sich über die TEXT-Funktion ärgert, weil sie in einem laufenden Programm nicht das Stackdisplay aktualisiert, dem kann auch nicht geholfen werden, denn TEXT soll nur die Rückkehr vom Grafik-Modus zum normalen Stackdisplay ermöglichen, den Erfolg zeigt diese Funktion aber erst, nachdem ein Programm angehalten hat, und auch nur dann, wenn FREEZE nicht dazwischenfunke.

Eine echte Abhilfe kenne ich nicht, aber beim HP-28S hat HP für das gleiche Problem eine SYSEVAL-Adresse angegeben. Das sollte man endlich auch für den 48SX erwägen.

FREEZE friert das Display dauerhaft ein, ein DISP-Befehl hebt die Wirkung von FREEZE nicht auf.

Man kann FREEZE an den Anfang eines Programms setzen und dann mehrfach DISP ausführen. Zwar ändert DISP den Inhalt des LCD, aber nach dem Ende des Programms bleibt der mit FREEZE eingefrorene Teil des LCD auch weiterhin eingefroren - mit dem, was das letzte DISP erzeugt hat.

Die Funktion ARC zeichnet Kreise. Mal abgesehen von ihrer beneidenswerten Ruhe, die nur noch vom Equation-Writer übertroffen werden kann, hängt die Interpretation der Winkel in Ebene 1: und 2: vom eingestellten Winkelformat ab.

Um einen Viertelkreis zu zeichnen, muß man in Ebene 1: eine 0 und in Ebene 2 eine 90° , $\pi/2$ oder 100 schreiben.

Für ein Programm ist das ziemlich blöd, wenn es eine bestimmte geometrische Figur aufs LCD bringen soll, aber vorher immer nach dem Winkelmodus sehen muß.

Abhilfe:

Nehmen wir an, daß wir einen Teilkreis von 45° bis 240° zeichnen wollen.

Dazu dividieren wir die Winkel durch 180° , es ist also $45/180=0,25$ und $240/180=1,333\dots$

Diese beiden Zahlen fassen wir zu der komplexen Zahl (0,25;1,333333333333) zusammen, schreiben diese ins Programm und lassen dort die Schritte

...-1 ACOS * C→R...
folgen.

Damit haben wir die gewünschten Winkel korrekt in Ebene 1:+2. Um mit dieser Methode einen Vollkreis zu zeichnen nimmt man:

(0;2) -1 ACOS * C→R

(dieser Spezialfall läßt sich außerdem noch kürzer als

0 2 -1 ACOS *

programmieren. Spezielle Winkel kann man außerdem mit folgender Strategie erzeugen:

180° ergibt sich als -1 ACOS, 270° bekommt man mit -1 ASIN (das Ergebnis lautet zwar -90° , $-\pi/2$ oder -

100, je nach Winkelmodus, aber der Rechner versteht es schon als 270°), und 90° kann man wahlweise mit 0 ACOS oder 1 ASIN erzeugen.

In GROBs kann man einen Punkt entweder durch benutzerdefinierte Einheiten angeben oder durch die Pixelnummer in Binärzahlen.

Für viele Anwendungen wäre es einfach und nützlich wenn der FOR Befehl als Schleifenparameter diese Binärzahlen akzeptieren würde, zumal sie sich einfacher bearbeiten lassen als reelle Schleifenparameter. Leider muß man im Reellen zählen und dann in der Schleife jedesmal den Zähler in eine binäre Zahl umrechnen.

Die Systemvariable PICT enthält die mit den Funktionen des PLOT-Menüs gezeichneten Grafiken. Bemerkenswerterweise kann man PICT löschen (spart Platz), sobald man jedoch ein GROB, egal wie klein, in PICT speichert, vergrößert es dessen Dimension auf mindestens 131 Pixel Breite und 64 Pixel Höhe.

Der alte IR-Drucker kann einige der HP-48SX ASCII-Zeichen nicht ausdrucken, und gibt stattdessen Schachbretter aus.

Um das zu vermeiden kann man vor einen PR1-Befehl die Befehle 2 →GROB einfügen, dann stellt der Drucker alles exakt so wie in der Anzeige dar, was zu folgenden Einschränkungen führt:

Zwischen zwei Zeichen auf dem Drucker liegen zwei Leerspalten, im LCD nur eine, der Ausdruck erscheint daher trotz gleicher Zeichenbreite kompakter.

Im LCD gibt es keine Fettschrift, so daß ein GROB entweder nur normale oder nur fette Zeichen erzeugen kann, und schließlich dauert die Übertragung zum Drucker länger, und belastet die Batterien des 48SX stärker.

Gegen die vielen kleinen anderen Übel, die dem Benutzer von Grafikobjekten drohen, helfen vielleicht die folgenden Programme.

GAND überlagert zwei GROBs mit einem logischen AND, d.h. im überlagerten Bereich bleiben nur noch die Pixel dunkel, die in beiden

GROBs dunkel waren. Für die Überlagerung zweier Kurven bedeutet das, daß nur die Schnittpunkte und Achsen übrig bleiben (gleiches PPAR beim Erzeugen der Kurven-GROBs vorausgesetzt).

GAND erwartet die gleichen Eingaben wie die eingebauten Funktionen **GOR** und **GXOR**:

In Ebene 3: das Ziel-GROBs oder die Systemvariable **PICT**, in Ebene 2: den Punkt im Ziel-GROB, an dem die Überlagerung beginnen soll (in Benutzer- oder Pixelkoordinaten), und in Ebene 1: das GROB, dessen linke obere Ecke an den Punkt aus Ebene 2 angelegt wird.

Im Gegensatz zu **NEG** invertiert **GNOT** nur ein genau definiertes Rechteck in einem GROB. Die Eingaben zu **GNOT** entsprechen der eingebauten Funktion **SUB**:

In Ebene 3: erwartet **GNOT** das GROB oder **PICT**, in Ebene 2: und 1 sucht es die linke obere und die rechte untere Ecke des Rechtecks im GROB, das invertiert werden soll. Die Eckpunkte können entweder in Benutzer- oder Pixelkoordinaten eingegeben werden.

GPACK packt Grafische Objekte. Die Anwendung zielt vorallem auf die GROBs, die der Equation-Writer erzeugt.

Nimmt man z.B.

'SIN(X)' 0 →GROB

so erzeugt man damit ein GROB 131*56, was deutlich zu groß ist.

GPACK erwartet ein solches GROB in Ebene 1: und entfernt alle Leerzeilen oben und unten, sowie alle Leerspalten rechts und links. Das Beispiel verkleinert sich damit in ein GROB 33*11 und spart 93% Platz !

Das Programm **IDISP** zeigt Objekte wahlweise positiv oder in Negativdarstellung an. Es erwartet die Eingaben wie die eingebaute **DISP**-Funktion:

Ein beliebiges Objekt in Ebene 2: und eine Zahl 1..7 für die normale Darstellung (exakt wie **DISP**) oder eine Zahl -1..-7 für die inverse Darstellung. Allerdings blinkt die Anzeige ein bißchen.

Wenn man die Variable **PPAR** richtig

einstellt, kann man beliebig große Funktionsplots erstellen, sofern der Speicher reicht.

Die Grenze von 131*64 Pixel stellt nur die minimale Pixelzahl dar. Man könnte auch einen Plot mit 1000*1000 Pixeln erstellen (für dieses GROB alleine geht dann eine 128K Karte drauf) und automatisch ausdrucken.

Das Problem bleibt aber der Speicher, da der 48SX zuerst den gesamten Funktionsplot anfertigt und dann erst ausdrückt.

Andererseits hat der Infrarotdrucker nur eine Druckbreite von 166 Pixeln. Es wäre also enorm platzsparend, wenn man statt einem GROB 1000*1000 dieses in sechs GROB 166*1000 und ein GROB 4*1000 aufteilt, so daß die kleinen GROBs jeweils nur $\frac{1}{6}$ des Speichers verbrauchen, weil sie sofort gedruckt und wieder gelöscht werden.

Diesen Schritt erledigt mein Programm **GPLOT**. Es ist auch kaum langsamer als das eingebaute **DRAW**, weil die Funktionen in **EQ** nicht häufiger als beim eingebauten Plotter ausgewertet werden.

Zur Anwendung von **GPLOT** läßt man zunächst einmal die Funktionen mit den ganz normalen Funktionen im Display plotten (mit 131*64 Pixeln). Dazu kann man Autoscaling oder Zoom verwenden; wichtig ist nur, daß das LCD genau das zeigt, was später (in anderen Längen- und Breitenverhältnissen) zu Papier kommen soll, und daß die Äste der Funktionen das LCD möglichst optimal ausnutzen.

Bei den Funktionstypen gibt es keine Beschränkungen, außerdem kann man **RES** und **CNCT** (im **MODES**-Menü, verbindet die geplotteten Pixel zu einer Kurve) verändern.

Erst wenn alles wunschgemäß erscheint, verläßt man das Plot-Menü und legt in Ebene 1: eine Liste ab, die Anzahl der Zeilen und die Anzahl der zu druckenden Streifen enthält. Wenn man dann den IR-Drucker eingeschaltet hat, kann man **GPLOT** starten.

Wer trotz **GPLOT** immer noch zuwenig Speicher, aber zuviel Zeit hat, der kann sich mit **SQPL** behelfen.

Nach den gleichen Vorarbeiten und den gleichen Eingaben kann man alternativ zu GPLOT auch SQPL starten, welches ein Grafikobjekt der Größe GROB 166*64 erzeugt.

Einen kleinen Nachteil haben die beiden Programme noch:

Haben in der Variablen PPAR sowohl *axes* als auch *pmax* den gleichen Wert, zeichnen die Programme die Achsen nicht, weil GPLOT immer die äußerste rechte Pixelspalte und SQPL zusätzlich noch die oberste Pixelzeile abschneiden. Außerdem wenden die Programme immer die eingebaute Funktion LABEL an, um die Achsen zu beschriften, durch eine besondere Strategie verwenden sie aber nur die ersten drei Stellen der Zahlenwerte und vermeiden zu häufiges beschriften (damit die Übersicht erhalten bleibt).

Nach soviel Grafik noch ein paar andere Punkte. So mancher hat vielleicht schon gehört, daß der Rechner mit einer 32K-Karte schneller als mit einer 128K-Karte sein soll.

An diesem Gerücht ist etwas wahres dran:

Der Rechner räumt von Zeit zu Zeit sein RAM auf ("Garbage Collection"). Passiert das, während man etwas über die Tastatur eingibt, scheint er einen Augenblick lang gelähmt zu sein.

Dieser Hausputz dauert um so länger, je mehr freies RAM verfügbar ist. Hat man also gerade eine 128K Karte mit MERGE eingebunden, so braucht der 48SX auch viermal mehr Zeit als bei einer neuen 32K Karte. Deshalb aber die kleinere Karte zu kaufen, bei der jedes Bit etwa doppelt soviel kostet, wie bei der 128er Karte, wäre Quatsch - oder Luxus.

Mein Programm BALLAST erzeugt nämlich das Backup-Objekt :0:LEER (also in Port 0), welches das freie RAM verkleinert.

Die Idee: Ein riesiger Vektor (mit lauter Nullen gefüllt) belegt den größten Teil des RAMs, so daß dort der Hausputz nicht stattfindet.

Das Programm BALLAST löscht zunächst :0:LEER, mißt dann mit MEM den freien RAM-Bereich und erzeugt dann ein neues :0:LEER, welches

so groß ist, daß nur noch 12000 Bytes (steht in Zeile 2 von BALLAST, beliebig veränderbar) freies RAM übrigbleiben. Wenn man BALLAST im HOME-Directory ablegt, eignet sich das Ganze außerdem zur Automatisierung:

Man nehme einen Steueralarm, speichere einen beliebigen Tag (z.B. den nächsten Sonntag), eine beliebige Uhrzeit (z.B. 4:00 früh, weil dann keiner den Rechner braucht), setze ein beliebiges Wiederholungsintervall (z.B. wöchentliche Ausführung) und speichere mit EXEC dieses Programm:

```
" DROP BALLAST OFF "
```

Jetzt überprüft der Rechner wöchentlich, ob der freie RAM-Bereich gerade 12000 Bytes umfaßt - ganz automatisch.

Beim Thema Alarme fragt man sich vielleicht auch folgendes:

Die Regionalgruppe Rhein/Ruhr trifft sich an jedem zweiten Sonntag eines Monats in Essen, aber wie bringe ich den 48SX dazu, einen wiederholenden Alarm zu setzen ?

Dazu braucht man zunächst einmal die Funktion RCLALARM, welche einen Alarm zurückruft. Ein solcher Alarm ist immer eine Liste mit 4 Elementen, auf Platz eins finden wir Datum, auf Platz 2 die Uhrzeit des Alarms und auf Platz 4 das Wiederholungsintervall (0=keine Wiederholung). Steht auf Platz 3 ein String, dann haben wir einen Meldealarm, der automatisch piepst, und falls man ihn ignoriert, weist der Rechner bei jedem Einschalten darauf hin, daß man ihn vergessen hat.

Jedes andere Objekt auf Platz 3 verwandelt den Alarm in einen Steueralarm, der versucht, dieses Objekt mit EVAL zu starten.

Nur solche Steueralarme kann man dazu veranlassen, sich in unregelmäßigen Abständen zu wiederholen. Jedesmal, wenn der HP-48SX einen Steueralarm startet, schreibt er dessen Nummer in Ebene 1, so daß sich der Alarm selbst mit der Funktion RCLALARM (sie verbraucht den Wert in Ebene 1:) auf den Stack kopieren kann.

Vorsicht Bug: Aus mir nicht erklärli-

chen Gründen bleibt der Alarm nach RCLALARM im Rechner gespeichert!

Ohne RCLALARM löscht der Rechner den laufenden Alarm einwandfrei.

Doch gehen wir das Programm ALMBSP (Alarm- Beispiel) gemeinsam durch:

Zunächst beginne ich mit einem RCLALARM, um mit der Zahl in Ebene 1: den gerade ablaufenden Alarm noch einmal als komplette Alarmliste auf den Stack zu legen.

Das LASTARG DELALARM löscht den laufenden Alarm für die Zukunft aus dem Alarmkatalog (wegen dem BUG). Die Kopie des Alarms auf dem Stack kann dann das Alarmprogramm bearbeiten, wozu es zunächst die Liste mit LIST→ auflöst. Dieser Befehl läßt die Größe der Liste auf dem Stack, aber das bringt mir keine neue Information, denn in hier muß in Ebene 1: immer eine 4 stehen. Also verfüttere ich diese überflüssige 4 an ein ROLL, welches mir das Datum aus Ebene 5 in Ebene 1: bringt. Das neue Alarmdatum muß mindestens vier und höchstens fünf Wochen nach dem aktuellen liegen. Das kann man sich so überlegen:

Der zweite Sonntag muß zwischen dem 8. und 14. Tag einschließlich liegen. Im kürzesten Fall liegt er auf dem 14. und der Monat hat nur 28 Tage, so daß nur 28 Tage vergehen, bis der nächste Sonntag zwischen dem 8.-14. liegt. Im längsten Fall fällt der zweite Sonntag auf dem 8. und der Monat hat 31 Tage, so daß 35 Tage bis zum nächsten zweiten Sonntag vergehen. Wir addieren also mit

```
28 DATE+
```

vier Wochen, und prüfen dann mit

```
IF DUP 8 <
```

(Preisfrage für ganz Schlaue:

Warum würde $7 \leq$ hier ganz sicher in die Hose gehen ? Der CCD verlost das Standardwerk für Action-Demos "Der große autonome Pflastersteinatlas" von Haßmann & Kappe) ob das errechnete Datum vor dem 8. liegt. Falls ja, müssen wir mit

```
THEN 7 DATE+ END
```

eine weitere Woche addieren. Jetzt rollen wir das neue Datum mit 4 ROLLD an seinen alten Platz im Stack zurück. Nun haben wir in Ebene 1: das Wiederholungsintervall, welches 0 sein sollte.

Um das sicherzustellen, führen wir TYPE aus, denn die Alarmliste stammt ja vom Rechner, und der gibt uns garantiert nur ein zulässiges RPT-Intervall zurück, welches eine reelle Zahl sein muß.

TYPE verwandelt diese dann in eine 0. Mit 4 →LIST fassen wir die vier Einzeldaten im Stack wieder zu einer korrekten Alarmliste zusammen, die STOALARM wieder der automatischen Alarmverwaltung des 48SX übergibt.

Dabei ersetzt STOALARM die Liste in Ebene 1: durch die Nummer, die der Alarm im Alarmkatalog bekommen hat, aber die brauchen wir nicht, weshalb ein DROP folgt. Was jetzt im Programm **ALMBSP** noch folgt, soll nur dem Anwender die Meldung zu Auge und Ohr bringen.

Bei dieser Art von Manipulationen sollte man sich vor zwei Fehlern hüten:

1. Das neue Datum sollte nicht mit Hilfe des Datums der eingebauten Uhr (mit DATE) berechnet werden, sondern aus dem Alarm im Stack. Es könnte nämlich sein, daß der Steueralarm verspätet ausgeführt wurde, weil eine andere Funktion vorher abließ, und den Alarm verhinderte.
2. Man sollte das Wiederholungsintervall (Platz 4 in der Liste) auf Null setzen (haben wir mit TYPE gemacht). Das verhindert, daß der Alarm durch die eingebaute Alarmverwaltung zu früh gestartet wird.

Strings kann man auf dem 48SX auf zwei Arten eingeben: Mit den ""-Zeichen oder mit C\$.

Letztere Methode ist dann notwendig, wenn die "-Zeichen Bestandteil des Strings werden sollen.

Auf dem 48SX kann man "" entweder als 34 CHR oder als C\$ 1 " eingeben.

Interessant wird die Frage, was C\$ 0 ergibt !?

Den Nullstring ("") der dann auch

nur 2,5 Bytes braucht. Das bleibt auch in einem Programm so, während die Eingabe "" über die Tastatur 5 Bytes braucht. Freude hat macht diese Ersparnis allerdings nicht lange, denn aus 2,5 Bytes werden sofort wieder 5, sobald man dieses Programm erneut mit EDIT/VISIT bearbeitet und mit ENTER abschließt.

Die Funktionen IP und FP wirken nur auf reelle Zahlen, um ihre Wirkung auf komplexe Zahlen und Arrays zu erweitern, muß man

O TRNC (für **IP**)

oder für **FP**

DUP O TRNC -
programmieren.

Eine Hilfe der geheimen Art besitzt die ENTER-Taste.

Da während der Eingabe von Programmen weder der Matrix- noch der Equation-Writer aufgerufen werden können, bewirken diese Funktionen ein DUP.

Wie nützlich die Eingabehilfen des PRG BRANCH Menüs sein können, habe ich erst bemerkt, als ich meine gesamten Programme verloren habe und mit der Hand eintasten mußte. In diesem Menü kann man nämlich nicht nur den IF-Befehl sondern gleichzeitig auch THEN, ELSE und END eingeben, mit nur zwei Tastendrücken.

Eine weitere Hilfe bietet das CST-Menü. Setzt man die Variable CST ins HOME-Menü, so kann man überall und jederzeit auf sie zugreifen.

Die Funktionen, die ich am häufigsten eintaste habe ich so belegt: Auf Menütaste A liegen DUP, DUP2 und DUPN, auf B findet man LIST→, ARRAY→ und STR→, auf C liegt nur OVER, auf D ROT, E führt SIZE, SUB und POS aus und SAME liegt auf F. Die zweiten und dritten Funktionen müssen mit der Shift-Taste angewählt werden.

Das Problem, Programme und Daten vor Verlust zu schützen konnte ich für mich noch nicht zufriedenstellend lösen. Das Problem ist bekannt, Dr. G. Heilmann hat es inzwischen

mehrfach beschrieben: Befindet man sich im VARS-Menü und kombiniert man die orangefarbene Vortaste mit einer der Menütasten, so speichert man den Inhalt von Ebene 1: in die entsprechende Variable.

Wer jetzt einen panischen Anfall erleidet und die Übersicht verliert, hat auch schon verloren.

Der erste Tastendruck muß jetzt nämlich LASTARG lauten, denn eine sinnvolle Ausnahme bei STO (auch über die orangefarbene Vortaste) und PURGE sorgt dafür, daß diese Funktionen nicht wie sonst den letzten Stackinhalt retten (wäre ja auch quatsch, den hat man ohnehin in die Variable gespeichert), sondern PURGE und STO retten den gelöschten Inhalt einer Variablen, also das, was eigentlich verloren wäre.

Dieser alte Inhalt hält sich aber nur bis zur nächsten Operation, schließlich heißt die Funktion LAST Argument.

Um den Ärger ganz zu vermeiden, könnte man den Menütasten mit ASN Funktionen zuweisen. Die oben genannten Funktionen ROT, DUP ließen sich dann mit der orangefarbenen Vortaste und einer Menütaste erreichen.

Dazu müßte der USER-Modus permanent eingeschaltet bleiben, bei mir ging er jedoch häufig aus, weil ich entweder eine Funktion brauchte, die ich an einer anderen Stelle des Tastenfeldes mit ASN überschrieben habe, und dann vergaß, USER wieder einzuschalten, oder weil ein RESET stattfand (automatisch nach Ein- und Ausbau von Karten, außerdem erforderlich bei Systemtests oder Untersuchungen mit dem RAM-ROM-Editor.

Dieses Verhalten ist kein Bug, weil man nämlich mit einigen Tastendrücken das Tastenfeld im USER-Modus abschalten kann, und dann auch nicht mehr aus dem USER rauskommt). Ruft man dann eine der undefinierten Funktionen auf (z.B. ROT, hier muß der Stack Daten enthalten), dann provoziert man geradezu die Löschung der Daten. Man könnte die Daten auch in PORT0 sichern.

Nachteil: In Port 0 existiert keine Menüstruktur, man könnte hier bloß einige Programme auslagern, die ohnehin im HOME-Directory stehen.

Will man ein Programm in Port 0 von einem anderen Programm aus starten, muß man z.B. :0:PRP EVAL oder :&:PRP EVAL nehmen, wenn PRP aber im HOME-Directory steht, reicht einfach PRP, was außerdem noch 7 Bytes spart.

Wer ganz sicher sein will und eine RAM-Karte besitzt, könnte folgenden Weg gehen:

Man speichert alle Programme in PORT1 und stellt die Karte mit dem kleinen Schalter auf "nur-lesen" um. Im VARS-Menü ersetzt man dann die Programme durch den Aufruf von Port 1. Beispiel: Das Programm PRP kommt zuerst in Port 1:

```
'PRP' RCL :1:PRP STO
```

dann speichert man das Programm

```
« :&:PRP EVAL »
```

```
'PRP' STO
```

Ob man & oder 1 verwendet, ist egal, & hat den Nachteil, daß eine Endlosschleife beginnt, wenn sich PRP in keinem Port befindet und 1 führt dazu, daß die RAM-Karte mit PRP nur in Port 1 stecken darf.

Auch wenn im freien RAM jetzt viel Platz frei bleibt, gefällt mir auch diese Lösung nicht, denn man verschwendet trotzdem viel Speicher, weil die Karte die Programme enthält, aber zusätzlich noch von den kleinen Programmen im VARS-Menü der Arbeitsspeicher belegt wird.

Als Vorteil bleibt, daß man diese RAM-Karte gefahrlos ziehen kann, weil sie nicht eingebunden, und nicht vom Rechner gelöscht werden kann. Speichert man außerdem mit ARCHIVE noch das VARS-Menü auf der Karte, so kann man nur mit dieser Karte bewaffnet jeden 48SX mit einigen Tastendrücken zum Ebenbild des eigenen Rechners klonen.

Der Weisheit letzten Schluß könnten allerdings die MCODE-Programmierer präsentieren:

Wenn Bibliotheken so funktionieren, wie im Handbuch beschrieben, und sich die eigenen Programmsammlungen entsprechend in Bibliotheken verwandeln ließen (bisher fand ich

noch keine entsprechende eingebaute Funktion), wäre das Problem des Programmschutzes erledigt.

Unter der Überschrift "Der HP-48SX als Barcodeleser - Ein Denkanstoß" habe ich in PRISMA 91.1.18 Überlegungen vorgestellt, wie man fast ohne Materialaufwand und MCODE-Software dem 48SX Barcodes (z.B. für den Programmaustausch in PRISMA) zugänglich machen könnte. Die Überwältigende Flut von Briefen (einer) bewies mir die immense Wichtigkeit dieses Problems. Stefan von Borries wies mich nämlich auf eine Barcodebibel hin.

Nach deren Lektüre weiß ich nun immerhin ein paar Dinge:

Barcodes sind genormt, das ganz normale Standardisierungschaos der Computerbranche (z.B. hunderte von Druckern, die alle drucken, aber nur über eine individuelle Emulation verfügen) scheint vermeidbar.

Außerdem verwenden professionelle Anwender Infrarotbarcodeleser, so daß auch hier ausreichende Erfahrungen vorliegen. Wer also noch Ideen hat, soll sich bei mir melden - oder mit W&W ein Modul produzieren.

Zum Abschluß noch zwei kleine Gimmicks:

Dreht man den Rechner um, so sieht man vier Gummifüße.

Die sind nicht verklebt, sondern klemmen einfach nur in ihrer Nut.

Entfernt man sie (nach ein wenig Übung geht es ohne Werkzeug), erscheint in jeder Nut ein kleines rundes Loch.

Das in der Nut oben rechts ist außerdem mit dem Buchstaben R gekennzeichnet, was wohl auf "RESET-Schalter" hindeuten soll.

Drückt man hier mit einer aufgebogenen Büroklammer hinein, dann schaltet sich der Rechner aus und gelangt dabei in den "Coma Mode", eine besonders energiesparende Abschaltung, die auch die Uhr anhält!

Nach dem Einschalten führt der Rechner einen Reset (Warmstart) durch, dabei bleibt der Inhalt des VARS-Menüs (Programme, Daten u.s.w.) erhalten, nur der Stack, einige Flags und die LAST-Funktionen

(ARG, STACK, MENU und CMD) werden gelöscht.

Beim Einbau hat HP sicher nicht nur an MCODE-Programmierer gedacht, denn es gibt ganz normale Anwendungen:

Läuft ein Programm ab, dann reagiert der Rechner während eines DRAW oder IFERR-Befehls nicht auf die ON-Taste. Das Programm läuft zwar langsamer, hält aber nicht an. Gerade bei zeitaufwendigen Programmen wie GPLOT oder SQPL ist dieser Schalter dann die einzige Möglichkeit, den Rechner anzuhalten, wobei sich die gleiche Wirkung wie bei einem Systemhalt mit ON und C, vgl. Handbuch Anhang A, einstellt.

Die Notwendigkeit des mechanischen Resets in Programmen kann man durch die folgende Maßnahme verringern:

Man fügt eine Tastaturabfrage ein (zwischen IFERR und THEN bzw. zwischen verschiedenen Aufrufen der DRAW-Funktion), denn alle Tastendrücke außer ON landen im Tastaturpuffer. An einer geeigneten Stelle fragt man z.B. mit

```
IF KEY THEN DROP ... END
```

ab, ob der Anwender das Programm unterbrechen möchte. Anstelle der Punkte fügt man dann noch weitere Schritte ein, die z.B. Stack aufräumen oder Variable retten, und einen Programmstop (HALT, KILL oder 0 DOERR) verursachen.

Den Effekt des mechanischen Resets bekommt man aber auch über die Tastatur:

Rechner einschalten, ON drücken, SPC drücken, ON loslassen, dann SPC loslassen, und schon entschläft das Gerätchen mit einem sanften Pieps.

Egal welche Methode man wählt, der Befehl WSLOG dokumentiert den Zeitpunkt des Aufwachens (weil die Uhr stehenblieb ist das auch der Zeitpunkt des Einschlafens), ON SPC löscht allerdings frühere Ereignisse aus der WSLOG-Liste (sonst merkt sich WSLOG die letzten vier Warmstarts und deren Gründe).

Ralf Pfeifer (116)

Rubensstr.5

5000 Köln 50

RS232-Schnittstellenpaket für den HP48

Für den HP48SX bzw. HP48S gibt es Schnittstellenpaket für die serielle Schnittstelle, die neben dem eigentlichen Kabel auch einige sehr nützliche Programme enthält.

Es trägt die Nummer HP82208A und kostet runde 100,-DM.

Das Handbuch hat sich HP hier leider gespart, die Dokumentationen für die einzelnen Programme befinden sich auf den beiden mitgelieferten Disketten, eine 3 1/2" und eine 5 1/4" Diskette identischen Inhalts.

Wie die meisten bin ich erst einmal daran gegangen alle diese Dateien auszudrucken und hatte nacher einen fürchterlichen Papiersalat auf dem Tisch liegen.

Aus diesem Grunde habe ich mich entschlossen die Inhalte der Anleitungsdateien in kompakter Form zum Nachschlagen im PRISMA zu präsentieren.

Ich habe nichts an den Dateien geändert, sie dienen hiermit auch als Information für diejenigen, die noch unschlüssig darüber sind, ob sie sich dieses Schnittstellenpaket zulegen sollen oder nicht. Ich persönlich meine, daß es sich für jeden lohnt, auch wenn man auf den PC oder ATARI oder MAC eines Freundes angewiesen ist.

Eine Sicherheitskopie des gesamten Rechnerinhalts ist ja auch immer möglich, die letzte Rettung für ein MEMORY LOST.

I README

Einführung in das Schnittstellenpaket HP 82208A Version 1.0

I.1 Anmerkung

Dieses Programm wird in der jeweils aktuellsten Fassung ausgeliefert und kann ohne Vorankündigung geändert werden. Die Hewlett-Packard GmbH bietet keinerlei Gewährleistung im Zusammenhang mit diesem Programm. Sie haftet nicht für Fehler oder zufällige Schäden bzw. Folgeschäden im Zusammenhang mit der Lieferung, der Lei-

stung oder dem Einsatz dieses Programmes.

Garantieinformationen über Hardware-Zubehör finden Sie in dem Handbuch des Taschenrechners.

Copyright (c) 1990, Hewlett-Packard GmbH. Alle Rechte vorbehalten. Die Hewlett-Packard GmbH erteilt Ihnen hiermit das Recht, jedes einzelne Programm, das zum Lieferumfang gehört, auf Taschenrechnern von Hewlett-Packard ein zu setzen.

Copyright (c) 1989, Trustees of Columbia University in the City of New York. Hiermit wird jedem bzw. jeder Institution die Erlaubnis erteilt, das Programm Kermit einzusetzen, zu kopieren und weiterzugeben, sofern das Programm nicht mit Gewinnabsichten verkauft wird und sofern dieser Hinweis auf das Urheberrecht übernommen wird.

MS-DOS ist ein in den Vereinigten Staaten eingetragenes Warenzeichen der Microsoft Corporation.

I.2 Lieferumfang Ihres Schnittstellenpaketes

Zum Lieferumfang des Schnittstellenpaketes gehören ein Kabel sowie die Programme, mit denen Sie Dateien zwischen Ihrem Taschenrechner HP 48 und Ihrem Computer übertragen können. Außerdem haben wir einige hilfreiche Anwendungsprogramme hinzugefügt.

In dieser Online-Dokumentation werden normale Tasten des HP 48 durch [] und Menütasten durch { } dargestellt.

Die folgende Übersicht gibt Ihnen zu jedem der zum Schnittstellenpaket gehörenden Programme eine kurze Erläuterung:

KERMIT.EXE

Übertragungsprotokoll für die Dateiübertragung zwischen einem MS-DOS(R)-Computer und einem HP 48. (Kermit muß, bevor eine Übertragung möglich ist, auch auf dem Computer gestartet werden.)

GROB2TIF.EXE

Ein MS-DOS-Programm, mit dem Grafikobjekte des HP 48 in das Dateiformat TIFF (Tag Image File Format) konvertiert werden können.

APPT

Ein Terminkalender für Ihren Taschenrechner.

USAG

Ein Taschenrechner-Programm, das Ihnen zeigt, wie Sie bei der Anwendung eines bestimmten HP-48-Befehls vorgehen müssen.

CLK

Ein Taschenrechner-Programm, mit dem Sie die Systemuhr des Taschenrechners einstellen können.

INPRT

Ein Taschenrechner-Programm, mit dem der HP 48 Daten empfangen kann, die ein anderer HP-Taschenrechner über die Infrarot-Schnittstelle ausgibt.

STPWATCH.LIB

Eine HP-48-Bibliothek, mit der Sie Ihren Taschenrechner als Stoppuhr einsetzen können.

PCLPRINT.LIB

Eine HP-48-Bibliothek, mit der Sie Grafikobjekte auf einem PCL-Drucker wie z.B. HP DeskJet ausdrucken können.

EPSPRINT.LIB

Eine HP-48-Bibliothek, mit der Sie Grafikobjekte auf einem Epson-Drucker wie z.B. dem FX-85 ausdrucken können.

I.3 Einsatz dieser Anwendungsprogramme

Zu jedem Programm gehört eine Dokumentations-Datei mit der Dateierweiterung .TXT (zum Beispiel EPSPRINT.TXT, INPRT.TXT und KERMIT.TXT).

Um mit einem Programm vertraut zu werden, müssen Sie die entsprechende .TXT-Datei lesen und die dort aufgeführten Anweisungen beachten. Sie können die Dateien natürlich auch ausdrucken.

In den Erläuterungen in den .TXT-Dateien wird davon ausgegangen, da Sie bereits mit der Terminologie und den Prozeduren des HP 48 vertraut sind. Es kann deshalb geschehen, daß Sie wegen bestimmter Details im Handbuch des HP 48 nachschlagen müssen. Da Sie für die Dateiübertragung zwischen Ihrem Computer und Ihrem HP 48 auf Kermit angewiesen sind, empfiehlt es sich, mit der Datei KERMIT.TXT zu beginnen.

1.4 Anschlussbelegung des Schnittstellenkabels

Das zum Lieferumfang gehörende Kabel ist für den Anschluß an MS-DOS-Computer bzw. ATARI-ST/TT ausgelegt. Falls Sie das Kabel für den Anschluß an einen anderen Rechner oder Drucker verwenden möchten, benötigen Sie unter Umständen einen Adapter. In der folgenden Tabelle ist die Pin-Belegung des 9-poligen Steckverbinders und des 25-poligen Adapters dargestellt.

HP 48	IBM 9-PIN	IBM 25-PIN
RX (Input) ←	3 TX (Output)	2 TX (Output)
TX (Output) →	2 RX (Input)	3 RX (Input)
SGND	5 GND	7 GND
SHIELD GND	SHIELD	SHIELD

1.5 Mailbox für Anwender von HP-Taschenrechnern

Für den Austausch von Programmen und Informationen zwischen Anwendern, Entwicklern und Händlern von HP-Taschenrechnern wurde eine spezielle Mailbox eingerichtet. Die Übertragungsparameter dieser Mailbox sind 300/1200/2400 Baud, Vollduplex, keine Parität, 8 Bit, 1 Stoppbit. Die Rufnummer der Mailbox ist (in den USA) 503-750-4448. Bei der Nutzung der Mailbox entstehen keine Benutzungsgebühren. Sie müssen lediglich die Telefontkosten zahlen.

Das Benutzer-Handbuch "MS-DOS KERMIT USER GUIDE" ((c) Columbia University in the City of New York, 1981, 1988) kann als Datei aus der Mailbox abgerufen werden. (Das Handbuch ist nur auf englisch vorhanden.)

Anmerkungen:

Für weiterführende Informationen über das Kermit-Protokoll können Sie das Buch von Frank da Cruz, "KERMIT, A File Transfer Protocol" (1987. Bedford, MA: Digital Press) bestellen.

Sie können den Inhalt dieser Datei ausdrucken und haben dann gleich eine gute Übersicht über das Produkt.

Und nun wünschen wir Ihnen viel Spaß mit Ihrem leistungsfähigen HP 48!

II KERMIT

II.1 Überblick

Mit dem Datei-Übertragungsprotokoll Kermit können Sie Dateien zwischen Ihrem Taschenrechner HP 48 und einem Computer übertragen. Kermit ist bereits im HP 48 eingebaut. Sobald Sie Kermit auch in Ihren Computer geladen haben, können der HP48 und Ihr Computer miteinander kommunizieren.

Das Kermit-Protokoll wurde durch

das "Center for Computing Activities" an der Columbia University entwickelt. Kermit ist für eine Vielzahl von Systemen verfügbar und kann für eine geringe Gebühr bei der Columbia University oder bei einer Vielzahl von Benutzergruppen bestellt werden.

II.2 Prozedur

Zur Durchführung der folgenden Schritte müssen Sie unter Umständen in den Handbüchern Ihres Taschenrechners oder Computers nachschlagen.

1. Schließen Sie den 4-poligen Steckverbinder des seriellen Schnittstellenkabels an Ihren HP 48 an. Schließen Sie dann die 9-polige Steckverbindung des Schnittstellenkabels an die Schnittstelle COM1 oder COM2 Ihres Computers an. Dazu ist unter Umständen der 25-polige Adapter erforderlich.
2. Starten Sie das Programm Kermit auf Ihrem Computer. (Dazu müssen Sie in dem Verzeichnis, in dem Kermit installiert ist, auf

der DOS-Befehlsebene den Befehl KERMIT eingeben und die Taste [ENTER] betätigen.)

3. Konfigurieren Sie Kermit so, daß auch die Schnittstelle, an die Sie gemäß Schritt 1 das Schnittstellenkabel angeschlossen haben, angesteuert wird. Außerdem muß die Kermit-Baudrate an die Baudrate Ihres Computers angepaßt werden. (Für diese Einstellung können Sie den Kermit-Befehl SET verwenden, der im folgenden Beispiel erläutert ist.)

Anmerkung:

Die Standard-Baudrate für den HP 48 beträgt 9600. Sie müssen also sowohl Ihren Computer als auch Kermit ebenfalls auf 9600 Baud einstellen.

4. Überprüfen Sie, ob die Baudrate des HP 48 der Einstellung gemäß Schritt 3 entspricht. Außerdem muß die Übertragungsart IR/wire auf "wire" (Leitung) eingestellt werden. Beide Parameter können Sie mit Hilfe des I/O-SETUP-Menüs des HP 48 überprüfen und ändern.
5. Beachten Sie die Anweisungen im Benutzer-Handbuch des HP 48 für die Datenübertragung zwischen einem HP 48 und einem Computer.
6. Nach Beendigung betätigen Sie zum Verlassen des Programmes Kermit die Tasten Q und [ENTER].

II.3 KERMIT-Hilfen

Für Hilfe-Informationen über Kermit auf Ihrem Computer können Sie die Tasten (?) und [ENTER] betätigen. Kermit zeigt dann eine Befehlsliste an. Für Hilfe-Informationen über einen bestimmten Befehl müssen Sie den Befehl, ein Leerzeichen und ein Fragezeichen eingeben und dann die Taste [ENTER] betätigen. Durch Eingabe von "SET ?" [ENTER] erhalten Sie zum Beispiel Hilfe-Informationen über den Befehl SET.

II.4 Beispiel

Bei diesem Beispiel wird vorausgesetzt, daß Sie das serielle Schnittstellenkabel bereits sowohl an Ihren

Taschenrechner als auch an die COM1-Schnittstelle Ihres Computers angeschlossen haben und daß sich im VAR-Menü Ihres HP 48 eine Variable mit der Bezeichnung DATA1 befindet. Außerdem wird vorausgesetzt, daß die Baudrate des HP 48 und des Computers auf 9600 eingestellt ist und daß sie sämtliche Anweisungen gemäß dem Benutzer-Handbuch des HP 48 zur Vorbereitung einer Dateiübertragung durchgeführt haben.

Durch Betätigung der folgenden Tasten können Sie die Datei DATA1 vom HP 48 zum Computer übertragen:

Tastenfolge auf dem Computer:

KERMIT [ENTER] Kermit wird gestartet.
SET PORT 1 COM1 wird eingestellt.
SET BAUD 9600 Kermit wird auf eine Baudrate von 9600 eingestellt und so an den Taschenrechner und Computer angepaßt.
RECEIVE [ENTER] Der Computer wartet auf den Eingang einer Datei.

Tastenfolge auf dem Taschenrechner:

[VAR] [] {DATA1} Holt die zu übertragende Variable in den Stack.
[linke Umschalttaste] Aktivierung des E/A-Menüs und
[I/O] {SEND} Übertragung der Datei zum Computer.
 Die Datei DATA1 wird in das Verzeichnis kopiert, in dem auch Kermit gespeichert ist.

Nun können Sie Kermit verlassen und die Datei DATA1 auf Ihrem Computer bearbeiten. Mit den folgenden Tastatureingaben können Sie die Datei anschließend zum HP 48 zurück übertragen:

Tastenfolge auf dem Computer:

KERMIT [ENTER] Kermit wird gestartet.
SET PORT 1
SET BAUD 9600
SEND DATA1 [ENTER] Übertragung der bearbeiteten Datei DATA1 zum HP 48, wo sie im VAR-Menü angezeigt wird.

(Falls die alte Datei DATA1 noch besteht, erhält der Name der neuen Datei DATA1 automatisch eine Erweiterung, damit die alte Version nicht überschrieben wird)

Tastenfolge auf dem Taschenrechner:

[linke Umschalttaste] Der Taschenrechner HP 48
[I/O] {RECV} wartet auf den Eingang einer Datei.

II.5 Wichtige Hinweise

o Mit dem Kermit-Befehl PUSH können Sie Kermit zeitweilig verlassen, um auf der DOS-Ebene Befehle auszuführen. Durch Eingabe des Befehls EXIT und Betätigung der Taste [ENTER] können Sie wieder in das Programm Kermit zurückkehren.

o Durch Eingabe des Kermit-Befehls STATUS erscheinen auf dem Bildschirm Statusinformationen über die aktuellen Kermit-Einstellungen. Am Anfang dieser Informationen stehen die Baudrate und die eingestellte Schnittstelle.

o Der DOS-Befehl DIR für die Anzeige des Verzeichnisses kann auch von Kermit aus durch Eingabe von DIR und Betätigung der Taste [ENTER] ausgeführt werden.

o Für weiterführende Informationen über das Kermit-Protokoll können Sie das Buch von Frank

da Cruz, "KERMIT, A File Transfer Protocol" (1987. Bedford, MA: Digital Press) bestellen.

II.6 KERMIT auf ATARI ST/TT

Ich benutze auf verschiedenen Rechnermodellen (ST/STE/TT) die alte Version 1.0 des GEM-Kermit von Jörg W. Lohse von 1986 für die Datenübertragung zum HP48SX. Es

lief bislang auch problemlos auf den 19" Großmonitoren SM194 und TTM194, ob es Probleme mit Farbmonitoren gibt ist mir bislang nicht bekannt geworden.

III USAG

III.1 Überblick

Das Programm USAG (Abkürzung für usage = Anwendung) gibt Ihnen Hinweise für die Anwendung der im HP 48 eingebauten Befehle. Sie können sich insbesondere die für einen Befehl zulässigen Objekttypen und die Belegung des Stack durch die Argumente des Befehls anschauen.

III.2 Prozedur

Und so arbeiten Sie mit USAG:

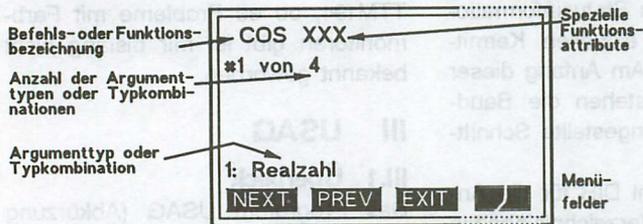
1. Übertragen Sie die USAG-Datei von Ihrem Computer zum HP 48. Im VAR-Menü Ihres HP 48 erscheint das Menüfeld {USAG}.
2. Geben Sie in die Ebene 1 des Stacks eine Liste mit einem Befehlsnamen ein.
3. Betätigen Sie die Menütaste {USAG}. In der Anzeige des HP 48 erscheinen Informationen über den Befehl, den Sie in Schritt 2 eingegeben haben.

III.3 Beispiel

Nachdem Sie das Programm USAG in Ihren HP 48 geladen haben, eine Liste mit dem Befehl COS in Ebene 1 eingegeben und anschließend im VAR-Menü die Taste {USAG} betätigt haben, erscheint die in Bild 1 zu sehende Anzeige.

Dieser Anzeige können Sie als erstes entnehmen, daß COS eine Funktion ist (alle Befehle sind entweder RPN-Befehle oder Funktionen). Nach der Kennzeichnung "(Funct.)" in Ihrer Anzeige erscheinen drei weitere Zusatzzeichen (hier durch "xxx" dargestellt), die die folgende Bedeutung haben: ein Pfeil nach unten (1), der angibt, daß für diese Funktion eine Umkehrung existiert, (2) ein Ableitungssymbol, welches angibt, daß für diese Funktion eine Ableitung existiert und (3) ein Integralsymbol, welches angibt, daß diese Funktion integriert

Bild 1: Kennzeichnung als Befehl oder Funktion



werden kann. Diese Zeichen erscheinen nicht, wenn der angezeigte Befehl nicht über die entsprechenden Eigenschaften verfügt.

In der zweiten Zeile der Anzeige erscheint die Information "#1 von 4". Dies bedeutet, da es insgesamt 4 Kombinationsmöglichkeiten für Argumente der Funktion COS gibt und daß derzeit die erste Möglichkeit angezeigt wird. Bestimmte Befehle erlauben, wie COS, nur wenige Argumenttypen oder Kombinationen von Argumenttypen; andere Funktionen oder Befehle erlauben sehr viel mehr.

In den folgenden Zeilen der USAG-Anzeige stehen die Argumenttypen, die zusammen mit diesem Befehl möglich sind, sowie ihre jeweiligen Stack-Ebenen. Die COS-Funktion bezieht sich hier auf eine reelle Zahl in der Ebene 1. Die COS-Funktion verarbeitet also nur ein Argument. Bei Befehlen mit mehr Argumenten werden die einzelnen Argumente in den für sie richtigen Stack-Ebenen angezeigt. Die Argumentbezeichnungen entsprechen den spezifischen HP-48-Objekttypen. (Ausnahmen sind: "Any" - dies bedeutet, daß sämtliche Objekttypen zulässig sind; "Symbolic" - dies bedeutet, daß das Argument ein algebraisches Objekt, ein globaler Name oder ein lokaler Name sein kann; und "PICT".

Mit den Menütasten im unteren Bereich der Anzeige können Sie vorwärts ({NEXT}) und rückwärts ({PREV}) durch die verschiedenen Kombinationsmöglichkeiten der Argumenttypen blättern. Sobald Sie die Tasten {NEXT} und {PREV} betätigen, wird die zweite Zeile der Anzeige geändert. Zum Beispiel erscheint als nächstes "#2 von 4", dann "#3 von 4" usw. Wenn Sie

nach der letzten Kombinationsmöglichkeit weiterblättern (wieder zur ersten Möglichkeit) oder wenn Sie rückwärts von der ersten zur letzten Möglichkeit blättern, ertönt ein akustisches

Signal. Wenn Sie zur normalen Stack-Anzeige zurückkehren wollen, betätigen Sie die Taste {EXIT}. (Sie können das Programm auch durch Betätigen der Taste [ATTN] verlassen.)

Wenn Sie sich mit dem Programm USAG über einen Befehl informieren, für den kein Argument oder nur ein Argument möglich ist, erscheinen die Menüfelder {NEXT}, {PREV} und {EXIT} nicht in der Anzeige, so daß Ihre aktuellen Menüfelder aktiv bleiben.

IV APPT

IV.1 Überblick

Das Programm APPT ist ein Terminkalender für Ihren HP 48. Sie können durch die verschiedenen Termine, die im Kalenderformat auf dem HP 48 dargestellt werden, blättern und Termine einsehen, hinzufügen, ändern, löschen bzw. Terminvereinbarungen oder Erinnerungen für einen ausgewählten Tag bestätigen.

(Dieses Programm sollten Sie nicht für die Eingabe von Terminen, auch Steuerungstermine genannt, verwenden. Näheres dazu im Benutzerhandbuch des HP 48SX.)

IV.2 Prozedur

1. Mit Hilfe des Übertragungsprogrammes Kermit müssen Sie die Dateien APPT und APDIR von Ihrem Computer in das Hauptverzeichnis Ihres HP 48 übertragen.
2. Durch Betätigung der Tasten [VAR] {APPT} können Sie das Programm starten.
3. Optional: Durch Betätigung der Taste {GOTO} können Sie den gewünschten Kalendertag anzeigen. Zur Eingabe eines neuen Termins ist es nicht erforderlich,

da zuerst der Kalender angezeigt wird.

4. Wählen Sie die gewünschte Menütaste, und beenden Sie die Eingabe.
5. Betätigen Sie die Taste {STOP} zum Verlassen des Programmes.

Anmerkung:

Falls Sie aus Versehen eine Taste betätigen und anschließend zu einer Eingabe aufgefordert werden, müssen Sie lediglich die Taste [ENTER] und anschließend entweder die Funktionstaste {ABORT} betätigen oder die Eingabe wiederholen.

IV.3 Menütasten

{FIND}

Fordert zur Eingabe einer Zeichenfolge, nach der gesucht werden soll, auf; anschließend wird in den Terminkommentaren nach dieser Zeichenfolge gesucht (Termine, die Rechnerfunktionen steuern, werden ignoriert). In der Anzeige muß der Monat Januar 1990 oder ein späterer Monat stehen. Bei der Suche werden Kommentare, die zu einem Termin vor dem angezeigten Datum gehören, nicht berücksichtigt.

{GOTO}

Fordert zur Eingabe eines bestimmten Kalendertages auf und zeigt diesen Kalendertag an.

{ADD}

Fordert zur Eingabe eines neuen Termins auf.

{DATE}

Fordert zur Eingabe eines bestimmten Tages auf.

{TIME}

Fordert zur Eingabe einer Uhrzeit auf (24-Stunden-Format).

{MSG}

Fordert zur Eingabe einer Meldung oder eines Kommentars auf (dabei wird automatisch auf die Alpha-Tastenbelegung umgeschaltet).

{RPT}

Fordert zur Eingabe eines Wertes für das Intervall zwischen ständig wiederkehrenden Terminen/Erinnerungen auf; die dabei gültige Zeiteinheit wird durch Drücken von {WEEK}, {DAY}, {HOUR}, {MIN}, {SEC} oder {NONE} angegeben.

{SET}

Speichert einen Termin und kehrt zum zuletzt angezeigten Kalendermenü zurück.

{ABRT}

Schaltet zum Kalendermenü zurück, ohne da ein Termin gespeichert wird.

{UPLD}

Bestimmte Terminblöcke werden auf einen Computer überspielt. Dabei wird zur Eingabe eines Anfangstermins und eines Endtermins für diesen Terminblock aufgefordert.

{APPTS}

Zeigt alle Termine für den im Kalender markierten Tag an; dabei werden auch neue Menüfelder angezeigt.

{EDIT}

Nach Eingabe dieses Befehls können Termine mit den ADD-Menütasten geändert werden.

{ACK}

Mit diesem Befehl kann das akustische Terminalsignal beim Erreichen eines Termins unterdrückt werden; Wiederholungstermine werden nicht ausgeführt.

{DEL}

Der ausgewählte Termin wird gelöscht.

{ADD}

Anzeige des ADD-Menüs zur Eingabe eines neuen Termins.

{VIEW}

Anzeige des Kommentars zu dem ausgewählten Termin.

{RTN}

Rückkehr zum Kalender-Menü.

{STOP}

Programm verlassen.

[+] - Blättert im Kalender 1 Monat vorwärts.

[-] - Blättert im Kalender 1 Monat zurück.

[linke Umschalttaste][+] - Blättert im Kalender 1 Jahr vorwärts.

[linke Umschalttaste][-] - Blättert im Kalender 1 Jahr zurück.

IV.4 Beispiel

Es ist z.B. der 7. Februar 1990, und Sie möchten einen Termin für den 27. August 1990 um 14:30 Uhr eingeben, der jeweils nach zwei Wochen wiederholt werden soll.

1. Betätigen Sie die Tasten {rechte Umschalttaste} {HOME} {VAR}

{APPT} zum Start des Programmes.

2. Bei Bedarf: Mit der Taste {GOTO} können Sie den 27. August 1990 in die Anzeige bringen.

3. Nach Betätigung der Taste {ADD} können Sie einen Termin eingeben.

4. Betätigen Sie die Tasten {DATE} 27.81990 {ENTER} zur Eingabe des Datums.

5. Betätigen Sie die Tasten {TIME} 14.30 {ENTER} zur Eingabe der Uhrzeit im 24-Stunden-Format.

6. Geben Sie nun {MSG} Abteilungsbesprechung {ENTER} ein, um den Kommentar zu speichern (beachten Sie, daß automatisch auf Alpha-Tastenbelegung umgeschaltet wird).

7. Betätigen Sie die Tasten {RPT} 2 {ENTER} {WEEK}, damit der Termin jeweils nach zwei Wochen wiederholt wird.

8. Speichern Sie den Termin durch Eingabe des Befehls {SET}.

9. Falls erforderlich, geben Sie {GOTO} 27. August, 1990 ein. Betätigen Sie die Taste {APPTS}, um zu überprüfen, ob der Termin gespeichert worden ist.

10. Betätigen Sie die Taste {RTN}, um zum Hauptmenü zurückzukehren. Mit {GOTO} können Sie wieder den 7. Februar 1990 anzeigen.

11. Geben Sie die Befehlsfolge {FIND} Besprechung {ENTER} ein, um den nächsten Termin in die Anzeige zu holen, bei dem im Kommentar das Wort "Besprechung" enthalten ist. Beachten Sie dabei, da die Suche an dem im Kalender markierten Tag beginnt und daß nur vorwärts gesucht wird.

12. Durch Betätigung der Taste {STOP} können Sie das Programm verlassen.

Anmerkung:

APDIR ist ein Verzeichnis, in dem die Programme und Variablen des APPT-Programmes gespeichert sind. Sie müssen lediglich das Programm APPT starten.

Falls der Programmablauf von APPT durch eine fehlerhafte Eingabe

abgebrochen wird, erscheint der Verzeichnispfad von APDIR in der oberen Zeile Ihrer Anzeige. Durch Eingabe der Tastenfolge [rechte Umschalttaste] [HOME] können Sie in das Hauptverzeichnis zurückkehren.

V STPWATCH

V.1 Überblick

Mit dem Programm STPWATCH können Sie Ihrem HP 48 sämtliche Funktionen einer Stoppuhr geben:

- o "Tickende" Echtzeit-Stoppuhr, menügesteuert (bis zu 100 Stunden).
- o Rücksetzen der Stoppuhr
- o Abrufen und Speichern von Zwischenzeiten (Anzahl nur durch Speicherkapazität begrenzt).
- o Anzeige und Ausdruck von Zwischenzeiten (absolut und relativ) im Format Stunden.Minuten.Sekunden.
- o Programmierbare Stoppuhr-Befehle

Das Programm erzeugt eine Liste in einer Anwendervariablen mit der Bezeichnung SWDAT:

SWDAT: { n array(mx2) d }

dabei ist:

n Natürliche Zahl (Anzahl gespeicherter Zwischenzeiten). Der Standardwert für n beträgt 0.

array(mx2)

Reelle, zweidimensionale Matrix (mx2) zur Speicherung von m Zwischenzeiten. Der Standardwert für m beträgt 25.

d Reelle Zahl, Verzögerung (in Sekunden) bei der Anzeige von mehreren Zwischenzeiten nacheinander. Der Standardwert beträgt 0,4.

Die folgende Befehlssequenz erzeugt zum Beispiel eine Variable SWDAT, in der 100 Zwischenzeiten gespeichert werden können und bei der die Verzögerung bei der Anzeige der gespeicherten Zwischenzeiten 0,5 Sekunden beträgt.

```
0 { 100 2 } 0 CON 0,5 3
→LIST 'SWDAT STO
```

V.2 Prozedur

1. Übertragen Sie die Datei STPWATCH.LIB mit Hilfe des

- Übertragungsprogrammes Kermit von Ihrem Computer zum Taschenrechner. Im VAR-Menü Ihres HP 48 erscheint das Menüfeld {STPW}.
- Binden Sie die Bibliothek in Ihr Hauptverzeichnis ein:
 - Betätigen Sie die VAR-Menütaste {STPW}, um die Bibliothek in den Stack zu laden.
 - Geben Sie die Portnummer an, der die Bibliothek zugeordnet werden soll (0, 1 oder 2) und betätigen dann die Taste [STO].
 - Schalten Sie den HP 48 aus und wieder ein. {SW-A} bindet sich automatisch ein und erscheint als Bibliothek im Menü LIBRARY.
 - Betätigen Sie die Tasten [linke Umschalttaste] [LIBRARY] {SW-A} zur Anzeige des Stoppuhr-Menüs.
 - Betätigen Sie die Taste {DOSW}, um die Stoppuhr zu aktivieren. Die Stoppuhr wird mit den Tasten {START}, {STOP} und {RESET} bedient; durch Betätigung der Taste [ENTER] können Sie Zwischenzeiten speichern, ohne daß die Stoppuhr angehalten werden muß.
 - Betätigen Sie die Taste [ATTN], um das Stoppuhr-Programm zu verlassen.

V.3 Menütasten

{DOSW}

Das Stoppuhr-Programm wird aktiviert, und das Stoppuhr-Menü wird angezeigt.

{START} - Startet die Stoppuhr.

{STOP} - Hält die Stoppuhr an.

{RESET}

Die Stoppuhr wird zurückgesetzt, und die Anzahl der gespeicherten Zwischenzeiten wird auf 0 gesetzt. (Betätigen Sie die Taste [ENTER], um Zwischenzeiten bei laufender Stoppuhr zu speichern.)

{VSP+}

Die n-te Zwischenzeit wird angezeigt, wobei n in Ebene 1 stehen muß. Durch wiederholte Betätigung wird der Wert n inkrementiert (n+1) und die jeweils nächste Zwischenzeit angezeigt.

{VSP-}

Die n-te Zwischenzeit wird angezeigt, wobei n in Ebene 1 stehen muß. Durch wiederholte Betätigung wird der Wert n dekrementiert (n-1) und die jeweils vorhergehende Zwischenzeit angezeigt.

{VALL}

Anzeige sämtlicher Zwischenzeiten (einmal) mit der angegebenen Verzögerung.

{PSP+}

Ausdruck der n-ten Zwischenzeit, wobei n in Ebene 1 stehen muß; anschließend wird n+1 in den Stack zurückgeschrieben.

{PALL}

Druckt alle gespeicherten Zwischenzeiten.

{RCLSP}

Anzeige der absoluten (Ebene 2) und der relativen (Ebene 1) Werte der n-ten Zwischenzeit, wobei n in Ebene 1 stehen muß. ({RCLSP} finden Sie auf der zweiten Menüseite.)

{RESET}

Stoppuhr-Zeit und Anzahl der gespeicherten Zwischenzeiten werden auf 0 zurückgesetzt. ({RESET} finden Sie auf der zweiten Menüseite.)

V.4 Anzeige der Stoppuhr

Anzahl der Zwischenzeiten, die gespeichert werden können	00025	00:00:12:34	Stoppuhrzeit
Anzahl gespeicherter Zwischenzeiten	00003	00:00:09:82	Zeit der letzten Tastenbetätigung

Anmerkung:

Wie bei jedem anderen Programm führt ein längerer Betrieb der Stoppuhr zur Entladung der Batterien. Dadurch könnten unter Umständen alle Daten, die im Taschenrechner gespeichert sind, verloren gehen.

Während des Stoppuhrbetriebes sind die Terminfunktionen ausgeschaltet.

VI CLK

VI.1 Überblick

Mit dem Anwendungsprogramm CLK können Sie die Genauigkeit der Systemuhr durch Synchronisation mit einem Zeitstandard verbessern, indem die nach Ablauf eines bestimmten Zeitintervalls entstandenen

Differenzen zur Ermittlung eines Korrekturwertes verwendet werden. Dies ist besonders bei Anwendungen wie der Navigation, bei denen eine extrem genaue Zeitbestimmung erforderlich ist, wichtig.

Bei der Auswertung der Differenzen werden auch evtl. Änderungen der Zeitzone in 1/2-Stunden-Intervallen berücksichtigt.

Bei Zeitdifferenzen, die ein Vielfaches von einer halben Stunde sind, wird eine Änderung der Zeitzone unterstellt, so daß diese Differenzen bei der Fehlerberechnung ignoriert werden. Alle sonstigen Differenzen werden als Abweichung interpretiert und korrigiert.

So entspricht z.B. eine Zeitdifferenz von +1:35 Stunden einer Zeitonenänderung von +1:30 Stunden (wird ignoriert) und einer Abweichung von +0:05 Stunden (die Uhr geht 5 Minuten nach). Eine Differenz von +1:25 Stunden wird als eine Zeitonenänderung von +1:30 Stunden und eine Abweichung von -0:05 Stunden eingestuft (die Uhr geht 5 Minuten vor). Analoge Beispiele sind -1:35 Stunden (-1:30 Stunden Zeitonenkorrektur und -0:05 Stunden Abweichung) und -1:25 Stunden (-1:30 Stunden Zeitonenänderung und +0:05 Abweichung).

VI.2 Prozedur

- Laden Sie das Programm CLK mit Hilfe von Kermit in das Hauptverzeichnis Ihres HP 48. Nach dem Start dieses Anwendungsprogramms sollten Sie die im HP 48 eingebauten Funktionen SET/ADJUST nicht verwenden.
- Geben Sie zum Aufruf des Programms die Befehle [VAR] {CLK} ein.
- Geben Sie eine Liste mit dem minimal zulässigen Zeitintervall zwischen zwei Korrekturen und der maximal zulässigen Zeitabweichung der Uhr ein, und betätigen Sie anschließend die Taste {C.RST}. Dadurch wird das Programm initialisiert.

4. Falls erforderlich, geben Sie eine Liste mit dem aktuellen Datum und der aktuellen Uhrzeit ein, und betätigen Sie die Funktion {C.SET}.
5. Geben Sie den gewünschten Einstellwert ein, und betätigen Sie {C.ADJ}.
6. Wiederholen Sie, sofern erforderlich, die Schritte 4 und 5.
7. Betätigen Sie {C.XCT}, sobald die Uhr synchronisiert ist.
8. Warten Sie einige Wochen und wiederholen Sie dann die Schritte 4, 5, 6 und 7.
9. Betätigen Sie die Taste {C.STS} zur Anzeige der Statusinformationen.
10. Betätigen Sie {C.SSP}, um den Korrekturterminzyklus zeitweise auszusetzen; zur Wiederaufnahme des ausgesetzten Korrekturtermins betätigen Sie {C.CNT}.
11. Wiederholen Sie Schritt 3 zum Löschen des Korrekturtermins. Sie initialisieren damit sämtliche CLK-Variablen neu.

Anmerkung:

Wenn Sie das Programm verlassen, werden die durch das Programm geänderten Flags nicht zurückgesetzt.

VI.3 Funktionstasten

Die Benutzerschnittstelle dieses Programmpaketes besteht aus insgesamt 5 Einzelprogrammen. Jedes Programm gibt eine Fehlermeldung aus, wenn die erwartete Eingabe nicht erfolgt.

Bei dieser Fehlermeldung wird in der Regel angegeben, welche Eingabe erwartet wurde. So muß die Eingabe einer Uhrzeit im Format HH.MMSSs und die Eingabe eines Datums im Format MM.DDYYYY erfolgen.

{C.SET}

Übernimmt aus Ebene 1 eine Liste mit der aktuellen Zeit und dem aktuellen Datum. Die eingebaute Uhr wird auf diese Uhrzeit und dieses Datum eingestellt.

{C.ADJ}

Übernimmt einen Einstellwert für die Uhrzeit aus Ebene 1. Ein positiver Wert stellt die Uhr vor, und ein negativer Wert stellt die Uhr zurück.

{C.XCT}

Übernimmt kein Argument aus dem Stack; dem Programm wird mitgeteilt, daß die Uhr nun exakt mit dem Zeitstandard synchronisiert ist. Falls dies die erste Synchronisierung ist, wird eine Zeitbasis aktiviert und sämtliche Fehlerkorrekturen werden ab diesem Zeitpunkt akkumuliert. Falls dies die zweite oder eine darauffolgende Synchronisierung ist, wird die Zeitbasis bis zu diesem Zeitpunkt erweitert, und die erforderlichen Berechnungen für das Zeit-Korrektur-Programm werden durchgeführt.

Das Programm führt ab diesem Zeitpunkt die erforderlichen Korrekturen der Zeitanzeige durch. Zu diesem Zweck wird festgelegt, in welchen Abständen Termine ausgelöst werden. Durch spätere und zusätzliche Synchronisierungen wird die Genauigkeit der Korrekturen gesteigert.

{C.RST}

Übernimmt aus Ebene 1 eine Liste mit dem Mindestintervall zwischen zwei Korrekturen und der maximal zulässigen Abweichung. (Beide Werte werden im Zeit-Format eingegeben).

Das Korrekturprogramm wird völlig zurückgesetzt, und die Zeitbasis wird deaktiviert. Der Termin wird gelöscht, und zukünftige Fehlerkorrekturen werden nicht mehr akkumuliert (die Summenspeicher werden gelöscht).

Alle Variablen werden neu initialisiert und das Verzeichnis wird reorganisiert.

Die maximale Zeitabweichung definiert den maximal zulässigen Fehler für die Uhr. Falls als Absolutwert für die maximale Abweichung ein Wert unter 0,01 Sekunden eingegeben wird, wird dieser auf 1 Sekunde gesetzt.

Das Mindestintervall definiert den Zeitraum, der zwischen zwei Korrekturvorgängen der Uhr mindestens abgewartet wird. Falls der Absolutwert dieses Zeitintervalls kleiner ist als eine Sekunde, wird dieses Intervall automatisch auf eine Minute gesetzt. Falls die Uhr, um den vorgegebenen maximalen Fehler nicht zu überschreiten, häufi-

ger korrigiert werden müßte als durch das Mindestintervall vorgegeben ist, wird der Korrekturwert erhöht, um zu erreichen, daß der Termin nicht öfter als vorgegeben ausgelöst wird. Dies kann dazu führen, da die Zeit nicht mehr mit der angegebenen Genauigkeit angezeigt wird. Dadurch wird allerdings erreicht, da Termine nicht so oft ausgelöst werden, daß der HP 48 keine Rechenzeit mehr für andere Aufgaben hätte (siehe auch "Anmerkungen" am Ende dieses Textes).

{C.STS}

Übernimmt keine Eingabe aus der Ebene 1, sondern zeigt 3 Seiten mit Statusinformationen an, die sich auf das Korrekturprogramm beziehen.

Alle Seiten haben die folgende Kopfzeile:

Korrekturstatus -N- (N = 1, 2 oder 3)

Alle Seiten haben die folgende Fußzeile:

prev:<exit:V next:>

(Verwenden Sie die entsprechende Cursor-Taste des HP 48.)

Beispiel für die Anzeige (Programmstatus), Seite 1:

Exact initiated():*

9.1455 10.301989

Deviation: 0.000025

Min rate: 1.0000

In den ersten beiden Zeilen wird die Uhrzeit und das Datum der ersten exakten Synchronisierung angezeigt.

Falls in der Anzeige "none" erscheint, ist bisher keine exakte Synchronisierung erfolgt. Falls ein (*) angezeigt wird, wurden bereits 2 oder mehr exakte Synchronisierungen durchgeführt. Dann ist das Programmpaket vollständig aktiviert. Falls kein (*) angezeigt wird, wurde bisher lediglich eine Synchronisierung durchgeführt (Abweichungen werden bereits akkumuliert, es wurden jedoch noch keine Korrekturwerte errechnet). In der dritten Zeile steht die durch den Anwender festgelegte maximale Abweichung (z.B. 1/4 Sekunde). Die vierte Zeile zeigt das festgelegte Mindestintervall zwischen zwei Korrekturen an (z.B. 1 Stunde).

Anzeigebeispiel (Einstelldaten),
Seite 2:
Timezones: 0.0000
Errors: 0.001890
Timebase: 1718.3909

In der ersten Zeile wird die Summe der Zeitonenänderungen angezeigt. Die zweite Zeile zeigt die bis zum jeweiligen Zeitpunkt akkumulierten Abweichungen (z.B. 18,9 Sekunden). In der dritten Zeile wird die Gesamtzeit, die bisher durch die Zeitbasis erfaßt worden ist, angezeigt (z.B. 1718 Stunden, 39 Minuten und 9 Sekunden).

Anzeigebeispiel (Korrekturdatum),
Seite 3:
Next correction:
11.1644 1.101990
Repeat: 22.4520
Adjust: 0.000025

In der ersten Zeile kann entweder "Next correction:" (nächste Korrektur - die nächste Korrekturzeit wird angezeigt) oder "Adjust suspended:" (Einstellung ausgesetzt - der Zeitraum, für den die Korrekturen ausgesetzt wurden, wird angezeigt). Die dritte Zeile enthält Informationen über das Wiederholintervall des Korrekturtermins (im Beispiel 22 Stunden, 45 Minuten und 20 Sekunden). Die vierte Zeile zeigt den Korrekturwert an (z.B. +1/4 Sekunden).

Falls eine dieser Anzeigen länger als 1 Minute stehen bleibt, ohne daß eine Taste betätigt wird, wird die Anzeige gelöscht und das Programm verlassen.

{C.SSP}

Die Korrekturtermine werden ausgesetzt (falls Korrekturen aktiviert sind). In der Statusanzeige wird das Zeitintervall der Aussetzung angegeben. Falls während der Aussetzung der Korrekturtermine eine der Tasten {C.SET}, {C.ADJ} oder {C.XCT} betätigt wird, kann dies unter Umständen zu Fehlern führen. ({C.SSP} befindet sich auf der Seite 2 des Menüs.)

{C.CNT}

Der Korrekturterminzyklus wird fortgesetzt, falls Korrekturen aktiviert sind. Ausgelassene Korrekturen

werden durchgeführt, und der nächste Korrekturtermin wird eingestellt ({C.CNT} finden Sie auf Seite 2 dieses Menüs.)

VI.4 Menü-Funktionstasten

Hinter dem Menüfeld {C.TLS} verbergen sich Unterprogramme und Variablen, die durch das Anwendungsprogramm intern benötigt werden. Für die Arbeit mit dem Programm wird die zugehörige Funktionstaste nicht benötigt.

Anmerkung:

Der Korrekturtermin wird unabhängig von dem Verzeichnispfad, der gerade aktiv ist, ausgelöst und sollte auch an der aktuellen Einstellung des Rechners nichts ändern. Falls jedoch der HP 48 beim Auslösen eines Termins ausgeschaltet ist, wird er automatisch eingeschaltet und bleibt eingeschaltet, bis der HP 48 sich bei Überschreiten einer Wartezeit selbst wieder ausschaltet. Falls der Termin in kürzeren Intervallen als der Wartezeit ausgelöst wird, bleibt der HP 48 immer eingeschaltet. Falls bei der Auslösung des Termins gerade ein anderer Vorgang abläuft, wird entweder die Korrektur nicht sofort ausgeführt (wenn gerade ein Programm läuft, wird der Termin erst bearbeitet, wenn das Programm verlassen wird) oder der Vorgang wird abgebrochen (ein Editiervorgang wird z.B. so abgebrochen, als ob die Taste [ATTN] betätigt worden wäre).

VII PCLPRINT

VII.1 Überblick

Mit der Befehlsbibliothek PCLPRINT können Sie Grafikobjekte auf einem PCL-Drucker wie z.B. dem HP ThinkJet, dem HP DeskJet oder einem HP LaserJet ausdrucken. Wenn die Bibliothek in das Hauptverzeichnis eingebunden ist, erhält sie die Bezeichnung HPPRT.

(Im Menü der Befehlsbibliothek erscheint das Menüfeld {HPPRT}).

Die beiden Befehle HPON und MAG fügen eine Parameterliste als fünftes Element an die im HP 48 reservierte Variable PRTPAR an. Diese Parameter enthalten Informa-

tionen, die Ihr HP 48 zum Ausdruck von Grafikobjekten auf einem PCL-Drucker benötigt. Im folgenden werden die Parameter in der gleichen Reihenfolge, in der sie auch in der Liste erscheinen, erläutert:

- o Die reelle Zahl 0, die darauf hinweist, da es um einen PCL-Drucker geht.
- o Eine reelle Zahl zur Angabe des Vergrößerungsfaktors. Der Vergrößerungsfaktor kann mit Hilfe des Befehles MAG geändert werden und steht standardmäßig auf 2. Dies bedeutet, da jedes Pixel des Grafikobjektes beim Ausdruck auf ein 2x2 Punkte großes Quadrat vergrößert wird.
- o Eine Zeichenfolge mit der Escape-Sequenz, die vor dem Ausdruck eines Grafikobjektes abgeschickt werden muß. Die Standardsequenz lautet "<esc>*r0A" und enthält die Anweisung an den Drucker, mit dem Ausdruck einer Rastergrafik am linken Seitenrand zu beginnen. Diese Sequenz kann ausschließlich durch Editieren von PRTRAR geändert werden.
- o Eine Zeichenfolge mit der Escape-Sequenz, die nach dem Ausdruck eines Grafikobjektes an den Drucker geschickt wird. Die Standard-Sequenz lautet "<esc>*rB<cr><lf>" und teilt dem Drucker mit, daß die Rastergrafik abgeschlossen ist und daß eine Zeilenschaltung und ein Druckkopfrücklauf durchgeführt werden sollen. Diese Sequenz kann ausschließlich durch Editieren von PRTRAR geändert werden.

VII.2 Prozedur

Im folgenden wird die Vorgehensweise zur Übertragung der Bibliothek in Ihren Taschenrechner und zum Ausdruck von Grafikobjekten Ihres HP 48 auf einen PCL-Drucker erläutert:

1. Überspielen Sie die Datei PCLPRINT.LIB von Ihrem Computer auf Ihren Taschenrechner. In dem VAR-Menü Ihres HP 48 erscheint das Menüfeld {PCLPR}.
2. Binden Sie die Bibliothek in Ihr

Hauptverzeichnis ein:

- A. Betätigen Sie im VAR-Menü die Funktionstaste {PCLPR}, um die Bibliothek in den Stack zu laden.
 - B. Geben Sie die Portnummer an, der die Bibliothek zugeordnet werden soll (0, 1 oder 2).
 - C. Geben Sie den Befehl STO ein.
 - D. Schalten Sie den HP 48 aus und wieder ein. {HPPRT} bindet sich selbst ein und erscheint als Bibliothek im LIBRARY-Menü.
3. Schließen Sie den HP 48 mit einem seriellen Schnittstellenkabel, einem 25-poligen Adapter und einem Stecker-Stecker-Adapter (gehört nicht zum Lieferumfang) an den PCL-Drucker an.
 4. Als nächstes müssen Sie die Baudrate des HP 48 (mit Hilfe des I/O-SETUP-Menüs) und die Baudrate des Druckers aneinander anpassen. Der Drucker muß auf XON/XOFF-Handshake-Betrieb eingestellt werden.
 5. Geben Sie den Befehl HPON ein (betätigen Sie dazu [linke Umschalttaste] [LIBRARY] {HPPRT} {HPON}).
 6. Geben Sie nun wahlweise den Befehl MAG oder den Befehl DPI ein (diese Befehle werden im folgenden erläutert).
 7. Geben Sie nun einen der Druckbefehle des HP 48 für den Ausdruck von Grafikobjekten ein: PRLCD, PR1 oder PRVAR. ([ON][PRINT] funktioniert nicht und sollte nicht ausprobiert werden.)
 8. Mit dem Befehl HPOFF können Sie den Taschenrechner wieder auf den Betrieb mit dem Infrarot-Drucker HP 82240 umschalten (betätigen Sie dazu [linke Umschalttaste][LIBRARY] {HPPRT} {HPOFF}).

VII.3 Befehlsübersicht

In der PCL-Bibliothek sind die folgenden Befehle enthalten:

HPON

Ermöglicht den Grafikausdruck und fügt die bereits erläuterte Unterliste

an PRTAR an. Außerdem wird das Flag -34 gesetzt und das Flag -33 zurückgesetzt, so daß der Ausdruck zur seriellen Schnittstelle geleitet wird. Weiterhin wird IOPAR so abgeändert, daß ein XON/XOFF Betrieb stattfindet, und die Parität wird auf "keine Parität" eingestellt.

HPOFF

Der Grafikausdruck wird gesperrt, und das Flag -34 wird zurückgesetzt (so daß Ausdrücke zum Drucker HP 82240B geleitet werden). Durch HPOFF wird die Unterliste in PRTPAR nicht geändert. Die eingestellten Vergrößerungsfaktoren sowie sonstige Änderungen bleiben also für die nächste Ausführung des Befehls HPON gespeichert.

MAG

Übernimmt eine reelle Zahl aus Ebene 1 und schreibt sie als Vergrößerungsfaktor in die Unterliste in PRTPAR. Ein Vergrößerungsfaktor N bewirkt, da ein Pixel des Grafikobjektes als ein auf NxN Punkte vergrößertes Quadrat ausgedruckt wird. (Bei Eingabe eines Vergrößerungsfaktors MAG = 0 findet kein Ausdruck statt.)

DPI

Übernimmt eine reelle Zahl aus Ebene 1 und sendet eine Escape-Sequenz an den Drucker, mit der die Anzahl der Punkte pro Zoll definiert wird. Dieser Befehl funktioniert ausschließlich bei den Druckern HP DeskJet und HP LaserJet.

VII.4 Zusätzliche Anmerkung für den HP LASERJET

Da der HP LaserJet erst dann einen Ausdruck macht, wenn eine ganze Seite vollgeschrieben ist, müssen Sie zum Ausdruck von Grafikobjekten, die kleiner sind als eine Seite, einen Seitenvorschub (das ASCII-Zeichen 12) senden.

Dies ist möglich, indem Sie eine Zeichenfolge mit dem Seitenvorschub-Befehl in die Ebene 1 schreiben und den Befehl PR1 ausführen. (Sie können auch die Unterliste in PRTPAR editieren und in die Abschlußsequenz den Befehl für einen Seitenvorschub aufnehmen. Dadurch erreichen Sie, daß jedes Grafikobjekt auf einer neuen Seite ausgedruckt wird.

VIII EPSPRINT

VIII.1 Überblick

Mit der EPSPRINT-Bibliothek können Sie ein Grafikobjekt auf einem Epson-Drucker mit serieller Schnittstelle wie z.B. dem FX-80 und dem FX-85 ausdrucken. Nach dem Einbinden in das Hauptverzeichnis erhält die Bibliothek die Bezeichnung EPPRT. In dem LIBRARY-Menü erscheint das Menüfeld {EPPRT}.

Zwei Befehle in der Bibliothek, EPON und MAG, fügen eine zusätzliche Parameterliste als fünftes Element an die im HP 48 reservierte Variable PRTPAR an. Diese Parameter enthalten Informationen, die Ihr HP 48 zum Ausdruck von Grafikobjekten auf einem EPSON-Drucker benötigt. Im folgenden werden die Parameter in der gleichen Reihenfolge, in der sie auch in der Liste erscheinen, erläutert:

- o Die reelle Zahl 1, die angibt, daß auf einem EPSON-Drucker gedruckt werden soll.
- o Eine reelle Zahl (1, 2 oder 4) zur Angabe des Vergrößerungsfaktors. Dieser Parameter kann mit Hilfe des Befehls MAG eingestellt werden und steht standardmäßig auf 2.

Das bedeutet, daß jedes Pixel des Grafikobjektes beim Ausdrucken auf ein 2x2 Punkte großes Quadrat vergrößert wird.

- o Eine Zeichenfolge mit der Escape-Sequenz, die vor dem Ausdruck eines Grafikobjektes abgeschickt werden muß. Die Standardsequenz lautet "<esc>A<08>". Durch diese Sequenz wird der Drucker auf einen Zeilenabstand von 8 Punkten eingestellt. Diese Zeichenfolge kann nur durch Editieren der Variablen PRTPAR geändert werden.
- o Eine Zeichenfolge mit der Escape-Sequenz, die nach dem Ausdruck eines Grafikobjektes an den Drucker geschickt wird. Die Standardsequenz lautet "<esc>2". Dadurch wird der Drucker auf den standardmäßigen Zeilenabstand von 6 Zeilen pro Zoll zurückgeschaltet. Dieser Parameter kann nur durch Edi-

tieren des Parameters PRTPAR geändert werden.

- o Eine Zeichenfolge mit der Escapesequenz, die vor jeder Zeile mit Grafikdaten an den Drucker geschickt wird. Diese Zeichenfolge, die die Auflösung der Grafik festlegt, steht standardmäßig auf "<esc>K". Das ist die Druckeranweisung für die einfache Zeichendichte. Folgende Einstellungen sind möglich:

<esc>K Grafiken mit einfacher Zeichendichte (60 DPI)

<esc>L Grafiken mit doppelter Zeichendichte (120 DPI)

<esc>Y Hochgeschwindigkeits-Grafiken mit doppelter Zeichendichte (120 DPI). Bei dieser Einstellung gibt es hinsichtlich der ausgedruckten Daten gewisse Einschränkungen.

<esc>Z Vierfache Zeichendichte (240 DPI). Bei dieser Einstellung gibt es hinsichtlich der ausgedruckten Daten gewisse Einschränkungen. Die Standardsequenz <esc>K kann nur durch Editieren der Variablen PRTPAR geändert werden.

VIII.2 Prozedur

In den folgenden Schritten wird erläutert, wie Sie die Bibliothek in Ihren Taschenrechner laden und anschließend ein Grafikobjekt des HP 48 auf einem EPSON-kompatiblen Drucker ausdrucken können:

1. Übertragen Sie die Datei EPSPRINT.LIB von Ihrem Computer auf Ihren Taschenrechner. In dem VAR-Menü Ihres HP 48 erscheint das Menüfeld {EPSPR}.
2. Binden Sie die Bibliothek in Ihr Hauptverzeichnis ein:
 - A. Betätigen Sie im VAR-Menü die Funktionstaste {EPSPR}, um die Bibliothek in den Stack zu laden.
 - B. Geben Sie die Schnittstellennummer an, der die Bibliothek zugeordnet werden soll

(0, 1 oder 2).

- C. Geben Sie den Befehl STO ein.
 - D. Schalten Sie den HP 48 aus und wieder ein. {EPPRT} bindet sich selbst in das Hauptverzeichnis ein und erscheint in dem LIBRARY-Menü als Bibliothek.
3. Schließen Sie den HP 48 mit einem seriellen Schnittstellenkabel, einem 25-poligen Adapter und einem Stecker/Stecker-Adapter (gehört nicht zum Lieferumfang) an den EPSON-Drucker an.
 4. Stellen Sie (mit Hilfe des I/O SETUP-Menüs) die Baudrate des HP 48 so ein, da sie mit der Baudrate des Druckers übereinstimmt. Der Drucker muß auf XON/XOFF-Handshake Betrieb eingestellt werden.
 5. Geben Sie den Befehl EPON ein (betätigen Sie dazu [linke Umschalttaste] [LIBRARY] {EPPRT} {EPON}).
 6. Falls erforderlich, geben Sie den Befehl MAG ein. (Erläuterungen dazu finden Sie weiter unten).
 7. Geben Sie einen der HP-48-Druckbefehle für den Ausdruck von Grafikobjekten ein: PRLCD, PR1 oder PRVAR. Der Befehl [ON][PRINT] funktioniert nicht und sollte nicht ausprobiert werden.
 8. Falls Sie wieder mit dem Infrarot-Drucker HP 82240 arbeiten möchten, geben Sie den Befehl EPOFF ein (betätigen Sie dazu [linke Umschalttaste] [LIBRARY] {EPPRT} {EPOFF}).

VIII.3 Befehlsübersicht

In der PCL-Bibliothek sind die folgenden Befehle enthalten:

EPON

Ermöglicht den Ausdruck spezieller Grafiken und fügt die bereits erwähnte Unterliste an die Variable PRTPAR an. Außerdem wird das Flag -34 gesetzt und das Flag -33 zurückgesetzt, so daß der Ausdruck direkt zur seriellen Schnittstelle geleitet wird. Ferner wird die Variable IOPAR so geändert, daß XON/XOFF-Betrieb stattfindet, und die

Parität wird auf "keine Parität" eingestellt.

EPOFF

Der Grafikausdruck wird gesperrt, und das Flag -34 wird zurückgesetzt (so daß die Ausdrücke wieder direkt zum Drucker HP 82240B geleitet werden). Durch den Befehl EPOFF wird die Unterliste in PRTPAR nicht geändert. Die eingestellte Vergrößerung und sonstige Änderungen der Parameter bleiben also für die nächste Ausführung des Befehls EPON gespeichert.

MAG

Übernimmt eine reelle Zahl aus der Ebene 1 und schreibt diese als Vergrößerungsfaktor in die Unterliste von PRTPAR. Zulässige Argumente sind 1, 2 oder 4 (das Vorzeichen wird ignoriert). Ein Vergrößerungsfaktor N bewirkt, daß ein Pixel des Grafikobjektes beim Ausdruck auf dem Drucker als NxN Punkte großes Quadrat ausgedruckt wird.

IX GROB2TIF

IX.1 Überblick

Das Computer-Programm GROB2TIF konvertiert Grafikobjekte (GROB) des HP 48 in das TIFF-Dateiformat (Tag Image File Format). Grafiken im TIFF-Dateiformat können durch sehr viele Textverarbeitungs-, Grafik- und Desktop-Publishing Programme verarbeitet werden.

IX.2 Prozedur

Im folgendem wird die Vorgehensweise zur Erzeugung von TIFF-Dateien erläutert:

1. Erzeugen Sie mit Ihrem HP 48 ein Grafikobjekt (durch Eingabe der Befehlsfolge PICT RCL.)
2. Speichern Sie das Grafikobjekt in einer Variablen Ihres HP 48.
3. Übertragen Sie die Variable auf Ihren Computer. Da das Programm GROB2TIF als Ausgangsdatei ASCII-Dateien benötigt, müssen Sie sicherstellen, da Ihr Taschenrechner im I/O SETUP-Menü auf die Übertragung von ASCII-Zeichen eingestellt ist.
4. Starten Sie das Programm GROB2TIF.

GROB2TIF liest die GROB-Datei vom Standardeingabegerät und schreibt die Ausgabedatei zum Standardausgabegerät. Sie müssen der Ausgabedatei deshalb beim Aufruf des Programms GROB2TIF einen neuen Namen geben:

```
GROB2TIF < GROB_Datei >
TIFF_Datei
```

IX.3 Beispiel

Falls z.B. eine GROB-Variable in die Computer-Datei BILD.GRO überspielt worden ist, müssen Sie folgenden Befehl eingeben:

```
GROB2TIF <BILD.GRO >
BILD.TIF
```

Nach Betätigung der Taste [ENTER] wird die GROB-Datei (BILD.GRO) in eine TIFF-Datei (BILD.TIF) konvertiert.

X INPRT

X.1 Überblick

Mit dem Anwendungsprogramm INPRT können Sie Druckerdaten, die andere Hewlett-Packard Taschenrechner über die Infrarot-Schnittstelle ausgeben, in den Stack des HP 48 einlesen. INPRT geht davon aus, daß die eingehenden Bytes mit dem Zeichensatz Roman 8 auf dem Drucker HP 82240 ausgedruckt werden sollten und konvertiert diese Bytes in den Zeichensatz des HP 48 (ISO 8859 Latin 1).

X.2 Prozedur

Im folgenden wird die Vorgehensweise zur Übertragung von Druckerangaben über die IR-Schnittstelle in den HP 48 erläutert:

- Übertragen Sie die Datei INPRT von Ihrem Computer in den HP 48. Falls Sie sie von einem anderen dem HP 48 übertragen, muß dies im Binär-Modus erfolgen. Im VAR-Menü Ihres HP 48 erscheint das Menüfeld {INPRT}. Falls Ihr Taschenrechner doppelte Zeilenabstände erzeugen kann, muß zuerst der einfache Zeilenabstand eingestellt werden.
- Stellen Sie die Taschenrechner einander gegenüber (Informationen dazu finden Sie weiter un-

ten).

- Betätigen Sie auf dem HP 48 die Funktionstaste {INPRT}. INPRT schaltet sich selber nach 10 Sekunden wieder aus, wenn kein IR-Ausgabesignal empfangen wird.
- Geben Sie auf dem anderen Taschenrechner einen Druckbefehl ein.
- Wiederholen Sie die Schritte 2 bis 4, wenn weitere Druckdaten übertragen werden sollen.

X.3 Ergebnis

Nach Abschluß der Übertragung zeigt der HP 48 die empfangenen Daten als Zeichenfolge in Ebene 2 sowie ein Flag in Ebene 1 an. Falls das Flag auf 0 gesetzt ist, deutet dies darauf, daß bei der Übertragung Fehler aufgetreten sind, die nicht korrigiert werden konnten.

Falls das Flag auf 1 steht, kann es trotzdem sein, daß einzelne Bytes fehlen. Der Wert 1 bedeutet, daß jedes Byte, da erfaßt worden ist, auch korrekt empfangen wurde. Trotzdem müssen Sie empfangene Daten überprüfen, da in der Zeichenfolge bei Namen, Ausdrücken usw., die länger sind als 24 Zeichen, unter Umständen ein Zeilenvorschub enthalten sein kann. Falls das Flag auf 0 steht, wird jedes Byte, das nicht korrekt empfangen wurde, in der Zeichenfolge in Ebene 2 als grauer Block (Zeichencode 127) angezeigt.

X.4 Anweisungen für das Ausrichten der Taschenrechner

Die Taschenrechner sollten zur Übertragung höchstens 1,3 cm voneinander entfernt stehen. Außerdem sind bei einzelnen Taschenrechnern einige weitere Aspekte zu beachten:

Beim **HP41**: Das IR-Druckermodule sollte sich in einem der beiden oberen Steckplätze befinden. Der glänzende "IR-Kegel" in der Mitte des Moduls sollte genau zwischen die Buchstaben E und T von HEWLETT auf dem Hewlett-Packard-Logo des HP 48 gerichtet sein.

Beim **HP27S**, **HP17B** und **HP42S**: Der Taschenrechner muß 0,75 cm höher gestellt werden. Die Mitte der

Leuchtdiode (die sich am oberen Ende des Taschenrechners zwischen den Buchstaben E und W von HEWLETT befindet) sollte genau zwischen die Buchstaben E und T von HEWLETT auf dem Hewlett-Packard-Logo des HP 48 gerichtet sein.

Beim **HP28**, **HP18C** und **HP19B**: Es reicht aus, wenn Sie die Leuchtdiode zwischen die Buchstaben E und T von HEWLETT auf dem Hewlett-Packard-Logo des HP 48 richten.

Inhaltsverzeichnis

Kap.	Titel
I	README
I.1	Anmerkung
I.2	Lieferumfang Ihres Schnittstellenpaketes
I.3	Einsatz dieser Anwendungsprogramme
I.4	Anschlussbelegung des Schnittstellenkabels
I.5	Mailbox für Anwender von HP-Taschenrechnern
II	KERMIT
II.1	Überblick
II.2	Prozedur
II.3	KERMIT-Hilfen
II.4	Beispiel
II.5	Wichtige Hinweise
II.6	KERMIT auf ATARI ST/TT
III	USAG
III.1	Überblick
III.2	Prozedur
III.3	Beispiel
IV	APPT
IV.1	Überblick
IV.2	Prozedur
IV.3	Menütasten
IV.4	Beispiel
V	STPWATCH
V.1	Überblick
V.2	Prozedur
V.3	Menütasten
V.4	Anzeige der Stoppuhr
VI	CLK
VI.1	Überblick
VI.2	Prozedur
VI.3	Funktionstasten
VI.4	Menü-Funktionstasten
VII	PCLPRINT
VII.1	Überblick
VII.2	Prozedur
VII.3	Befehlsübersicht
VII.4	Zusätzliche Anmerkung für den HP LASERJET
VIII	EPSPRINT
VIII.1	Überblick
VIII.2	Prozedur
VIII.3	Befehlsübersicht
IX	GROB2TIF
IX.1	Überblick
IX.2	Prozedur
IX.3	Beispiel
X	INPRT
X.1	Überblick
X.2	Prozedur
X.3	Ergebnis
X.4	Anweisungen für das Ausrichten der Taschenrechner

Martin Meyer (1000)

Statik auf dem HP48SX

Durchlaufträger für beliebig viele - Felder - Belastungen - Stützmente von Claus Dachselt

Das Programm "DULA" ermittelt für einen Durchlaufträger mit beliebiger Belastung und unterschiedlichen, feldweise jedoch konstanten Steifigkeiten die Stützmente.

Als Vorlage diente das seinerzeit für den HP97 konzipierte und in die UPLE (Users Program Library Europe) aufgenommene Programm unter der Nummer 50885 mit dem Namen "Continuous Beam of 2 to 8 Spans - Support Moments". Es stammte von Hannes Rop aus Graz.

Die hier vorliegende Version wurde noch um den Lastfall "Stützensenkung" erweitert.

Nach der Abfrage der Anzahl n der Felder wird das Menü zur Eingabe der Längen und Steifigkeitsverhältnisse aufgerufen.

```

Feld 1
4:
0:
2:
1:
LNGE  IC/I  CONT
  
```

Dabei entspricht eine starre Einspannung an einem Trägerende einem Blindfeld mit der Länge $l=0$; eine elastische Einspannung kann durch ein gedachtes Feld mit $l=3 \cdot E \cdot I_c / C$ berücksichtigt werden.

I_c ist das für den gesamten Träger geltende Vergleichsträgheitsmoment (siehe auch das Rechenbeispiel), C die Federkonstante der elastischen Einspannung.

Bei bekanntem Einspanngrad ε [%] ermittelt sich die Ersatzlänge des fiktiven Endfeldes zu

$$l = 3/4 \cdot (100 - \varepsilon) \cdot l_i / \varepsilon \quad (l_i = \text{Länge des wirklichen Endfeldes}).$$

Die jeweilige Eingabe erfolgt durch Eintippen der Zahlenwerte und dem anschließenden Drücken des entsprechenden Softkeys "LNGE" bzw. "IC/I".

Allerdings braucht das Verhältnis I_c/I_i nur eingegeben zu werden, wenn dieser Wert $\neq 1$ ist.

Abgeschlossen wird dieser Programmteil durch Drücken von "CONT" im Menüfeld, worauf das Menü zur Eingabe der Belastung

erscheint. Die entsprechenden Tastendrücke sind den Beispielen der 6 verschiedenen Belastungsarten zu entnehmen, wobei \blacktriangledown die Cursor-Down Taste darstellen soll.

Sind alle Lasten eines Feldes eingegeben, so erfolgt der Abschluß durch das Drücken der Taste "EXIT"; das Programm geht nun zur Längeneingabe des nächsten Feldes weiter usw..

Nachdem die Belastung für das letzte Trägerfeld eingegeben worden ist, erfolgt nach kurzer Rechenzeit die Ausgabe der Stützmente. Bei mehr als 7 anzuzeigenden Werten, d.h. $n > 8$, erfolgt die Ausgabe blockweise:

M1-M7, dann "CONT" im Tastenfeld drücken, um M8 und gegebenenfalls weitere Momente anzuzeigen.

Nach Abschluß der Ergebnisausgabe und Drücken von "CONT" beginnt das Programm wieder mit der Längeneingabe für Feld 1; ein neuer Lastfall kann untersucht werden.

Nr.	Anweisung	Eingabe	Taste	Anzeige
1	Aufruf des Variablenmenüs mit Taste "VAR"		VAR	
2	Starten des Programms		DULA	Felderzahl:
3	Eingabe Anzahl d. Felder	4	ENTER	Feld 1
4	Eingabe Länge Feld 1	0	LNGE CONT	
5	Blindfeld; ohne Lasten !		EXIT	Feld 2
6	Eingabe Länge Feld 2	6.5	LNGE	
7	" I_c/I Feld 2	.8	IC/I CONT	

8	Belastung Feld 2 : Gleichlast		GLEI	Gleichlast q:
8a		10	ENTER	
8b	" Moment		M	Einzelmoment a: M:
		3.5 -17	▼ ENTER	
8c	Abschluß Belastg. Feld 2		EXIT	Feld 3
9	Eingabe Länge Feld 3 (Eing. Ic/I entfällt, da = 1!)	5.3	LNGE CONT	
10	Belastung Feld 3 : Trapezlast		TRAP	a: s: ql: qr:
9a		1.5 3 25 15	▼ ▼ ▼ ENTER	
9b	" , linke Einzellast		P	Einzellast a: P:
		1.5 20	▼ ENTER	
9c	" , rechte Einzellast		P	Einzellast a: P:
		4.5 30	▼ ENTER	
9d	Abschluß Belastg. Feld 3		EXIT.	Feld 4
10	Eingabe Länge Feld 4 " Ic/I "	4.2 1.2	LNGE IC/I CONT	
11	Eingabe Belastung Feld 4: Trapezlast		TRAP	a: s: ql: qr:
		0 4.2 0 18	▼ ▼ ▼ ENTER	
12	Abschluß Eingabe Belastg. Feld 4		EXIT	
13	Anzeige Stützmoment M1 - M3			M1: -31.397 M2: -48.967 M3: -40.882
14	Fortsetzung des Programms über Taste "CONT" im Tastenfeld zur Eingabe eines neuen Rechenlaufs		 CONT	Feld 1

Verwendete Formeln zur Ermittlung der Belastungsglieder

$L = R = q \cdot \ell^2 / 4$ ----- Prgm.teil "GLE"

$L = L + X1 \cdot q_l / \ell^2$
 $R = R + X2 \cdot q_l / \ell^2$ } mit $b = l - a - s$ ----- Prgm.teil "TRAP" ①

$L = L + X2 \cdot q_n / \ell^2$
 $R = R + X1 \cdot q_n / \ell^2$ } mit $a = b ; b = a$ ----- Prgm.teil "TRAP" ②

$X1 = s/6 (v \cdot a^2 \cdot (3 \cdot b + 2 \cdot s) + w \cdot b^2 \cdot (3 \cdot a + s) + x \cdot a \cdot b \cdot s + s^2/2 \cdot (8 \cdot a + 4 \cdot b + y + z \cdot s))$

mit $v = 1; w = 2; x = 10; y = b; z = 1.6$

X2 wie vor ;

mit $v = 2; w = 1; x = 8; y = -a; z = 1.4$

für "TRAP" ①

X1, X2 wie vor mit $a = b; b = a$ für "TRAP" ②

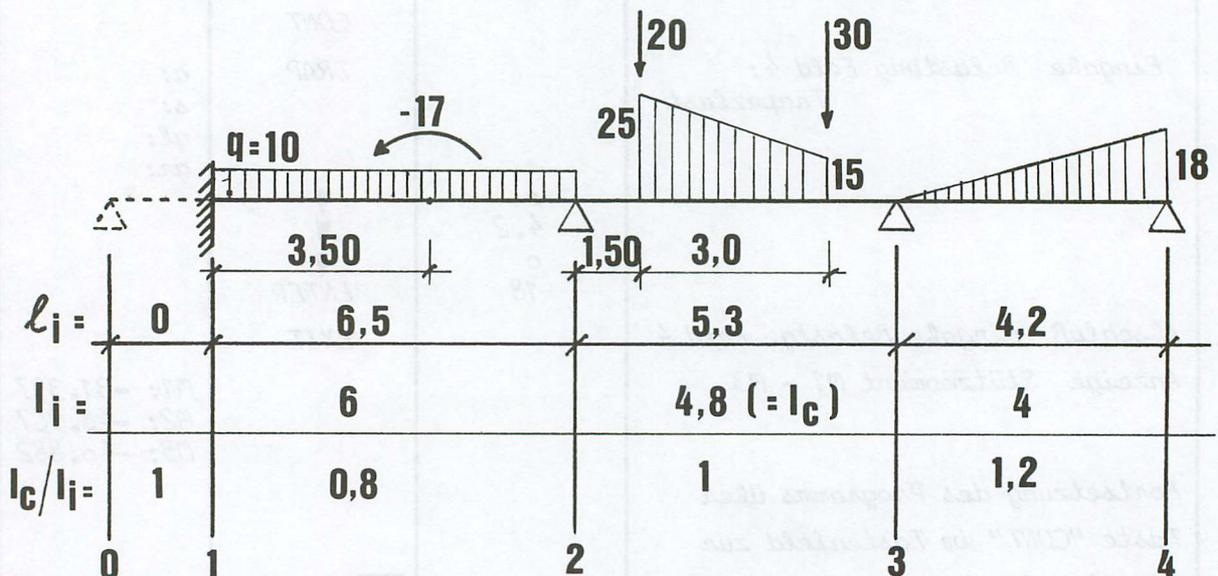
$L = L + P \cdot a \cdot b \cdot (l + b) / \ell^2$ ----- Prgm.teil "EIN"

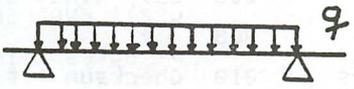
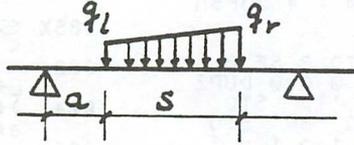
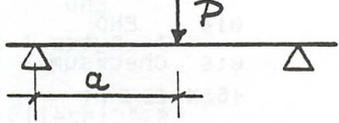
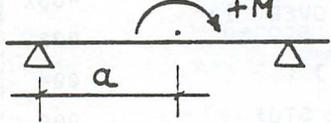
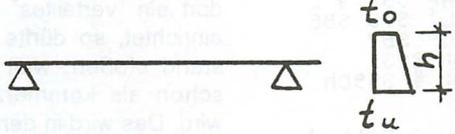
$R = R + P \cdot a \cdot b \cdot (l + a) / \ell^2$ ----- Prgm.teil "EIN"

$L = L + M \cdot (3 \cdot b^2 - \ell^2) / \ell^2$ ----- Prgm.teil "MO"

$R = R + M \cdot (\ell^2 - 3 \cdot a^2) / \ell^2$ ----- Prgm.teil "MO"

System zu Rechenbeispiel Seite 4



<p>Gleichlast: Taste GLEI</p>  <p>{ HOME DURCH } PRG Gleichlast</p> <p>:q: \leftarrow</p> <p>GLEI TRAP P M TEMP EXIT</p> <p>Eingabe: q ENTER</p>	<p>Trapezlast: Taste TRAP</p>  <p>{ HOME DURCH } PRG</p> <p>: a: \leftarrow</p> <p>: s: \leftarrow</p> <p>: q_l: \leftarrow</p> <p>: q_r: \leftarrow</p> <p>GLEI TRAP P M TEMP EXIT</p> <p>Eingabe: a ↓</p> <p>s ↓</p> <p>q_l ↓</p> <p>q_r ENTER</p>
<p>Einzellast: Taste P</p>  <p>{ HOME DURCH } PRG</p> <p>Einzellast</p> <p>:a: \leftarrow</p> <p>:P: \downarrow</p> <p>GLEI TRAP P M TEMP EXIT</p> <p>Eingabe: a ↓</p> <p>P ENTER</p>	<p>Einzelmoment: Taste M</p>  <p>{ HOME DURCH } PRG</p> <p>Einzelmoment</p> <p>:a: \leftarrow</p> <p>:M: \curvearrowright</p> <p>GLEI TRAP P M TEMP EXIT</p> <p>Eingabe: a ↓</p> <p>M ENTER</p>
<p>Temperatur: Taste TEMP</p>  <p>$L = R = E \cdot I \cdot \alpha_t \cdot 3 \cdot (t_u - t_o) / h$</p> <p>{ HOME DURCH } PRG</p> <p>Temperatur</p> <p>:L = R:</p> <p>GLEI TRAP P M TEMP EXIT</p> <p>Eingabe: L (= R) ENTER</p>	<p>Stützensenkung: Taste NXT STUE</p>  <p>{ HOME DURCH } PRG</p> <p>Stützensenk.</p> <p>:E·I: \leftarrow</p> <p>:delta s_l: \downarrow</p> <p>:delta s_r: \downarrow</p> <p>STUE</p> <p>Eingabe: E·I ↓</p> <p>delta s_l ↓</p> <p>delta s_r ENTER</p>

```

48SX TEM
002 < "Temperatur" {
OBJ> SB9
004 > Bytes : 53.5
Checksum : # 33F5h

48SX DUL1
002 < 'i' STO 0 0 0 DUP2
'j' STO 'lj' STO
004 'bj' STO 'BR' STO
'BI' STO 1 n 1 - 2
006 +LIST 0 CON 'MI'
STO 1 n 2 - 2 * DUP
008 IF 0 ==
THEN CLEAR
ELSE 2 +LIST 0
CON 'BI' STO
010 END SB1
012 > Bytes : 221
014 Checksum : # F480h

48SX SB0
002 < 'I' INCR DROP
> Bytes : 24.5
Checksum : # 00C7h

48SX SB1
002 < CLEAR BR 'B' STO+
2 'bj' STO+ lj DUP
'BI' STO* NEG 'bj'
004 STO* 1 DUP 'lj' STO
'j' STO+ STD
006 "Feld " j +STR + (
< "Lnge" j +STR + (
008 < DUP OVER 'lj'
STO* 'L' STO SQ
010 'LL' STO
> ) ( ) ( ) ( )
012 "ic/i"
< 'lj' STO*
> ) ( ) ( "CONT"
014 < CONT
> ) ) TMENU
016 PROMPT 0 DUP 'BL'
STO 'BR' STO AUF
018 > Bytes : 328.5
020 Checksum : # ED1EH

48SX SB2
002 < 'I' 'j' 1 STO- j
'I' STO B 'MI' MST
'I' STO
004 IF 1 j ==
THEN SB3
ELSE 'BI' MRC 'B'
006 STO* 'BI' MRC 'B'
STO+ RP
008 END
010 > Bytes : 173.5
Checksum : # C9D9h

48SX SB3
002 < CLLCD 1 DUP 'k'
STO n 1 -
004 FOR U STD " M" U
+STR + " : " + 3
006 FIX MI U GET + k
DISP 'k' INCR DROP
U 7 MOD
008 IF 0 ==
THEN 7 FREEZE
010 HALT 1 'k' STO
CLLCD
012 END
NEXT
014 IF n 7 MOD 0 *
THEN 7 FREEZE
016 HALT
END 2 MENU n DUL1
018 > Bytes : 223.5
Checksum : # 54D1h

48SX SB4
002 < DUP 3 ROLL * 3
ROLLD *
004 > Bytes : 27.5
Checksum : # DAB0h

48SX SB5
002 < 1 SF 1 2 10 L 1.6
SB6 2 1 8 a NEG 1.4
SB6
004 > Bytes : 83.5
Checksum : # 8D0Ah

48SX SB6
002 < > y w x y z
< > s/6*(v*a^2*(3*
L+2*s)+w*L^2*(3*a+s
004 )+x*a*L*s+s*s/2*(8*
a+4*L+y+z*s))' EVAL
006 > Bytes : 230
008 Checksum : # 4932h

48SX SB7
002 < LL ROT LL / 'BL'
STO+ / SB8
004 > Bytes : 48
004 Checksum : # CBC3h

48SX SB8
002 < 'BR' STO+
IF 1 FC?
THEN LL / 'L' STO
004 END
006 > Bytes : 60.5
006 Checksum : # BF18h

48SX SB9
002 < DUP 'BL' STO+ SB8
> Bytes : 32
004 Checksum : # B6FFh

48SX AUF
002 < ( ( "GLEI" GLE )
( "TRAP" TRAP ) (
004 "p" EIN ) ( "M" MO
) ( "TEMP" TEM ) (
"EXIT" EX ) (
006 "STUE" STUE ) )
TMENU
008 > Bytes : 155
008 Checksum : # 4D62h

48SX GLE
002 < "Gleichlast"
":q" INPUT OBJ+ 4
004 / LL * SB9
> Bytes : 57.5
004 Checksum : # 5D70h

48SX TRAP
002 < " " (
": a: (
": s: (
": q: (
": r: (
006 ( 1 0 ) ) INPUT
OBJ+ 'QR' STO 'QL'
008 STO 's' STO 'a' STO
'LI' s a + STO- SB5
010 QL SB4 SWAP SB7 a L
'a' STO 'LI' STO SB5
012 QR SB4 1 CF SB7
> Bytes : 223
014 Checksum : # 359Ch

48SX MST
002 < DUP EVAL 3 ROLL I
SWAP PUT SWAP STO
004 > Bytes : 34.5
004 Checksum : # B5Ah

48SX MRC
002 < EVAL I GET 'I'
DECR DROP
004 > Bytes : 34
004 Checksum : # 6679h

48SX RP
002 < SB2
> Bytes : 16.5
004 Checksum : # 9739h

48SX DULA
002 < "Felderzahl:" ""
INPUT OBJ+ DUL1
004 > Bytes : 43.5
004 Checksum : # 51F4h

48SX MO
002 < "Einzelmoment" (
":a: ( 1 0 ) )
":M: ( 1 0 ) )
004 INPUT OBJ+ > y z
< 'z*(3*(L-y)^2-
006 LL)' EVAL 'z*(LL-3*
y^2)' EVAL SB7
008 > Bytes : 163
010 Checksum : # 1710h

48SX STUE
002 < "Stützensenkgr." (
":E:I: (
":s:l: (
":s:r: (
004 ( 1 0 ) ) INPUT
OBJ+ - * 6 * LL /
006 DUP NEG 'BL' STO+
SB8
008 > Bytes : 112.5
010 Checksum : # 8F98h

48SX EX
002 < IF 2 FS?C
THEN SB1
ELSE 'bj' lj 2 *
004 STO- 'BI' bj STO/ lj
bj / DUP 'bj' STO
BL * 'BI' STO+
006 IF n j ==
THEN SB2
ELSE SB0 B 'BI'
008 MST SB0 bj 'BI' MST
SB1
010 END
012 END
014 > Bytes : 223
016 Checksum : # 17EEh

48SX EIN
002 < "Einzellast" (
":a: ( 1 0 ) )
":P: ( 1 0 ) )
004 INPUT OBJ+ > y z
< 'z*y*(L-y)*(2*L
006 -y)' EVAL 'z*y*(L-y
)*(L+y)' EVAL SB7
008 > Bytes : 179
010 Checksum : # EF57h

```

Claus Dachselt
Auf dem Hollen 10
W5810 Witten 6

Falls jemand glaubt, daß man für den Club und die Mitglieder noch mehr sparen kann, wenn man die CCD-Aktivitäten ganz ins FIDO-Net legt und dort ein "verteilttes" schwarzes Brett einrichtet, so dürfte das auf Widerstand stoßen, weil das von vielen schon als kommerziell angesehen wird. Das wird in dem privaten FIDO-Net nicht ganz zu Unrecht abgelehnt, denn es ist nicht leicht, da eine Grenz zu ziehen. Und etliche CCD-Mitglieder wollen wirklich kommerziell mit der BOX arbeiten. Wenn sich das jedoch ausschließlich in der Box von André oder Axel abspielt, dann ist das kein Problem.

Also Liebe Freunde, meldet euch bei Wolfgang Fritz oder mir, was Ihr von diesen Vorschlägen haltet.

Werner Dworak



Maschine&Sprache

Maschinensprache für den HP48

Teil II

von Georg Hoppen

Der Alleskönner oder ein Programm für alles

RUM→PRG

Im ersten praktischen Teil der Serie Maschine&Sprache soll ein Programm gemeinsam erarbeitet werden, das alle drei innerhalb des Rechners vorherrschenden Sprachen versteht und aus den in einer Liste gesammelten Einzelobjekten ein komplettes Programm erstellt.

Im Anschluss daran sollen die Grundwerkzeuge erstellt werden, ohne die auf dem Gebiet der "neuen Programmierung" eigentlich nichts läuft.

Weiterhin werde ich Programmbeispiele aus dem Toolkit von Donelly vorstellen, so daß auch "Nichtbesitzer" desselben in den Genuß sehr mächtiger Funktionen kommen können.

Hierbei wird Wert auf folgende Feststellung gelegt:

Die Erarbeitung wird Schritt für Schritt erfolgen, so daß auch absolute Laien in der neuen Programmierung mithalten können und nach diesem Artikel in der Lage sind bereits eigene Entwicklungen zu gestalten (schließlich möchte ich auch irgendwann einmal von Euren Gedanken profitieren...).

Weiterhin sind die folgenden Programme nicht in erster Linie auf Eleganz und Kürze getrimmt sondern orientieren sich an dem im Rechner ablaufenden Vorgängen, um diese besser deutlich machen zu können.

Ich bin daher im wesentlichen auf Eure Rückmeldungen angewiesen, ob auch alles klar und verständlich herüber gekommen ist und ob ich z.B. mehr voraussetzen kann oder soll, damit man schneller am Ziel

seiner Wünsche ankommt ??

Als Hardwarevoraussetzung empfiehlt sich eine zusätzliche Speichererweiterungskarte, da ansonsten durch die scheinbar unvermeidlichen "Memory Lost's" unnützer Aufwand betrieben werden müßte.

Bei sorgfältigem Arbeiten und doppelter Prüfung geht es natürlich auch ohne eine solche als Sicherungshaken...

Alle Programme wurden getestet und müßten eigentlich korrekt laufen, ich kann aber aus verständlichen Gründen keinerlei Haftung dafür übernehmen, daß sie unter allen Softwareumgebungen einwandfrei funktionieren. Bei mir laufen sie schon seit mehreren Monaten fast täglich einwandfrei.

Jetzt aber los, Ende des langen Vorspanns.

Vor dem Start muß noch kontrolliert werden, daß der Rechner im HEX-Modus steht und die Wortbreite auf 64 Bit gesetzt ist (notfalls mit 64 STWS setzen).

Im letzten PRISMA habe ich zwei kurze Routinen zum Schnuppern angeboten:

→PRG und SYSRCL.

Beide machen genau das, was ihr Name besagt:

→PRG formt die Elemente aus einer Liste zu einem Programm um, während SYSRCL Objekte, die an einer bestimmten #-Adresse stehen in den Stack holt.

Beide Routinen arbeiteten mit dem Befehl SYSEVAL zusammen, um eine bestimmte Prozedur im Rechner in Gang zu setzen.

Warum muß einer Adresse immer ein SYSEVAL folgen ?

Geht es denn nicht auch ohne - so

könnte man doch immer die 2,5 Byte sparen ?

Um diesen Vorgang besser verstehen zu können wollen wir uns noch einmal das Kurzprogramm SYSRCL näher ansehen und die Funktion auf sich selbst anwenden.

Der Ursprungscode lautet:

```
« # 4003h SYSEVAL
  # 56B6h SYSEVAL
  # 3244h SYSEVAL »
```

Diesen Vorgang umgeben wir mit einer kurzen Schleife, also :

```
« { } 'liste' STO DUP
  SIZE 1 SWAP FOR i DUP
  i GET # 4003h SYSEVAL
  # 56B6h SYSEVAL
  # 3244h SYSEVAL → obj
  « liste OBJ→ 1 + obj
  SWAP →LIST 'liste'
  STO »
  NEXT liste
»
```

Warum diese komplizierte Schleifentechnik - es geht doch scheinbar mit STO+ genauso gut ??

Leider, leider -- habt Ihr schon einmal eine leere Liste zu einer existierenden Liste gespeichert ?

Der Rechner klaut sie dabei anscheinend. D.h., daß unter Umständen mühsam erzeugte Objekte plötzlich nicht mehr in der Sammel-liste auftauchen. Auch werden 'Namen' zu Namen und ähnliche komische Dinge passieren, sofern eine Listenmanipulation stattfindet.

Andererseits ist eine Liste ein herrliches Objekt, um Dinge in den Stack zu packen, die ansonsten eine unmittelbare Ausführung bedeuten würden.

Was passiert z.B., wenn eine Liste nur den Befehl DROP enthält und

diese Liste mit →OBJ umgewandelt wird ?

Diese Funktion der Liste als natürliches Sammelbecken für Objekte machen wir uns schamlos für eigene Zwecke zunutze, um auf diese Weise alle Objekte im Stack zu sammeln.

Wenn wir nun eine Liste mit den bereits bekannten #-Adressen in den Stack bringen {# 4003h # 56B6h # 3244h} passiert folgendes: Im Stack stehen zwei Listen, die eine mit den Adressen, die andere mit den "Externals". Ihr wißt schon, diese komischen Dinger, fürchterlich anzusehen, böseartig grinsend und das nächste Memory Lost verursachend...

Doch so schlimm ist das diesmal nicht. Das, was der Rechner anzeigt ist lediglich der ausführbare RPL-Code an einer bestimmten RAM- bzw. ROM-Adresse.

Halt, war da nicht schon wieder etwas unbekanntes? RPL-Objekt? Ein fliegende Untertasse oder etwas ähnlich fremdartiges?

Weit gefehlt:

RPL (für ganz Wißbegierige: Es ist die Kurzfassung der Rechnerphilosophie, die (gelungene ?) Verbindung von Reverse Polish Lisp - also der umgekehrten polnischen Notation (UPN) in Verbindung mit LISP, einer Programmiersprache, die überwiegend bei der objektorientierten Programmierung und der Beschreibung und Darstellung von künstlicher Intelligenz eingesetzt wird. Sogenannte Expertensysteme auf PC's sind häufig in LISP programmiert oder in einer verwandten Sprache) soll an dieser Stelle nichts anderes bedeuten als ein unmittelbar ausführbares Objekt, das der Rechner nicht normal darstellen kann und somit als EXTERNAL maskiert.

Warum es dann doch gelingen ist mit dem gleichen Rechner seine eigenen Geheimnisse zu entschlüsseln liegt an der rechnerinternen Speicherung seiner Informationen. Dort wird nämlich so gut wie alles in Stringform gelagert und bei Bedarf z.B. für die Anzeige wieder

umgewandelt.

So wird ein angezeigtes EXTERNAL möglicherweise als folgender String gelagert: "88130".

Wenn ich jetzt verrate, daß Informationen im Rechner immer verkehrt herum gespeichert werden, meist 5 Nibbles lang sind und einer bestimmten Adresse zugeordnet werden können ... Halt, sehe ich da nicht irgendwo das letzte PRISMA (2-3/91) herumliegen mitsamt seiner wunderbaren Adressliste von Derek S. Nickel ??

Volltreffer - war's schwer ??

Ich verrate bestimmt kein Geheimnis, wenn jetzt alle im Bereich von #3187h und #3189h fündig geworden sein sollten...

Sollten dann nicht die Objekte in der Liste des Stacks das können, was sie anhand der Listenadresse versprechen?

Aber da ist doch wieder so ein komisches Zeichen: < 2h >

Was sind das denn schon wieder für komische Klammern um eine Hexzahl ?

Sie ist ein SystemInteger, also eine Zahl, die das Betriebssystem direkt versteht. In unserem Fall bedeutet sie eine Einsprungstellenanweisung für den Rechner an einer bestimmten Stelle ein Objekt zu holen (GET). Aber warum dieses sofort wieder löschen ? Ein nachfolgendes DROP macht doch irgendwie keinen Sinn.

Und doch ist es so, daß der SYSEVAL-Befehl ein zusätzliches Element mit zurück in den Stack bringt, wie man bei einer schrittweisen Kontrolle obiger Schleife leicht nachkontrollieren kann. Dieses zusätzliche EXTERNAL ist eine rechnerinterne Information, die dem Objekt noch anhängt.

Bein einer weiteren Verwendung würde sie eine Überprüfung an einer falschen Stelle im Rechner veranlassen.

Und unser treuer Wächter Namens HP48SX entdeckt sie natürlich sofort und schlägt laut Krach.

Günstigenfalls eine einfache ERROR-Meldung, leider jedoch ein recht ungnädiges MEMORY LOST

oder mit etwas Glück nur ein Software-RESET.

Und all dies für ein Objekt, das wir garnicht mehr benötigen, da doch schon bereits das erste EXTERNAL unsere Wünsche befriedigen kann... Also doch DROP - Einsicht ist schon immer der bessere Teil der Klugheit gewesen...

Wir haben jetzt immer noch die Liste mit den EXTERNALS im Stack stehen.

Mal sehen, was das zweite Programm mit dieser folgenden Liste macht:

```
{ →prg « # 54AFh SYSEVAL
# 5445h SYSEVAL » }
```

löst sie auf und packt drei EXTERNALS in den Stack 1:

```
{ < 2h > EXTERNAL
EXTERNAL }
```

Das soll ein Programm sein ?

Wo sind die üblichen Programmbegrenzer ?

Sie werden einfach nicht gebraucht - deshalb sind sie nicht vorhanden. Der Rechner hat interne Programmbegrenzer.

Wollen wir einmal sehen, ob das alles so stimmt, wie ich es gerade behauptet habe. Dann müßte doch diese merkwürdige Anordnung in der Lage sein das gleiche zu leisten, was SYSRCL auch tun kann. Also das Objekt hinter einem Kontrollnamen wegspeichern und gleich testen:

F55Dh und die Ausführung der neuen Variablen sollten jetzt eigentlich mit einem Applaus des Rechners belohnt werden..

Der hat sich bestimmt verschreiben, APPLY, er kann noch nicht einmal richtig gratulieren..

Aber das Ergebnis stimmt, #F55Dh SYSRCL fördert ebenfalls ein APPLY zutage. Der unmittelbare Vergleich der beiden Routinen zeigt uns Erstaunliches:

Das gleiche Ergebnis wird anstelle mit 56,5 Bytes mit nur 17,5 Bytes erreicht und anstelle von ca. 2,6/100 Sekunden in ca. 1,5/100 Sekunden ausgeführt. Dies zeigt den Vorteil, wenn man auf interne Adressen zurückgreifen kann.

Die Schnelligkeit wird erreicht, da hier auf interne Prüfroutinen verzichtet wird, die Kürze von dem dreimaligen Fehlen von SYSEVAL = 7,5 Bytes und dem Fehlen der Begrenzungszeichen = 5 Bytes.

Der Rest der eingesparten Bytes sind nicht benötigte interne Informationen.

Ähnliches kann natürlich auch bei längeren Programmen erreicht werden.

Doch hat diese Art der Programmierung auch seine Nachteile.

Einer davon ist es, daß sich die solcherart gewonnenen Objekte nicht mehr mit VISIT verändern lassen und schlimmer noch, diese werden durch VISIT auch noch entschärft!

Hier hilft nur ein Abbrechen mit der ATTN-Taste während eines VISIT anstelle von ENTER und Ansehen der Objekte auf eine gefahrlose Art und Weise.

Programmierhilfen stehen weiter unten - diese können jedoch erst sinnvoll angewendet werden, wenn ich noch einige wichtige Routinen vorgestellt habe.

Der Erfolg mit SYSRCL sollte sofort mit den beiden SYSEVAL-Adressen ausprobiert werden, die die Funktion →PRG ausmachen. Wir haben nun zwei neue Programme unter dem alten Namen.

Wenn jetzt erst einmal die große Entdeckungsreise durch die #-Adressen beginnen sollte, hier noch schnell ein kleiner Trick:

Zu unbedachtes Auslösen von EXTERNALS kann zu einem Aufhängen des Rechners führen.

Dies ist daran erkenntlich, daß der Rechner zwar an ist, jedoch auf keinen Befehl mehr reagiert.

Nur keine Panik in dieser Situation. Unter dem Rechner befinden sich die beiden oberen Gumminoppen als Füße, die problemlos aus ihrer Vertiefung heraus gehoben werden können, man macht dabei nichts kaputt. Darunter befinden sich jeweils ein kleines Loch, das mit einer kleinen Nadel oder auch einer Büroklammer durchstoßen werden kann. Dadurch wird ein Hardware-

RESET ausgeführt.

Der Rechner schaltet sich aus.

Nach dem Einschalten fragt er regelmäßig wieder an, ob der Speicher wiederhergestellt werden soll. Diese Frage sollte man besser verneinen, ein unbedachtes Spielen mit EXTERNALS kann sogenannte Zeitbomben im Rechner hinterlassen:

Tastenfunktionen können anders belegt sein, abgespeicherte Programme können verändert worden sein.

Also lieber alles wieder von vorne anfangen, als später nicht mehr wissen, was man alles angestellt hat und welche unverhofften Ereignisse durch beschädigte Daten bzw. Programme verursacht werden. Deshalb zu Anfang der Tip mit der Speicherkarte, die wird bei einem Hardware-RESET nämlich nicht zurückgesetzt, wenn sie nicht mit MERGE als interner Speicher deklariert worden ist.

Nach dem oben genannten Erfolg sind wir nun in der Lage, einen großen Teil des benötigten Programms zu schreiben. Die erste Fassung orientiert sich dabei an folgenden Gedanken:

unnötiger Datenmüll soll vermieden werden:

Darum erzeugt das Programm alle benötigten Unterprogramme selbst und löscht diese nach getaner Arbeit wieder;

Ergebnisse müssen in der später benötigten Reihenfolge wieder zur Verfügung stehen, eine Liste leistet das Erforderliche;

das Programm sollte selbst erklärend sein, so daß man es immer richtig anwenden kann.

Dies führt zu folgendem ersten Teil:
RUM→PRG

```
« { } 'liste' STO
« WHILE "#Adr ?↓
  Ende = # Oh !" { "#
  α } INPUT STR→ DUP
  # Oh ≠ REPEAT SYSRCL
  add END DROP »
'RPL' STO
« "User-Code ?" { "{"
  α } INPUT STR→ OBJ→
```

```
1 SWAP START DEPTH
  ROLL add NEXT »
'USR' STO
« →obj
« liste OBJ→ obj
  SWAP 1 + →LIST
  'liste' STO »
'add' STO
« liste →PRG "Name
für PRG ?" { ",'" α
} INPUT STR→ STO
  tmenu »
'→prg' STO
« {{ "RPL"
  « RPL »
  } { "USR"
  « USR »
  } "" "" { "→PRG"
  « →prg »
  } { "QUIT"
  « { liste RPL USR
  add tmenu →prg }
  PURGE 1 MENU »
  }} TMENU CLLCD
"Starte RUM→PRG mit↓
RPL, USR o. $→C↓
  Progformen = →PRG↓
  Ende = QUIT"
  3 DISP 2 FREEZE »
'tmenu' STO tmenu
»
```

(↓ ist das Newline-Zeichen, das man mit der blauen Shift-Taste und "." eingibt. Es ist allerdings im Display nicht zu sehen, da es ja ein Springen des Cursors in die nächste Zeile bewirken soll, a new line, für alle, die sich noch an die letzten Englisch Stunden erinnern können.)

Das Überraschende an diesem Programm ist wahrscheinlich, daß hier ganz normaler Usercode eine Rolle spielte, also ganz normale DROPS und SWAPs.

Warum dies ?

Vorhin haben wir doch ganz deutlich die Überlegenheit der neuen Programmieretechnik gesehen.

Da hier aber Strings intern auch nicht anders als Strings abgespeichert werden, können wir diese der Einfachheit halber auch gleich so eingeben. Wer will denn schon die Variablenamen erst ausrechnen und umformen, wenn sie auch richtig eingegeben werden können und

nicht mehr Platz benötigen als sonst auch ?

Also Gott sei Dank doch noch etwas Bekanntes ...

Die einzelnen Programmteile sind unterschiedlich gestaltet, da überwiegend mit #Adr. gearbeitet werden wird, ist dieser Teil als Endlosschleife gestaltet mit der Abbruchbedingung `x = # 0h`.

Sollte dieser Wert tatsächlich im Programm einmal eine Rolle spielen, so kann er auch über Usercode eingegeben werden. Auch letzterer kann innerhalb einer Liste eingegeben werden, sofern er auch so in der späteren Reihenfolge wieder auftreten soll. Ansonsten bitte immer die notwendige Reihenfolge beachten.

Der Teil des Programms, der letztlich Maschinencode erzeugen soll ist etwas schwieriger zu programmieren. Hier werden wir uns einiger Tricks bedienen, wie sie im Umgang mit SYSEVAL-Objekten manchmal notwendig sind.

Die Grundbausteine des Programms stammen vom 48'er Papst persönlich: Dr. Bill Wickes himself.

Sein in verschiedenen Mailboxen veröffentlichtes Programm ASC→ wird durch ein Stringunterteilungsprogramm erweitert, da dieser Mini-Assembler eine etwas merkwürdige Eigenart aufweist. Er benötigt ein Newline (↵) nach jeden 64. Zeichen innerhalb eines Strings, der verarbeitet werden soll.

Zunächst werden wir die Schleife programmieren, um einen String entsprechend den erwarteten Eingaben für den Mini-Assembler vorzubereiten. Die Schleife lautet:

```
"String ?" { α } INPUT
"0000" + (*) DUP SIZE 64
/ FLOOR → str x *
" 1 x FOR i str i 64 x
DUP 63 - SUB "J" +
NEXT str DUP SIZE x
64 * 1 + SWAP SUB 1
x START + NEXT
```

»

»

(*)

Dies geschieht, um dem Minias-

sembler eine Prüfsumme vorzugaukeln, da er nur mit einer korrekt berechneten Prüfsumme arbeiten will; glücklicherweise hat er jedoch einen Fehler in der Programmierung. Er akzeptiert die Prüfsumme "0000" für jedes Objekt (er rechnet dann zwar eine falsche Kontrollprüfziffer aus, diese wird aber sowieso nicht benötigt).

*
Dieser Schleifenteil erscheint unmittelbar vor dem Assembler \$→C (String zu Code)

Der Mini-Assembler erhält folgenden Code:

```
" RCWS → w
" 16 STWS # 0h NEWOB
SWAP DUP SIZE DUP 65
/ IP - 4 - # 18CEAh
SYSEVAL "" OVER
# 61C1Ch SYSEVAL
SWAP ass w STWS
```

»

Neu sind offensichtlich die Adressen # 18CEAh und # 61C1Ch. Die erste formt aus einer realen Zahl einen SystemInteger (SystemBinary = SB), die zweite erzeugt in einem String entsprechend dem vorgegebenen SystemInteger "Platzhalter". Die letztere ist beim Ausprobieren stets mit Vorsicht zu genießen, da sie falsche Stackeingaben erbarungslos mit MEMORY LOST behohnt.

Immer noch nicht der ersehnte Assembler - Kerl, nun mach doch endlich. Die Variable `ass` schreit förmlich nach einer Lösung. Hier ist sie:

'`ass`' ist eine ellenlange Hexzahl, die erst künstlich erzeugt werden muß.

Es ist am einfachsten dafür ein kleines Hilfsprogramm zu schreiben, da leider nicht davon auszugehen ist, daß man beim ersten Mal alles richtig abtippt.

Es sind zwölf Hexzahlen miteinander zu verknüpfen, die Verknüpfung selbst erfolgt auch durch einen SYSEVAL-Befehl, der am besten innerhalb einer Kurzvariablen abgespeichert wird - dies erspart Tipparbeit.

Also:

'A' = « # 518Ah SYSEVAL »

«

```
# 6641F8000AF02DCCh
# 130480679BF8CC0h A
# 19610313418D7EA4h A
# 7134147114103123h A
# 5F6A971131607414h A
# 12EA1717EA3F130Ch A
# 280826B3012808F4h A
# 6B7028080BEE9091h A
# BE5DC1710610C512h A
# 705D00003431A078h A
# 3D8FA26058961431h A
# 312B0514h A
```

»

Bevor jetzt dieses Programm aktiviert wird hier noch einige Informationen:

Im Stack wird scheinbar die erste Zahl stehen bleiben, eine Veränderung ist nur zu bemerken, wenn das Zwischenergebnis jeweils durch einen Ausdruck mitprotokolliert wird, im Ergebnis wird eine Zahl an die andere gefügt. Es findet also keine Addition oder Ähnliches statt.

Die entgültige Zahl bitte unbedingt vor dem Abspeichern mit DUP BYTES überprüfen. Das Ergebnis sollte 101 Bytes betragen und die Prüfsumme # EBEh aufweisen.

Das Ergebnis steht jetzt immer noch im Stack. Wer jetzt versucht diese Zahl mittels VISIT zu manipulieren, der wird sehr enttäuscht feststellen müssen, daß das Ergebnis seiner Bemühungen stets ein INVALID SYNTAX ist, nichts geht mehr ...

Aber es hilft ein kleiner Trick. Als nächste Zeile müssen wir jetzt ein Leerprogramm in den Stack bringen:

Also ein « ».

Dieses jetzt mit # 54AFh SYSEVAL auflösen, den SystemInteger einfach DROPen und mittels ROT SWAP die ellenlange Hexzahl zwischen den Programmgrenzen platzieren. Bevor jetzt das Programm fertig ist schnell noch # 18F23h SYSRCL SWAP interaktiv ausführen.

Eine Leerliste { } in den Stack bringen und alle 4 Elemente des Stacks in die Leerliste addieren.

Auf diese Liste jetzt die Funktion →PRG anwenden und das Ergebnis als 'ass' abspeichern.

Na, wer hat gemerkt, daß die Programmbegrenzer überflüssig sind ?

Jetzt gewinnt unser Programm RUM→PRG langsam seine endgültige Form.

Zunächst ist als Zwischenschritt als dritten selbstständigen Programmteil hinter

```
"UserCode ? ... » 'USR'
STO
```

der Teil von oben

```
« "String ?" { α } INPUT
"0000" + ...
```

bis zum letzten NEXT einzufügen.

Hinter NEXT aber nicht sofort die Programmbegrenzer abschließen, sondern noch ein

```
$→C add CLEAR
```

einfügen, jetzt

```
» '$→c' STO (wirklich kleines
c!). Die PURGE-Liste kurz danach
ist um die Variable $→c zu ergänzen.
```

Wer will, der kann den zweiten Leerstring durch folgende Anweisung ersetzen:

```
{ "NEXT" « { } 'liste'
STO tmenu » }
```

Es sei jedoch davor gewarnt, daß ein unbeabsichtigtes NEXT immer den Inhalt der Sammeliste zerstört und nicht den nächsten einzugebenden Befehl ermöglicht - dafür sind die Schleifen innerhalb der Programmierung gedacht. Der Maschinencodeteil erhält keinen Schleifenteil.

Erstens ist kaum davon auszugehen, daß eine Variable für sich mehr als einen Maschinencodeteil enthalten wird. Und wenn, dann können diese ja ebenfalls in der Liste gesammelt werden.

Zweitens bedarf das Eingeben eines Strings erhöhter Aufmerksamkeit und größte Genauigkeit, da der Assembler auf jeden Fall Code-Objekte erzeugt, die dann bei falscher Programmierung stets ein MEMORY LOST erzeugen.

Das bedeutet, daß immer ein syntaktisch korrektes Objekt gebildet werden muß. Für Listen und Programme bedeutet dies, daß sie

nach dem individuellen Anfang (hier "47A20" und "D9D20") immer mit "B2130" abzuschließen sind. Auch sind alle verschachtelten Abfragen immer selbstständige Programmteile, die auch mit Programmanfang und -ende erscheinen müssen.

Also

```
IF ----
```

```
THEN "D9D20 .... B2130"
```

```
ELSE "D9D20 .... B2130"
```

```
END.
```

Dies nur als kurzes Beispiel. Es gibt viele Dinge zu beachten, die später geklärt werden sollen. Zunächst werden wir jedoch so schnell wie möglich das Programm selbst verbessern. Ja, Schlimmer noch, wir müssen es entschärfen, da wir auf einer Zeitbombe sitzen!!

Es ist falsch !

Wie war das, die ganze Mühe umsonst ?

Halt, nicht löschen. Es müssen nur einige Befehle erneuert und verbessert werden. Als erstes ist SYSRCL fällig.

Diese Funktion ist rottenfalsch und höchst gefährlich !

Erinnert Ihr Euch an den internen Pointer, den wir mit DROP gelöscht haben ?

Etwas Ähnliches passiert passiert immer noch mit dem geholten RPL-Objekt. Dieses Objekt kann immer noch mit falschen Ergebnissen behaftet sein - es wird nämlich bei SYSRCL kein NEWOB durchgeführt - eine Funktion, mit der man eigentlich nie sparen sollte, wenn man mit RAM- bzw. ROM Adressen manipuliert.

Es wäre einfach, an geeigneter Stelle ein NEWOB zu ergänzen, aber wir wollen als Test für unser neues Programm eine völlig neue Version mit Maschinencode erstellen - sie ist erheblich schneller, 100% richtig, dafür aber leider geringfügig größer.

Sie wird jedoch unbedingt in dieser Form für künftige Anwendungen benötigt.

Also das neue Programm RUM→PRG aufrufen, über die neue Menüleiste den Maschinencodeteil

aktivieren (\$→C) und die Stringaufforderung mit

```
"D9D20ECE81D5040D9D20756
60CCD20E1000147137179143
1371418D39E60B2130B2130"
beantworten.
```

Na, wer hat aufgepaßt ?

Es ist natürlich mehr als nur ein Maschinencodeteil, beginnt doch ein reines Maschinencodeteil grundsätzlich zu Ehren des Clubs immer mit "CCD20" (es muß sich also um ein sehr altes Clubmitglied handeln..).

Es stimmt, der Assembler kann komplette Programme erstellen, nicht nur Codeobjekte.

Aber bitte, nutzt ihn nur in dieser ersten Anwendungsphase - er ist ebenfalls falsch, auch ihn werden wir schrittweise ersetzen.

Das eben gewonnene Programm speichern wir unter dem Namen 'A→N' (Adress zu Name). Wir können die Funktion unmittelbar mit SYSRCL vergleichen.

Also eine beliebige SYSEVAL-Adresse in den Stack:

Am einfachsten ist eine XLIB 002 Adresse, da sie einen Befehlsnamen in den Stack holt.

Das Ergebnis bei beiden Vorgängen (A→N bzw. SYSRCL) sollte identisch sein. Ansonsten die Stringangabe wiederholen.

(** Die Prüfwerte sollte ich vielleicht doch nachtragen: 75 Bytes für den String und # 7B85h für den reinen String, das reine Objekt hat später 35 Bytes und die Prüfziffer # 80DFh. Dies kann jetzt noch abweichen, da der Assembler noch ein überflüssiges "D9D20" und "B2130" als Beigabe versteckt hat; ich sagte ja, es ist noch falsch **).

Das heißt, wenn wir ihn ersetzt haben, einfach den String noch einmal eingeben. Aber keine Angst, wir bekommen den String gleich ganz komfortabel in den Stack, ohne ihn noch einmal eingeben zu müssen.

Die nächste Aufgabe ist es, den Assembler zu ersetzen. Dabei leistet er als guter Demokrat eigene

Schützenhilfe, d.h. er liefert alle Einzelteile selber, die ihn zusammengekommen ersetzen werden.

Wir erstellen zunächst Einzelwerkzeuge wie 'vadr' 'poke', die wir dann später zu '\$→V' zusammenfassen werden.

Das Programm 'vadr' dient dazu, bei einem vorgegebenen Variablennamen im Stack die zugehörige #-Adresse im RAM zu ermitteln.

```

$→V
D9D20FDE811192063000D9D2
08813059230C2A2050002C2
308B9C1339201000
0000000001090DA1AEC81C1
C1659230DCC023223084E204
06716462784E2040
07F6B656B2130B213006BB
Name Var : $→V

poke
D9D20FDE818AB46CCD207800
0808F8FB9760143133179147
134D2203103D7131
174143131174143818F84819
F0174215C114F90380818F28
2117115C0160CC53
E8F2D760179E7E780808DE42
305B213070EC
Name Var : poke

vadr
D9D20ECE81B2040D9D20E4A2
051000000000000000000000
2A13223004B02CCD
20D1000147174E7143133179
8DFFE50B2130B21303848
Name Var : vadr
    
```

Prüfziffern:
 'vadr' 110 Bytes # 4C9Fh
 'poke' 165 Bytes # 2541h
 '\$→V' 151 Bytes # B701h

Eingabesyntax:
1: # Adresse
vadr
2: String
1: # Adresse
poke

Neben dem Aufruf von 'vadr' und 'poke' enthält '\$→V' noch einige Prüfroutinen, die genaue Erklärung kommt etwas später.

Jetzt wollen wir nur schnell ein Programm überprüfen, bevor die neue Variable im RUM→PRG ihren korrekten Dienst aufnehmen kann.

Den folgenden String bitte ebenfalls mit dem neuen Programm prüfen:

```

"D9D20E1632DBBF19363B2130"
**
Die Stackangaben sind immer
2: String
1: 'Varname'
Command Zeile: $→V
**
    
```

Das Ergebnis müßte jetzt ein Programm folgenden Inhalts sein:
 « DUP »

Wenn dies zutrifft, dann ist die Umwandlungsroutine richtig und sie kann getrost in das Programm

```

48SX RUM→PRG
PROGRAM
< { } 'prl' STO
<
  WHILE
  "#-ADR ?
  Ende = # 0h !"
  { "#" α } INPUT OBJ→ DUP
  # 0h ≠
  REPEAT SYSRCL add
  END DROP
  > 'RPL' STO
  < "User-Code ?" { "("
  α } INPUT STR→ OBJ→ 1
  SWAP
  START DEPTH ROLL add
  NEXT
  > 'USR' STO
  < "string ?" PROMPT
  "0000" + DUP SIZE 64 /
  FLOOR → str x
  < 1 x
  FOR i str i 64 *
  DUP 63 - SWAP SUB "
  +
  NEXT str DUP SIZE
  x 64 * 1 + SWAP SUB 1 x
  START +
  NEXT
  > '$→C' add CLEAR
  > '$→C' STO
  < → obj
  < prl OBJ→ obj SWAP
  1 + →LIST 'prl' STO
  >
  > 'add' STO
  < prl →PRG
  "Name für PRG ?" { "" α
  } INPUT STR→ DUP ROT
  SWAP STO VBR thmenu
  > 'prg' STO
  < { { "rpl"
  < RPL
  > } { "usr"
  < USR
  > } { "$→C"
  < '$→C'
  > } { "" { "→PRG"
  < 'prg'
  > } { "QUIT"
  < { prl RPL USR cnt
  '$→c add thmenu →prg }
  PURGE 1 MENU
  > } } TMENU CLLCD
  "starte RUM→PRG mit
  RPL USR o. '$→C,
  Progformen = →PRG
  Ende = quit"
  $ DISP 2 FREEZE
  > 'thmenu' STO thmenu
  >
  Bytes : 1001.5
  Checksum : # D79Fh
    
```

RUM→PRG eingebaut werden. Das Programm kann dann die Fassung gemäß Ausdruck erhalten.

Möglicherweise kommt es während des Ausprobierens der neuen Funktion zu einigen merkwürdigen Erscheinungen. So kann ein Software-Reset durchgeführt werden, daran zu erkennen, daß der Rechner in das HOME-Menü springt und die Flaggrundstellung einnimmt. Gleichwohl müßte die neue Variable korrekt gebildet worden sein. Dies muß dann durch das Zurückgehen in das alte DIR überprüft werden.

Diese Erscheinung scheint darin begründet zu sein, wenn man eine nicht originale Speichererweiterungskarte benutzt. Da scheinen bei einigen preiswerten Karten ein paar Kleinigkeiten nicht 100%-tig abgestimmt zu sein. Falls ein Totalausfall erfolgt (MEMORY LOST), so war wahrscheinlich irgendwo ein Fehler entweder in den drei Teilprogrammen selbst oder in der Eingabe des oben genannten Strings. Also alles bis hier noch einmal überprüfen und vergleichen!

In wenigen Fällen kommt es zu Schwierigkeiten, wenn nach einem MEMORY LOST und anschließendem RESTORE die Funktion '\$→V' per Hand aufgerufen wird. Es erfolgt dann ein weiteres MEMORY LOST. Wenn die gleiche Funktion per Programm mit den gleichen Parametern aufgerufen wird, funktioniert sie dagegen immer korrekt. Dieses Phänomen kann ich derzeit nicht eindeutig zuordnen. Ich nehme an, daß es etwas mit meinen Speichererweiterungskarten zu tun, die möglicherweise nicht immer korrekt arbeiten.

(Kann ich mir zwar als Elektroniker nicht ganz vorstellen, weil wenn so eine Karte nicht richtig funktioniert, so wäre dies schon beim Speichertest beim Einbinden der Karte der Fall, so etwas ist nicht softwareabhängig. mm) Sollte jedoch auch bei Besitzern von Originalkarten die gleichen Phänomene auftreten, so ist das

adjoin fügt ein Objekt in Stack 1: einer Liste in Stack 2: nur dann zu, wenn es sich innerhalb der Liste um gleiche Objekttypen handelt.

adjoin

```
D9D20FDE8111920050001945
3B21303AC1
```

Name Var : adjoin

cdr

```
D9D20ECE8112040351500004
0C6150B21304EE3
```

Name Var : cdr

A+N

```
D9D20ECE81D5040D9D207566
0CCD20E10001471371791431
371418D39E60B213
0B2130FD08
```

Name Var : A+N

cdr gibt einen String oder eine Liste aus Stack 1: ohne das erste Zeichen bzw. ohne erstes Objekt zurück.

ndup

```
D9D20FDE811192010000D9D2
0AEC81073E544230B2130B21
303388
```

Name Var : ndup

keep

```
D9D2043C81D9D2095450B2D4
30D470C4130E62306B316794
7088B26B2130B213
06993
```

Name Var : keep

sxch

```
D9D20FDE8199040D9D20AEC8
166226D9D20C2A1611920302
00A3836B21303223
0AEC8166226D9D20C2A16119
2030200A3836B2130CA1306C
426A6526C413038D
30D9D20C2A161192030200A3
836B2130CA13091D3007916C
A13038D30CB91632
23047A2003D4303D43B21300
D4706B316523307E316E9330
7E316FED30523306
B316E933079470B2130B2130
E166
```

Name Var : sxch

ndup erwartet das Objekt in Stack 2: und die Vorgabe, wie oft es kopiert soll in Stack 1; also nicht Objekt +1!

keep löscht den gesamten Stackinhalt oberhalb der in Stack 1: vorgegebenen Zeilennummer.

sxch ist ein kleines Bonbon.

Alle, die schon immer ein gutes Programm haben wollten, um Zeilen untereinander zu tauschen, werden begeistert sein.

(Siehe auch die praktische Anwendung im Programmdebugger anstelle von CLEAR oder DROP).

Diese drei Kurzprogramme sind beinahe unentbehrlich, besonders sxch in Verbindung mit SYSEVAL Objekt.

Ebenso in Verbindung mit dem Toolkit optimal, für alle anderen erst, wenn der Databrowser fertig ist.

Ein kleines Beispiel zwischendurch, daß RPL nicht immer der Weisheit letzter Schluß ist:

Das Programm split kann man auch normal programmieren, der Zeitverbrauch liegt hier bei 6/100s statt 3/1000, dafür benötigt die normal programmierte Version nur 59 Bytes, die RPL-Version 165!

SPLIT-NORMAL

```
< → x
  < DUP SIZE x 1 + SWAP
  SUB LASTARG DROP 1 - 1
  SWAP SUB
  >
  Bytes: 59
```

Der wirklich letzte Teil der Vorstellung sollen zwei Programmwürfe werden, wie einigermaßen komfortabel auf die neuen Programmobjekte zurückgegriffen werden kann, um sie später noch verbessern zu können ohne wieder ganz von vorne anfangen zu müssen.

Die erste Version (SYSOBJCH) ist als Idee wirklich nur in der Struktur vorgegeben. Vielleicht erbarmt sich ja jemand und stellt für alle nicht Toolkit-Besitzer ein tolles Programm zur Verfügung?

Die zweite Version (PROGDEBUG) ist eine mögliche Verarbeitung mittels des Databrowsers aus dem Toolkit, der einzige Wehrmutstropfen im Textausdruck ist die Tatsache, daß ich eine leicht veränderte Library benutze und zur Kenntlichmachung für mich selbst die Befehle etwas umbenannt habe! Hier ist jedoch nur eine einzige Ab-

DATUM: SAT 17.08.91 14:27:49
HP 48 SX VERSION E. (C) 1990
Autor: G.Hoppen, Wallenhorst

48SX SYSOBJCH

```
PROGRAM
< "Name Var ?" { "" α }
INPUT OBJ→ RCL # 54AFh
SYS DROP { ( "A+N"
  < A+N
  > } { "SXCH"
  < SXCH
  > } "" { "→PRG"
  < DEPTH { } 1 ROT
  START +
  NEXT →PRG
  > } { "QUIT"
  < CLEAR 1 MENU
  > } } CLLCD TMENU
  >
  Bytes : 226
  Checksum : # 9285h
```

PRGDEBUG

```
< "Name Prg ?" { "" α }
INPUT OBJ→ DUP "" SWAP
RCL DUP BYTES ROT
# 54AFh SYS {
  "Chk" : "Bytes: " } 'L'
STO # 18DBFh SYS DUP 1
SWAP
FOR i L OBJ→ i " : " +
SWAP 1 + →LIST 'L' STO
NEXT 3 + →LIST L SWAP
{ "VIEWKEY" "NULLKEY"
  "SXCH" "INS" "DEL"
  "QUIT" } → LBL
  < LBL { 2 1 5 -1 }
  "x -DEBUG" "x" 7 ROLL
  →STDSTR REPLACE +
  < STD 700 .5 BEEP
  DBRX ABS
  CASE DUP 3 ==
    THEN DROP → x
    < SWAP OBJ→
  DUP "änderung ?" PROMPT
  SWAP x - 2 + 1 SXCH DROP
  →LIST L SWAP LBL 4 ROLL
  kor
  >
  END DUP 4 ==
  THEN DROP ROT
  SWAP 1 - SPLIT
  "Einfügung ?" PROMPT +
  SWAP + L DUP SIZE 2 -
  " : " + → 'L' STO L SWAP
  LBL 4 ROLL kor
  END DUP 5 ==
  THEN DROP ROT
  SWAP 1 - SPLIT SWAP CDR
  + L DUP SIZE 1 - 1 SWAP
  SUB 'L' STO L SWAP LBL 4
  ROLL kor
  END DUP 6 ==
  THEN DROP2 DROP
  DUP SIZE 4 SWAP SUB →PRG
  END
  END
  > 'kor' STO kor {
  kor L } PURGE
  >
```

weichung zu beachten:

DBRX entspricht dem Befehl DBR aus dem Original-Toolkit, also bitte beim Abtippen beachten.

Wer eine noch elegantere Lösung weiß, der soll mich bitte anrufen.

Die Arbeitsweise ist wie folgt:

VIEW ist eine Schnellübersicht des

DATE: Fri 16.08.91 21:10:55
 HP 48 SX VERSION E. (C) 1990
 Autor: G.Hoppen, Wallenhorst

```
48SX RUM→PRG
PROGRAM
« { } 'prl' STO
«
  WHILE
  "#-ADR ?
  Ende = # 0h !"
  { "#" α } INPUT OBJ→ DUP
  # 0h ≠
  REPEAT A→N add
  END DROP
  » 'RPL' STO
  « "User-Code ?" { "(" α
  » INPUT STR→ OBJ→ 1
  SWAP
  START DEPTH ROLL add
  NEXT
  » 'USR' STO
  « "String ?" PROMPT
  'obj' $→V 'obj' DUP RCL
  add PURGE
  » '$→C' STO
  « → obj
  « prl OBJ→ obj SWAP
  1 + →LIST 'prl' STO
  »
  » 'add' STO
  « prl →PRG
  "Name für PRG ?" { "" α
  » INPUT STR→ DUP ROT
  SWAP STO VBR thmenu
  » 'prg' STO
  « { "prl"
  « RPL
  » } { "usr"
  « USR
  » } { "$→C"
  « $→C
  » } { " " { "→PRG"
  « →prg
  » } { "QUIT"
  « { prl RPL USR cnt
  $→C add thmenu →prg }
  PURGE 1 MENU
  » } } TMENU CLLCD
  "starte RUM→PRG mit
  RPL USR o. $→C,
  Programmen →PRG
  Ende = Quit"
  3 DISP 2 FREEZE
  » 'thmenu' STO thmenu
  »
  Bytes : 853
  Checksum : # BFE7h
```

Vorläufige verbesserte Endversion von RUM→PRG:

Sysrcl → A→N
 \$→C → \$→V

Es muß eigentlich *thmenu* heißen, *thmenu* funktioniert aber auch.

cnt ist eine Hilfsvariable für VBR:

C = Bedingung für
 WHILE cnt REPEAT ...

Durch die vielen Umbenennungen stimmen die Variablennamen mit dem geschriebenen Text nicht mehr überein:

'prl' entspricht 'liste' und sollte entsprechend verbessert werden.

Die Arbeitsweise ist wie folgt: VIEW ist eine Schnellübersicht des Inhalts bei der angezeigten Seite; ansonsten immer die Menütaste betätigen und dann bei der entsprechenden Anweisung reagieren.

SXCH dient dazu, den Inhalt einer Stackzeile durch eine andere zu ersetzen. Aus diesem Grund sind auch keine Inputbefehle verwendet worden.

Jetzt können Zwischenschritte absolviert werden.

(z.B. RCL, A→N oder andere, bevor das Programm fortgesetzt wird.)

Achtung:

Das letzte Objekt kann leider nicht gelöscht werden. Weder mit DEL ist es sinnvoll, mit SXCH etwas hineinzusetzen. Hier handelt es sich um einen Programm- bzw. Denkfehler bei mir. Wer also Abhilfe weiß ...

→PRG hinterläßt das veränderte Objekt als neues Programm im

VADR

```
D9D20ECE81B2040D9D20E4A2
051000000000000000000000
2A13223004B02CCD
20D1000147174E7143133179
8DFFE50B2130B21303848
```

Name Var : "VADR"

A→ST

```
D9D20ECE81D5040D9D207566
0CCD20E10001471371791431
371418039E60B213
0B2130FD08
```

Name Var : A→ST

Stack 1: Jetzt kann es abgespeichert werden.

Die Prüfziffer hinter dem B2130 ist immer die richtige, braucht aber nicht eingegeben zu werden.

Vergleiche die neuen Namen:

'vadr';

A→ST wurde später zu A→N, d.h. Adresse zu Name und somit Ersatz für SYSRCL.

Bitte seht mir nach, daß ich nach diesem Marathon keine übergroße Lust verspürte auch die beiden letzten Programme optimal zu gestalten.

Sie taten ihren Dienst während des Schreibens dieses Artikels, das war's.

In Verbindung mit einem DECOMPILER ist das zweite Programm ein

wirklich überzeugendes "Erste-Hilfe-Programm", ich erwarte Eure Verbesserungsvorschläge.

Laßt von Euch hören, wie es Euch beim ersten Mal mit der neuen Art der Programmierung ergangen ist; wie gesagt, Verbesserungsvorschläge sind jederzeit willkommen.

Georg Hoppen

Hubertusring 5

D4512 Wallenhorst

Fortsetzung von Seite 22

gen steigt) nicht nachhaltig ändert oder falls wie bisher ersatzweise die vielen Einsendungen der CCD-Mitglieder nicht bei mir eintreffen könnte im Jahr 1993 die Disketten-Serie eingestellt werden. Ich werde dann aber als Ergebnis der letzten Jahre allen CCD-Mitglieder vielleicht die kleinste aber die "einzig wahre" Programmsammlung für den Atari-ST präsentieren können.

Um jetzt schon einem schnellen Ende der Atari-Gruppe vorzubeugen werde ich ab 1992 Programme für den HP-28 bzw. HP-48 in einer kleinen HP-Ecke mit auf die Disketten packen die ich leider mangels HP-Rechner nicht testen kann.

CCD-Mitglieder seien hiermit aufgerufen mir HP-Programme auf 3 1/2" DD Disketten (MS-DOS oder Atari-ST Format) zur Weitergabe zuzusenden. MS-DOS Fans sei hier auch gleich gesagt, daß ich diese Atari-Disketten auf Wunsch auch in einer MS-DOS-Rechner lesbaren Form zusenden kann (Atari-Programme würden jedoch nicht laufen!).

Ich werde im letzten Prisma 1992 dann berichten ob und wann welche Programme eingetroffen und "verteilt" werden konnten.

In Anbetracht der ständig steigenden Preise auf dem Mailbox-Markt sind die Kosten für die Atari-Disketten mit 90DM/Jahr = 5760 kbyte! geradezu billig.

Träfen genügend aktuelle Artikel/Nachrichten/Programme bei mir ein, so könnte die Diskette wie schon zu CPM-Zeiten wieder zum Informationsübertragungsmedium No.1 werden.

Werner Müller

5000 Köln 41

Schallstraße 6

Sortierung

von Listen und Menüs

von Michael Ritter

Das Programm SORT sortiert Inhalte von Listen in aufsteigender Folge. Die Sortroutine läuft dabei wesentlich schneller als die im Benutzerhandbuch Band II, Seite 609 abgedruckte "Bubble Sort Routine". Die Geschwindigkeit liegt je nach Länge und Grad der Sortierung um den Faktor 2 bis 10 höher.

Das Programm DSORT (welches SORT aufruft), ermöglicht es, die Variablenbezeichnungen im VAR-Menü alphabetisch zu sortieren.

Das Programm SORT:

Das vorliegende Programm ermöglicht die Sortierung von Listen, deren Inhalte Zeichenketten und Variablenbezeichnungen sind.

Mit kleinen Änderungen im SORT-Listing ist eine Anwendung auf Zeichenketten und Zahlen möglich.

Die SORT Routine behandelt alle Inhalte von Listen als Zeichenketten. Dadurch ist es möglich Variablenbezeichnungen zu sortieren. Bei Anwendung auf Zahlen entstehen Ergebnisse wie 9 größer 500, da der Vergleich zeichenweise erfolgt und die Bedingung "9" > "5" erfüllt ist. Vergleiche Benutzerhandbuch II, Seite 753 (Zeichencodes).

Sollen Zahlen und Zeichenketten sortiert werden, so müssen die →STR Befehle (im Listing unterstrichen) entfernt werden. Die so geänderte Routine sortiert nun keine Variablennamen mehr und sollte deshalb einen neuen Namen erhalten. Der Aufruf der geänderten Routine als SORT durch DSORT würde zu falschen Ergebnissen führen.

Wie SORT funktioniert:

Aus der Eingangsliste werden die Elemente einzeln in eine Zielliste einsortiert. Dabei sind drei Fälle zu unterscheiden.

1. Fall: Das neue Element ist kleiner als das erste Element in der Zielliste. Daraus folgt: das neue Element wird

vor dem bisher ersten Element der Zielliste eingefügt.

2. Fall: Das neue Element ist größer als das letzte Element in der bisherigen Zielliste.

Daraus folgt: das neue Element wird an die Zielliste angehängt.

Sind der 1. und der 2. Fall nicht eingetreten, so wird das neue Element in die Zielliste eingefügt. Da die Zielliste sortiert ist, wird die neue Position mit Hilfe von Einschachtelung gefunden.

Listen die teilsortiert sind, z.B. erweiterte Listen, werden in der Regel schneller bearbeitet als unsortierte Listen, da die sortierten Teile nur umkopiert werden. Dabei ist zu beachten, daß Erweiterungen von Listen am Ende der Liste angehängt werden sollten, um zu verhindern, daß sortierte Teile in die Zielliste eingeschoben werden müssen (Fall 3). Der Rechenaufwand für Fall 3 ist größer als für den Fall 1 und 2.

Das Programm DSORT:

DSORT sortiert die Variablenbezeichnungen im Variablen-Menü in alphabetischer Reihenfolge.

Dazu wird die SORT Routine benötigt. Zunächst werden alle Variablenbezeichnungen sortiert und mit ORDER in das Variablen-Menü geschrieben. '15 TVARS' ruft alle Verzeichnisvariablen auf und bringt sie an den Anfang des VAR-Menüs. Der folgende IF Satz vermeidet einen Fehler, wenn keine Verzeichnisvariablen vorhanden sind. Falls andere Variablentypen an den Anfang des Variablen-Menüs gesetzt werden sollen, muß der Objekttyp geändert werden. (Nummern für Objekttypen Benutzerhandbuch I, Seite 102).

Beispiel: Es sollen statt Verzeichnisvariablen (Typ: 15), Programme (Typ: 8) und globale Variablen (Typ: 6) zu Beginn des Variablen-Menüs stehen.

Dazu sind im Listing von DSORT folgende Änderungen durchzuführen:

```
15 TVARS
wird ersetzt durch
8 TVARS 6 TVARS +
```

Es entsteht eine Liste in der zuerst alle Programmnamen und dann alle globalen Variablen enthalten sind. Diese wird, falls sie nicht leer ist, an den Anfang des Variablen-Menüs gesetzt. Da zunächst alle Variablen sortiert wurden, sind alle vorgezogenen Variablen ebenfalls sortiert.

Listings Translationcode 3, Benutzerhandbuch II, Seite 680.

DSORT

```
« VARS SORT ORDER
  IF 15 TVARS DUP { } ≠
  THEN ORDER
  ELSE DROP
  END »

# 57878d, 70 Bytes

SORT
« DUP SIZE → L A
  « L 1 1 SUB → Z
    « IF A 1 >
      THEN 2 A
      FOR I L I GET → V
      « CASE Z I 1 - GET
        →STR V →STR ≤
        THEN Z V +
        END
        Z 1 GET →STR V
        →STR ≥
        THEN V Z +
        END 1 I 1 - → U O
      « DO O U - 2 / U +
      IP → H
        « H
          IF Z H GET
            →STR V →STR >
            THEN 'O'
            ELSE 'U'
            END STO »
      UNTIL O 1 - U == END Z 1 O
      1 - SUB V + Z O A SUB + »
      END 'Z' STO
      » NEXT END Z » » »

# 57613d, 450 Bytes
```

Michael Ritter
Fasanenstraße 8
W5060 Bergisch-Gladbach 1

Eingespannter Einfeldrahmen Trapez - oder Rechteckform

von Ludwig Kaindl

HP41CX, 1837Bytes, 263 REGs, SIZE043

Vom Aufruf unseres M. Meyer in Prisma 1/91 angeregt, beiliegend ein HP41CX Baustatik-Programm aus meinem Fundus:

"Eingespannter Einfeldrahmen, Trapez-oder Rechteckform"

Das Programm füllt in den einschlägigen Softwareangeboten, z.B. Mücke-Software, Vieweg Programm-bibliothek, UPLE-Verzeichnis usw. eine Lücke. Der Unterzeichnete hat die im bekannten Buch "Kleinlogel, Rahmenformeln" allgemein gehaltenen Lastansätze auf Trapezlasten umgerechnet. Dies kommt der Praxis entgegen, da außer Gleichlast auch Erddruck und beliebige Dreiecklast direkt ange-setzt und kombiniert werden kann.

Eine beliebige Anzahl von Einzel-lastfällen (auch Wiederholung) wird vom Programm selbsttätig überla-gert.

Der jeweilige Stand der Summie-rung kann mit "XEQ40" ausgedruckt werden, bis mit "XEQ50" die Sum-mierung auf Null gesetzt wird.

Die symmetrischen Fälle "C" und "D" erweitern das Angebot. So kann z.B. mit einem positiven Fall "C", plus einer negativen Lastan-gabe in Fall "A" die Trapezlast am rechten Stiel berechnet werden.

Die Rechnerkapazität ist mit acht Lastfällen voll ausgelastet.

Auflagerreaktionen sind positiv, wenn der Richtungssinn den Bei-spielen entspricht.

Biegemomente sind positiv, wenn die innere Rahmenfaser ge-zogen wird.

Das Programm wird in der Praxis verwendet und dürfte fehlerfrei sein.

Es wird gewarnt, den Ablauf der Lastfälle "B" und "D" vorzeitig für andere Arbeiten abzurechnen, da zwischen Schritt 354 und 395 die Systemwerte "a" (in Reg. 05) und "h" (in Reg. 03) vertauscht sind.

Gegebenenfalls beide Registerinhalte von Hand korrigieren, oder den Ablauf bei Schritt 01 neu beginnen. In gleicher Aufmachung wird von mir ein weiteres Programm "Zweigenkrahmen" verwendet, welches die Berechnung des Biegemomentenverlaufes von trapezbe-lasteten Stäben ermöglicht.

Formeln zum Programm EGR
Eingespannter Rahmen

Festwerte:

JR = Trägheitsmoment Riegel,
JS = Trägheitsmoment Stiel.

$$k = \left(\frac{JR}{JS}\right) * \frac{\sqrt{a^2 + h^2}}{b} \quad \alpha = \frac{a}{l}; \quad \beta = \frac{b}{l}$$

$$K1 = 2 * k + 3; \quad K2 = k * (1 + \beta) + \beta * (1 + k);$$

$$N1 = k + 2; \quad N2 = 2 * (1 + \beta + \beta^2) * k + \beta^2;$$

Lastfall A:

Horizontale Trapezlast am linken Stiel

$$X_1 = \frac{h^2 * ((8 * eu + 7 * eo) * K1 - (7 * eu + 8 * eo) * k)}{360 * N1}$$

$$X_2 = \frac{h^2 * k * (6 * eu + 9 * eo)}{360 * N1}$$

$$X_3 = \frac{h^2}{12 * N2} * (\beta * (2 * eo + eu) * K2 + \left(\frac{k}{10}\right) * (8 * eu + 7 * eo + \beta * (7 * eu + 8 * eo)))$$

$$\begin{matrix} MB \\ MC \end{matrix} > = -X_2 \pm \beta * \left(\frac{h^2 * (2 * eo + eu)}{12} - X_3\right)$$

$$\begin{matrix} MA \\ MD \end{matrix} > = -X_1 \mp X_3$$

$$HA = -\left(\frac{eo + eu}{2} * h - HD\right)$$

$$HD = \frac{h * (2 * eo + eu)}{12} - \frac{X_1 - X_2}{h}$$

$$VA = -VD = -\frac{h^2 * (2 * eo + eu) - 2 * X_3}{6 * l}$$

Lastfall B:

Vertikale Trapezlast am linken Stiel; X1 bis X3 wie Lastfall A, jedoch a an Stelle von h;

MB und MC wie Lastfall A, jedoch a an Stelle von h;

MA und MD wie Lastfall A.

Lastordinaten wie Lastfall A, jedoch el an Stelle von eu und er an Stelle von eo.

$$HD = HA = \frac{a^2 * (2 * er + el)}{12 * h} - \frac{X_1 - X_2}{h}$$

$$VD = \frac{a^2 * (2 * er + el) - 2 * X_3}{6 * l}$$

$$VA = \frac{(el + er)}{2} * a - VD$$

Lastfall C und D:

Die Schnittkräfte werden durch Spiegelung der Fälle A und B in Schritt 246 erzeugt.

Lastfall E:

Lotrechte Trapezlast auf dem Riegel

$$X_1 = \frac{b^2 * (el + er)}{24 * N1} = \left(\frac{el + er}{2} * b\right) * \frac{b}{2} * \frac{1}{6 * N1} = \frac{(L + R)}{6 * N1}$$

$$X_3 = \frac{\alpha * 10 * (L - R) * K2 + \beta * (L - R)}{2 * N2}$$

$$\text{mit } (L - R) = \frac{b^2 * (el - er)}{60}$$

$$\begin{matrix} MA \\ MD \end{matrix} > = +X_1 \mp X_3$$

$$\begin{matrix} MB \\ MC \end{matrix} > = -2 * X_1 \pm \left[\frac{\alpha * 10 * (L - R)}{2} - \beta * X_3\right]$$

$$HA = HD = \frac{S * a}{2 * h} + \frac{3 * X_1}{h}$$

$$\text{mit } S = \frac{el + er}{2} * a$$

$$\begin{matrix} VA \\ VD \end{matrix} > = \frac{S}{2} \pm \left[\frac{10 * (L - R)}{2 * l} + \frac{2 * X_3}{l}\right]$$

Lastfall F:

Einzellast auf Riegel

$$X_1 = \frac{(L + R)}{6 * N1} = \frac{P * c * d}{b} * 3 * \frac{1}{6 * N1}$$

$$X_3 = \frac{\alpha * P * (d - c) * K2 + \beta * (L - R)}{2 * N2}$$

$$\text{mit } (L - R) = (1 - 2 * c / b) * P * c * d / b$$

MA und MD wie Lastfall E:

$$\begin{matrix} MB \\ MC \end{matrix} > = -2 * X_1 \pm \left[\frac{\alpha * P * (d - c)}{2} - \beta * X_3\right]$$

HA = HD wie Lastfall E, mit S = P

$$\begin{matrix} VA \\ VD \end{matrix} > = \frac{P}{2} \pm \left[\frac{P * (d - c)}{2 * l} + \frac{2 * X_3}{l}\right]$$

Lastfall G:

H von links auf Ecke B

$$MA = -MD = -\frac{H \cdot h \cdot B \cdot K2}{2 \cdot N2}$$

$$MB = -MC = \frac{H \cdot h \cdot B \cdot k \cdot (2+B)}{2 \cdot N2}$$

$$HA = -HD = -\frac{H}{2}$$

$$VA = -VD = -2 \cdot \frac{MB}{b}$$

Lastfall H:

Temperaturänderung

E=Elastizitätsmodul, z.B. für

Beton 34 000 MN/M2

e=Ausdehnungszahl,

hier 0,000 01 für Beton und Stahl

t=Änderung in Grad Celsius. Plus

bei Zunahme

$$MA = MD = +\frac{3 \cdot E \cdot JR \cdot e \cdot t \cdot l}{b \cdot h \cdot k \cdot N1} \cdot (k+1)$$

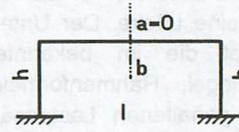
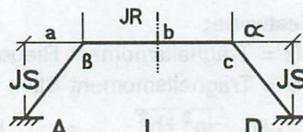
$$MB = MC = -\frac{3 \cdot E \cdot JR \cdot e \cdot t \cdot l}{b \cdot h \cdot k \cdot N1} \cdot k$$

$$HA = HD = \frac{MA - MB}{h} \quad VA = VD = 0$$

- 01 #LBL "EGR"
- 02 SF 21
- 03 CF 12
- 04 CF 13
- 05 ADV
- 06 ADV
- 07 ADV
- 08 "EINGESP. RAHM"
- 09 "EN, TRAPEZ-OD"
- 10 ACA
- 11 "ER RECHTECK, "
- 12 " SYMMETRISCH"
- 13 PRA
- 14 ADV
- 15 SF 27
- 16 CF 28
- 17 CF 29
- 18 FIX 3
- 19 CLRG
- 20 "JR="
- 21 XEQ 11
- 22 STO 01
- 23 " JS="
- 24 XEQ 11
- 25 STO 02
- 26 PRBUF
- 27 SF 13
- 28 "H="
- 29 XEQ 11
- 30 STO 03
- 31 " L="
- 32 XEQ 11
- 33 PRBUF
- 34 STO 04
- 35 "A="

00	01	02	03	04	05	06	07	08	09
k	JR≠J2	JS≠J1	h	l	a	$\alpha = \frac{a}{l}$	$\beta = \frac{b}{l}$	K1	K2
10	11	12	13	14	15	16	17	18	19
N1	ΣMA	ΣMB	ΣMC	ΣMD	ΣHA	ΣHD	ΣVA	ΣVD	b
20	21	22	23	24	25	26	27	28	29
N2	MA	MB	MC	MD	HA	HD	VA	VD	P (im Fall F) S=W
e _{unten} ³⁰	e _{oben} ³¹	H ³²							
e _{links} ³⁰	e _{rechts} ³¹	Zähler ³²	X ₁ ³³	X ₂ ³⁴	X ₃ ³⁵	$\pi \tau_{u,c}$ ³⁶	$\pi \tau_{re,c}$ ³⁷	$\xi = \frac{X}{l}$ ³⁸	Feldmom. ³⁹ in ξ-Station
(L+R) ⁴⁰	(L-R) ⁴¹	[...] ⁴²							
8el+7er	7el+8er	2er+el							

Tabelle: Speicheraufteilung



- 36 XEQ 11
- 37 STO 05
- 38 ST+ X
- 39 RCL 04
- 40 X<>Y
- 41 -
- 42 " B="
- 43 STO 19
- 44 XEQ 12
- 45 RCL 05
- 46 RCL 04
- 47 /
- 48 STO 06
- 49 RCL 19
- 50 RCL 04
- 51 /
- 52 STO 07
- 53 RCL 05
- 54 RCL 03
- 55 R-P
- 56 RCL 19
- 57 /
- 58 RCL 01
- 59 *
- 60 RCL 02
- 61 /
- 62 "K="
- 63 STO 00
- 64 XEQ 12
- 65 CF 13
- 66 ST+ X
- 67 3
- 68 +
- 69 "K1="
- 70 STO 08
- 71 XEQ 10

- 72 1
- 73 RCL 07
- 74 +
- 75 RCL 00
- 76 *
- 77 1
- 78 RCL 00
- 79 +
- 80 RCL 07
- 81 *
- 82 +
- 83 " K2="
- 84 STO 09
- 85 XEQ 12
- 86 RCL 00
- 87 2
- 88 +
- 89 "N1="
- 90 STO 10
- 91 XEQ 10
- 92 RCL 07
- 93 X^2
- 94 RCL 07
- 95 +
- 96 1
- 97 +
- 98 ST+ X
- 99 RCL 00
- 100 *
- 101 RCL 07
- 102 X^2
- 103 +
- 104 " N2="
- 105 STO 20
- 106 XEQ 12
- 107 "SU"

"EGR"

Eingespannter Einfeldrahmen, symmetrisch, Trapez- oder Rechteckform

Rechner HP41CX mit Drucker. MEMORY LOST. SIZE 043. Einlesen vom Laufwerk mit a EGRH α; XEQ α READP α; R/S Literatur Kleinlogel. 5 Dialogfragen zum System, je R/S. Nun stehen die 8 Einzellastfälle (A) bis (H), wie Beispiel, im USER-Modus auf Tastendruck zur Verfügung. Bei lotrechten Stielen ist a=0 zu setzen. Mit 33,035 PRREGX können je Lastfall die Werte X1,X2,X3 ausgedruckt werden. Alle Durchgänge werden selbstständig aufsummiert. Der Stand der Summierung wird mit XEQ 40 gedruckt, mit XEQ 50 wird die Summe im Rechner gelöscht. Für Feldmomente unter Trapezlast ist im Programm "Zweigenkranken" LBL-15 vorgesehen.

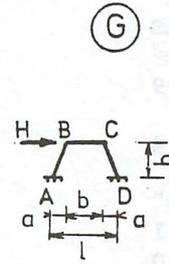
"EGR"

EINGESP. RAHMEN, TRAPEZ-ODER RECHTECK, SYMMETRISCH
KAINDL 1013 <KLEINLOGEL>

JR= 2,000 JS= 1,000
h= 12,000 l= 10,000
a= 2,000 b= 6,000
κ= 4,055
K1= 11,110 K2= 9,521
N1= 6,055 N2= 16,256

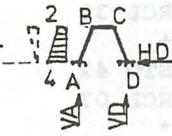
H VON LINKS AUF ECKE B

FALL <G>
H= 2,000
MA= -4,217
MB= 4,670
MC= -4,670
MD= 4,217
HA= -1,000
HD= 1,000
VA= -1,557
VD= 1,557



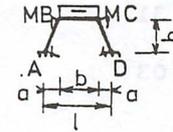
HOR. TRAPEZL. AM LI STIEL

FALL <A>
eu= 4,000 eo= 2,000
MA= -77,384
MB= 13,104
MC= -35,606
MD= 33,435
HA= -28,894
HD= 7,106
VA= -8,118
VD= 8,118



TEMPERERHÜHUNG UM 30GRAD C

FALL <H>
E-MOD= 34000 MN/M2
MA= 1,750
MB= -1,404
MC= -1,404
MD= 1,750
HA= 0,263
HD= 0,263
VA= 0,000
VD= 0,000



VERT. TRAPEZL. AM LI STIEL

FALL
el= 4,000 er= 2,000
MA= -2,150
MB= 0,364
MC= -0,989
MD= 0,929
HA= 0,197
HD= 0,197
VA= 5,774
VD= 0,226



SUMME FALL +A+B+E+F+G+H

XEQ 40
MA= -81,267
MB= 14,807
MC= -47,909
MD= 43,541
HA= -27,024
HD= 10,976
VA= 8,867
VD= 20,133



LOTR. TRAPEZL AUF RIEGEL

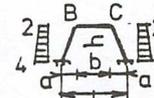
FALL <E>
el= 5,000 er= 2,000
MA= 0,647
MB= -2,321
MC= -4,616
MD= 2,822
HA= 2,184
HD= 2,184
VA= 11,617
VD= 9,383



SUM PURGED XEQ 50

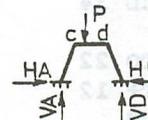
HOR. TRAPEZL. BEIDE STIELE

FALL <C>
eu= 4,000 eo= 2,000
MA= -43,949
MB= -22,502
MC= -22,502
MD= -43,949
HA= -21,797
HD= -21,797
VA= 0,000
VD= 0,000



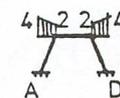
EINZELLAST AUF RIEGEL

FALL <F>
P= 2,000
c= 2,400 d= 3,600
MA= 0,087
MB= -0,326
MC= -0,625
MD= 0,389
HA= 0,226
HD= 0,226
VA= 1,150
VD= 0,350



LOTR. TRAPEZL LI+RE STIEL

FALL <D>
EL= 4,000 ER= 2,000
MA= -1,221
MB= -0,625
MC= -0,625
MD= -1,221
HA= 0,395
HD= 0,395
VA= 6,000
VD= 6,000



108 6	170 X^2	232 STO 23
109 SF 25	171 *	233 ST+ 13
110 PURFL	172 360	234 RCL 33
111 CRFLAS	173 /	235 RCL 35
112 ADV	174 RCL 10	236 +
113 XEQ 09	175 /	237 CHS
114 RTN	176 STO 34	238 STO 21
115#LBL 14	177 RCL 41	239 ST+ 11
116 SF 13	178 RCL 07	240 RCL 35
117 " EU="	179 *	241 RCL 33
118 XEQ 11	180 RCL 40	242 -
119 STO 30	181 +	243 STO 24
120 " EO="	182 RCL 00	244 ST+ 14
121 XEQ 11	183 *	245 RTN
122 STO 31	184 10	246#LBL 18
123 PRBUF	185 /	247 RCL 22
124#LBL 15	186 RCL 31	248 ST+ 13
125 CF 13	187 ST+ X	249 RCL 23
126 RCL 30	188 RCL 30	250 ST+ 12
127 RCL 31	189 +	251 RCL 22
128 +	190 STO 42	252 RCL 23
129 RCL 03	191 RCL 07	253 ST+ 22
130 *	192 *	254 RDN
131 2	193 RCL 09	255 ST+ 23
132 /	194 *	256 RCL 21
133 STO 29	195 +	257 ST+ 14
134 RCL 30	196 RCL 03	258 RCL 24
135 8	197 X^2	259 ST+ 11
136 *	198 *	260 RCL 21
137 RCL 31	199 12	261 RCL 24
138 7	200 /	262 ST+ 21
139 *	201 RCL 20	263 RDN
140 +	202 /	264 ST+ 24
141 STO 40	203 STO 35	265 RCL 26
142 RCL 08	204 RTN	266 ST+ 15
143 *	205#LBL 13	267 RCL 25
144 RCL 30	206 "-----"	268 ST+ 16
145 7	207 SF 12	269 RCL 25
146 *	208 PRA	270 RCL 26
147 RCL 31	209 CF 12	271 ST+ 25
148 8	210 RTN	272 RDN
149 *	211#LBL 16	273 ST+ 26
150 +	212 RCL 42	274 RCL 28
151 STO 41	213 RCL 03	275 ST+ 17
152 RCL 00	214 X^2	276 RCL 27
153 *	215 *	277 ST+ 18
154 -	216 12	278 RCL 27
155 RCL 03	217 /	279 RCL 28
156 X^2	218 RCL 35	280 ST+ 27
157 *	219 -	281 RDN
158 360	220 RCL 07	282 ST+ 28
159 /	221 *	283 XEQ 44
160 RCL 10	222 ENTER^	284 RTN
161 /	223 ENTER^	285#LBL A
162 STO 33	224 RCL 34	286 "+A"
163 RCL 41	225 -	287 APPREC
164 ST+ X	226 STO 22	288 XEQ 13
165 RCL 40	227 ST+ 12	289 "HOR.TRAPEZL."
166 -	228 RDN	290 " -AM LI.STIEL "
167 RCL 00	229 RCL 34	291 PRA
168 *	230 +	292 " FALL <A>"
169 RCL 03	231 CHS	293 AVIEW

294 XEQ 14	356 XEQ 15	418 APPREC
295 XEQ 16	357 XEQ 16	419 XEQ 13
296#LBL 17	358 RCL 42	420 "LOTR.TRAPEZL"
297 RCL 42	359 RCL 03	421 " - LI+RE STIEL"
298 RCL 03	360 X^2	422 ACA
299 *	361 *	423 PRBUF
300 12	362 12	424 " FALL <D>"
301 /	363 /	425 AVIEW
302 RCL 33	364 RCL 05	426 SF 01
303 RCL 34	365 /	427 XEQ 19
304 -	366 RCL 33	428 XEQ 18
305 RCL 03	367 RCL 34	429 RTN
306 /	368 -	430#LBL E
307 -	369 RCL 05	431 "+E"
308 STO 26	370 /	432 APPREC
309 ST+ 16	371 -	433 XEQ 13
310 CHS	372 STO 25	434 "LOTR.TRAPEZL"
311 RCL 29	373 ST+ 15	435 " - AUF RIEGEL"
312 +	374 STO 26	436 ACA
313 CHS	375 ST+ 16	437 PRBUF
314 STO 25	376 RCL 42	438 " FALL <E>"
315 ST+ 15	377 6	439 AVIEW
316 RCL 42	378 /	440 SF 13
317 6	379 RCL 03	441 " EL="
318 /	380 X^2	442 XEQ 11
319 RCL 03	381 *	443 STO 30
320 X^2	382 RCL 35	444 " ER="
321 *	383 ST+ X	445 XEQ 11
322 RCL 35	384 -	446 STO 31
323 ST+ X	385 RCL 04	447 PRBUF
324 -	386 /	448 CF 13
325 RCL 04	387 STO 28	449 RCL 30
326 /	388 ST+ 18	450 RCL 31
327 STO 28	389 CHS	451 +
328 ST+ 18	390 RCL 29	452 2
329 CHS	391 +	453 /
330 STO 27	392 STO 27	454 RCL 19
331 ST+ 17	393 ST+ 17	455 *
332 FC?C 01	394 RCL 03	456 STO 29
333 XEQ 44	395 X<> 05	457 RCL 19
334 RTN	396 STO 03	458 2
335#LBL B	397 FC?C 01	459 /
336 "+B"	398 XEQ 44	460 *
337 APPREC	399 RTN	461 STO 40
338 XEQ 13	400#LBL C	462 RCL 30
339 "VERT.TRAPEZL."	401 "+C"	463 RCL 31
340 " -AM LI.STIEL "	402 APPREC	464 -
341 PRA	403 XEQ 13	465 RCL 19
342 " FALL "	404 "HOR.TRAPEZL."	466 X^2
343 AVIEW	405 " -BEIDE STIELE"	467 *
344 SF 13	406 ACA	468 60
345#LBL 19	407 PRBUF	469 /
346 " EL="	408 " FALL <C>"	470 STO 41
347 XEQ 11	409 AVIEW	471 10
348 STO 30	410 XEQ 14	472 *
349 " ER="	411 XEQ 16	473 STO 37
350 XEQ 11	412 SF 01	474 XEQ 20
351 STO 31	413 XEQ 17	475 RTN
352 PRBUF	414 XEQ 18	476#LBL F
353 RCL 03	415 RTN	477 "+F"
354 X<> 05	416#LBL D	478 APPREC
355 STO 03	417 "+D"	479 XEQ 13

480 "EINZELLAST A"	542 STO 35	604 ST+ X
481 " -UF RIEGEL"	543 RCL 40	605 RCL 04
482 ACA	544 RCL 10	606 /
483 PRBUF	545 /	607 +
484 " FALL <F>"	546 6	608 STO 42
485 AVIEW	547 /	609 RCL 29
486 "P="	548 STO 33	610 2
487 XEQ 11	549 RCL 06	611 /
488 PRBUF	550 RCL 37	612 STO 41
489 STO 29	551 *	613 +
490 SF 13	552 2	614 STO 27
491 "C="	553 /	615 ST+ 17
492 XEQ 11	554 RCL 07	616 RCL 41
493 STO 36	555 RCL 35	617 RCL 42
494 RCL 19	556 *	618 -
495 X<>Y	557 -	619 STO 28
496 -	558 STO 34	620 ST+ 18
497 " D="	559 RCL 33	621 XEQ 44
498 STO 38	560 ST+ X	622 RTN
499 XEQ 12	561 CHS	623#LBL G
500 CF 13	562 STO 42	624 "+G"
501 RCL 29	563 +	625 APPREC
502 *	564 STO 22	626 XEQ 13
503 RCL 36	565 ST+ 12	627 "H VON LINKS"
504 *	566 RCL 42	628 " - AUF ECKE B"
505 RCL 19	567 RCL 34	629 ACA
506 /	568 -	630 PRBUF
507 STO 30	569 STO 23	631 " FALL <G>"
508 3	570 ST+ 13	632 AVIEW
509 *	571 RCL 33	633 "H="
510 STO 40	572 RCL 35	634 XEQ 11
511 1	573 +	635 PRBUF
512 RCL 36	574 STO 24	636 STO 32
513 ST+ X	575 ST+ 14	637 RCL 03
514 RCL 19	576 RCL 33	638 RCL 07
515 /	577 RCL 35	639 *
516 -	578 -	640 *
517 RCL 30	579 STO 21	641 RCL 09
518 *	580 ST+ 11	642 *
519 STO 41	581 RCL 29	643 2
520 RCL 38	582 RCL 05	644 /
521 RCL 36	583 *	645 RCL 20
522 -	584 2	646 /
523 RCL 29	585 /	647 STO 24
524 *	586 RCL 03	648 ST+ 14
525 STO 37	587 /	649 CHS
526 XEQ 20	588 RCL 33	650 STO 21
527 RTN	589 3	651 ST+ 11
528#LBL 20	590 *	652 RCL 32
529 RCL 37	591 RCL 03	653 RCL 03
530 RCL 06	592 /	654 RCL 07
531 *	593 +	655 *
532 RCL 09	594 STO 25	656 *
533 *	595 ST+ 15	657 RCL 00
534 RCL 07	596 STO 26	658 *
535 RCL 41	597 ST+ 16	659 RCL 07
536 *	598 RCL 37	660 2
537 +	599 2	661 +
538 2	600 /	662 *
539 /	601 RCL 04	663 2
540 RCL 20	602 /	664 /
541 /	603 RCL 35	665 RCL 20

666 /	725 RCL 10	784 STO 32
667 STO 22	726 /	785 ENTER^
668 ST+ 12	727 STO 29	786#LBL 45
669 CHS	728 RCL 00	787 RCL IND 32
670 STO 23	729 1	788 "MA="
671 ST+ 13	730 +	789 XEQ 21
672 RCL 32	731 *	790 "MB="
673 2	732 STO 21	791 XEQ 21
674 /	733 ST+ 11	792 "MC="
675 STO 26	734 STO 24	793 XEQ 21
676 ST+ 16	735 ST+ 14	794 "MD="
677 CHS	736 RCL 29	795 XEQ 21
678 STO 25	737 RCL 00	796 "HA="
679 ST+ 15	738 *	797 XEQ 21
680 RCL 22	739 CHS	798 "HD="
681 ST+ X	740 STO 22	799 XEQ 21
682 RCL 19	741 ST+ 12	800 "VA="
683 /	742 STO 23	801 XEQ 21
684 STO 28	743 ST+ 13	802 "VD="
685 ST+ 18	744 RCL 21	803 XEQ 12
686 CHS	745 RCL 22	804 ADV
687 STO 27	746 -	805 GTO 09
688 ST+ 17	747 RCL 03	806#LBL 21
689 XEQ 44	748 /	807 XEQ 12
690 RTN	749 STO 25	808 ISG 32
691#LBL H	750 ST+ 15	809 RCL IND 32
692 "+H"	751 STO 26	810 RTN
693 APPREC	752 ST+ 16	811#LBL 50
694 XEQ 13	753 0	812 XEQ 13
695 FIX 0	754 STO 27	813 "SUM PURGED"
696 "TEMPERHOHUNG"	755 STO 28	814 AVIEW
697 "└ UM"	756 XEQ 44	815 11.018
698 XEQ 11	757 RTN	816 CLRGX
699 "GRAD C"	758#LBL 40	817 "SU"
700 ACA	759 XEQ 13	818 CLFL
701 PRBUF	760 "SUMME FALL "	819#LBL 09
702 STO 32	761 ACA	820 "FALL ?"
703 " FALL <H>"	762 CLX	821 CF 21
704 AVIEW	763 "SU"	822 AVIEW
705 " E-MOD="	764 SEEKPTA	823 SF 21
706 XEQ 11	765#LBL 41	824 CLA
707 PRBUF	766 SF 25	825 RTN
708 FIX 3	767 GETREC	826#LBL 11
709 3	768 FC? 25	827 TONE 5
710 *	769 GTO 42	828#LBL 00
711 RCL 01	770 ACA	829 CF 22
712 RCL 32	771 GTO 41	830 PROMPT
713 *	772#LBL 42	831 FC? 22
714 *	773 PRBUF	832 GTO 00
715 RCL 04	774 11.1	833#LBL 10
716 *	775 STO 32	834 ACA
717 1 E5	776 XEQ 45	835 ACX
718 /	777 CLA	836 RTN
719 RCL 19	778 "XEQ50?"	837#LBL 12
720 /	779 ASTO X	838 XEQ 10
721 RCL 03	780 ADV	839 PRBUF
722 /	781 RTN	840 END
723 RCL 00	782#LBL 44	
724 /	783 21.1	

Ludwig Kaindl, Hermann-Lönsstr. 26
W8012 Ottobrunn

EGR

Zeile 1 von EGR (1-6) CCD-Barcodes Ludwig Kaindl



Zeile 2 von EGR (7-10) CCD-Barcodes Ludwig Kaindl



Zeile 3 von EGR (11-15) CCD-Barcodes Ludwig Kaindl



Zeile 4 von EGR (16-17) CCD-Barcodes Ludwig Kaindl



Zeile 5 von EGR (18-18) CCD-Barcodes Ludwig Kaindl



Zeile 6 von EGR (19-21) CCD-Barcodes Ludwig Kaindl



Zeile 7 von EGR (22-27) CCD-Barcodes Ludwig Kaindl



Zeile 8 von EGR (28-31) CCD-Barcodes Ludwig Kaindl



Zeile 9 von EGR (32-37) CCD-Barcodes Ludwig Kaindl



Zeile 10 von EGR (38-41) CCD-Barcodes Ludwig Kaindl



Zeile 11 von EGR (42-48) CCD-Barcodes Ludwig Kaindl



Zeile 12 von EGR (49-50) CCD-Barcodes Ludwig Kaindl



Zeile 13 von EGR (51-60) CCD-Barcodes Ludwig Kaindl



Zeile 14 von EGR (61-67) CCD-Barcodes Ludwig Kaindl



Zeile 15 von EGR (68-75) CCD-Barcodes Ludwig Kaindl



Zeile 16 von EGR (76-84) CCD-Barcodes Ludwig Kaindl



Zeile 17 von EGR (85-86) CCD-Barcodes Ludwig Kaindl



Zeile 18 von EGR (87-95) CCD-Barcodes Ludwig Kaindl



Zeile 19 von EGR (96-106) CCD-Barcodes Ludwig Kaindl



Zeile 20 von EGR (107-114) CCD-Barcodes Ludwig Kaindl



Zeile 21 von EGR (115-118) CCD-Barcodes Ludwig Kaindl



Zeile 22 von EGR (119-125) CCD-Barcodes Ludwig Kaindl



Zeile 23 von EGR (126-127) CCD-Barcodes Ludwig Kaindl



Zeile 24 von EGR (128-137) CCD-Barcodes Ludwig Kaindl



Zeile 25 von EGR (138-147) CCD-Barcodes Ludwig Kaindl



Zeile 26 von EGR (148-158) CCD-Barcodes Ludwig Kaindl



Zeile 27 von EGR (159-165) CCD-Barcodes Ludwig Kaindl



Zeile 28 von EGR (166-175) CCD-Barcodes Ludwig Kaindl



Zeile 29 von EGR (176-187) CCD-Barcodes Ludwig Kaindl



Zeile 30 von EGR (188-191) CCD-Barcodes Ludwig Kaindl



Zeile 31 von EGR (192-200) CCD-Barcodes Ludwig Kaindl



Zeile 32 von EGR (201-203) CCD-Barcodes Ludwig Kaindl



Zeile 33 von EGR (204-209) CCD-Barcodes Ludwig Kaindl



Zeile 34 von EGR (210-220) CCD-Barcodes Ludwig Kaindl



Zeile 35 von EGR (221-226) CCD-Barcodes Ludwig Kaindl



Zeile 36 von EGR (227-236) CCD-Barcodes Ludwig Kaindl



Zeile 37 von EGR (237-244) CCD-Barcodes Ludwig Kaindl



Zeile 38 von EGR (245-250) CCD-Barcodes Ludwig Kaindl



Zeile 39 von EGR (251-257) CCD-Barcodes Ludwig Kaindl



Zeile 40 von EGR (258-264) CCD-Barcodes Ludwig Kaindl



Zeile 41 von EGR (265-271) CCD-Barcodes Ludwig Kaindl



Zeile 42 von EGR (272-278) CCD-Barcodes Ludwig Kaindl



Zeile 43 von EGR (279-283) CCD-Barcodes Ludwig Kaindl



Zeile 44 von EGR (284-289) CCD-Barcodes Ludwig Kaindl



Zeile 45 von EGR (290-296) CCD-Barcodes Ludwig Kaindl



Zeile 46 von EGR (297-306) CCD-Barcodes Ludwig Kaindl



Zeile 47 von EGR (307-309) CCD-Barcodes Ludwig Kaindl



Zeile 48 von EGR (310-316) CCD-Barcodes Ludwig Kaindl



Zeile 49 von EGR (317-327) CCD-Barcodes Ludwig Kaindl



Zeile 50 von EGR (328-331) CCD-Barcodes Ludwig Kaindl



Zeile 51 von EGR (332-338) CCD-Barcodes Ludwig Kaindl



Zeile 52 von EGR (339-341) CCD-Barcodes Ludwig Kaindl



Zeile 53 von EGR (342-342) CCD-Barcodes Ludwig Kaindl



Zeile 54 von EGR (343-349) CCD-Barcodes Ludwig Kaindl



Zeile 55 von EGR (350-354) CCD-Barcodes Ludwig Kaindl



Zeile 56 von EGR (355-362) CCD-Barcodes Ludwig Kaindl



Zeile 57 von EGR (363-367) CCD-Barcodes Ludwig Kaindl



Zeile 58 von EGR (368-372) CCD-Barcodes Ludwig Kaindl



Zeile 59 von EGR (373-379) CCD-Barcodes Ludwig Kaindl



Zeile 60 von EGR (380-389) CCD-Barcodes Ludwig Kaindl



Zeile 61 von EGR (390-397) CCD-Barcodes Ludwig Kaindl



Zeile 62 von EGR (398-407) CCD-Barcodes Ludwig Kaindl



Zeile 63 von EGR (408-412) CCD-Barcodes Ludwig Kaindl



Zeile 64 von EGR (413-416) CCD-Barcodes Ludwig Kaindl



Zeile 65 von EGR (417-420) CCD-Barcodes Ludwig Kaindl



Zeile 66 von EGR (421-429) CCD-Barcodes Ludwig Kaindl



Zeile 67 von EGR (430-437) CCD-Barcodes Ludwig Kaindl



Zeile 68 von EGR (438-440) CCD-Barcodes Ludwig Kaindl



Zeile 69 von EGR (441-445) CCD-Barcodes Ludwig Kaindl



Zeile 70 von EGR (446-446) CCD-Barcodes Ludwig Kaindl



Zeile 71 von EGR (447-452) CCD-Barcodes Ludwig Kaindl



Zeile 72 von EGR (453-454) CCD-Barcodes Ludwig Kaindl



Zeile 73 von EGR (455-455) CCD-Barcodes Ludwig Kaindl



Zeile 74 von EGR (456-459) CCD-Barcodes Ludwig Kaindl



Zeile 75 von EGR (460-460) CCD-Barcodes Ludwig Kaindl



Zeile 76 von EGR (461-465) CCD-Barcodes Ludwig Kaindl



Zeile 77 von EGR (466-470) CCD-Barcodes Ludwig Kaindl



Zeile 78 von EGR (471-476) CCD-Barcodes Ludwig Kaindl



Zeile 79 von EGR (477-479) CCD-Barcodes Ludwig Kaindl



Zeile 80 von EGR (480-482) CCD-Barcodes Ludwig Kaindl



Zeile 81 von EGR (483-490) CCD-Barcodes Ludwig Kaindl



Zeile 82 von EGR (491-501) CCD-Barcodes Ludwig Kaindl



Zeile 83 von EGR (502-512) CCD-Barcodes Ludwig Kaindl



Zeile 84 von EGR (513-516) CCD-Barcodes Ludwig Kaindl



Zeile 85 von EGR (517-520) CCD-Barcodes Ludwig Kaindl



Zeile 86 von EGR (521-532) CCD-Barcodes Ludwig Kaindl



Zeile 87 von EGR (533-538) CCD-Barcodes Ludwig Kaindl



Zeile 88 von EGR (539-544) CCD-Barcodes Ludwig Kaindl



Zeile 89 von EGR (545-554) CCD-Barcodes Ludwig Kaindl



Zeile 90 von EGR (555-562) CCD-Barcodes Ludwig Kaindl



Zeile 91 von EGR (563-572) CCD-Barcodes Ludwig Kaindl



Zeile 92 von EGR (573-583) CCD-Barcodes Ludwig Kaindl



Zeile 93 von EGR (584-591) CCD-Barcodes Ludwig Kaindl



Zeile 94 von EGR (592-600) CCD-Barcodes Ludwig Kaindl



Zeile 95 von EGR (601-610) CCD-Barcodes Ludwig Kaindl



Zeile 96 von EGR (611-620) CCD-Barcodes Ludwig Kaindl



Zeile 97 von EGR (621-629) CCD-Barcodes Ludwig Kaindl



Zeile 98 von EGR (630-637) CCD-Barcodes Ludwig Kaindl



Zeile 99 von EGR (638-644) CCD-Barcodes Ludwig Kaindl



Zeile 100 von EGR (645-654) CCD-Barcodes Ludwig Kaindl



Zeile 101 von EGR (655-663) CCD-Barcodes Ludwig Kaindl



Zeile 102 von EGR (664-673) CCD-Barcodes Ludwig Kaindl



Zeile 103 von EGR (674-682) CCD-Barcodes Ludwig Kaindl



Zeile 104 von EGR (683-689) CCD-Barcodes Ludwig Kaindl



Zeile 105 von EGR (690-694) CCD-Barcodes Ludwig Kaindl



Zeile 106 von EGR (695-696) CCD-Barcodes Ludwig Kaindl



Zeile 107 von EGR (697-702) CCD-Barcodes Ludwig Kaindl



Zeile 108 von EGR (703-710) CCD-Barcodes Ludwig Kaindl



Zeile 109 von EGR (711-716) CCD-Barcodes Ludwig Kaindl



Zeile 110 von EGR (717-728) CCD-Barcodes Ludwig Kaindl



Zeile 111 von EGR (729-737) CCD-Barcodes Ludwig Kaindl



Zeile 112 von EGR (738-750) CCD-Barcodes Ludwig Kaindl



Zeile 113 von EGR (751-759) CCD-Barcodes Ludwig Kaindl



Zeile 114 von EGR (760-765) CCD-Barcodes Ludwig Kaindl



Zeile 115 von EGR (766-770) CCD-Barcodes Ludwig Kaindl



Zeile 116 von EGR (771-776) CCD-Barcodes Ludwig Kaindl



Zeile 117 von EGR (777-777) CCD-Barcodes Ludwig Kaindl



Zeile 118 von EGR (778-780) CCD-Barcodes Ludwig Kaindl



Zeile 119 von EGR (781-784) CCD-Barcodes Ludwig Kaindl



Zeile 120 von EGR (785-789) CCD-Barcodes Ludwig Kaindl



Zeile 121 von EGR (790-798) CCD-Barcodes Ludwig Kaindl



Zeile 122 von EGR (799-807) CCD-Barcodes Ludwig Kaindl



Zeile 123 von EGR (808-819) CCD-Barcodes Ludwig Kaindl



Zeile 124 von EGR (820-825) CCD-Barcodes Ludwig Kaindl



Zeile 125 von EGR (826-829) CCD-Barcodes Ludwig Kaindl



Zeile 126 von EGR (830-835) CCD-Barcodes Ludwig Kaindl



Zeile 127 von EGR (836-841) CCD-Barcodes Ludwig Kaindl



Zeile 128 von EGR (842-844) CCD-Barcodes Ludwig Kaindl



Zeile 129 von EGR (845-853) CCD-Barcodes Ludwig Kaindl



Zeile 130 von EGR (854-859) CCD-Barcodes Ludwig Kaindl



Zeile 131 von EGR (860-867) CCD-Barcodes Ludwig Kaindl



Zeile 132 von EGR (868-872) CCD-Barcodes Ludwig Kaindl



Zeile 133 von EGR (873-879) CCD-Barcodes Ludwig Kaindl



Zeile 134 von EGR (880-887) CCD-Barcodes Ludwig Kaindl



Zeile 135 von EGR (888-894) CCD-Barcodes Ludwig Kaindl



Zeile 136 von EGR (895-903) CCD-Barcodes Ludwig Kaindl



Zeile 137 von EGR (904-913) CCD-Barcodes Ludwig Kaindl



Zeile 138 von EGR (914-919) CCD-Barcodes Ludwig Kaindl



Zeile 139 von EGR (920-923) CCD-Barcodes Ludwig Kaindl



Zeile 140 von EGR (924-926) CCD-Barcodes Ludwig Kaindl



Zeile 141 von EGR (927-930) CCD-Barcodes Ludwig Kaindl



Zeile 142 von EGR (931-933) CCD-Barcodes Ludwig Kaindl



RS232-HPIL Interface

Testbericht für den HP48SX

von Martin Meyer

Da steht es nun, ein kleines Kästchen in der Größe des HP41C, aber halt, es war doch für den HP48 gedacht...

Eine Probe auf's Exempel zeigt, daß der HP48 tatsächlich darüber passt, wie ja angepriesen worden war, auch wenn er ein wenig über das Gehäuse des Interfaces hinausragt.

Das Gerät macht einen soliden Eindruck und liegt irgendwie gut in der Hand, was für einen mobilen Einsatz schon in's Gewicht fallen würde.

Das mir vorliegende Interface ist für den stationären Einsatz gedacht und somit über ein Steckernetzteil mit dem lebenswichtigen "Saft" versorgt.

Auf der Rückseite des Gehäuses befinden sich die Anschlüsse für die RS232-Schnittstelle (9-polig), eine Klinkenbuchse für die Stromversorgung, wie wir sie von den Kopfhörern unseres Walkmans her kennen. Ebenso befindet sich hier ein kleiner roter Knopf, der die 9V-Versorgung ein- bzw. ausschaltet. Das Steckernetzteil muß man aber trotzdem von Hand aus der Steckdose ziehen.

Eine Version des Interfaces für Accu-Betrieb ist ebenso erhältlich, dazu muß nur das Gehäuseoberteil ausgetauscht werden. Damit ist also ein portabler Betrieb mit den anderen netzunabhängigen Geräten möglich, wofür die IL-Schleife ja konzipiert worden war.

Lange Rede, kurzer Sinn, wenden wir uns nun der Bedienung zu und beginnen mit den

Voraussetzungen

1. Ein funktionsfähiger HP48SX mit mindestens 32kByte freiem !! RAM. D.h. entweder ist der Rechner völlig leer oder es wird

eine Speichererweiterung benötigt.

Der Hintergrund ist die Befehlsbibliothek, die 20kByte groß ist, in Worten zwanzigtausend...

2. Das RS232 Interface-Kit für IBM-kompatible oder Apple Macintosh. Benötigt wird in erster Linie das serielle Kabel, mit dem der HP48SX mit dem RS232-HPIL Interface verbunden werden soll.

3. Das Datenübertragungsprogramm KERMIT, um die Befehlslibrary in den HP48 übertragen zu können. Das eben erwähnte Programm wird bei den entsprechenden Interface-Kits von HP mitgeliefert. Für Besitzer von ATARI-ST bzw. TT gibt es schon seit Urzeiten GEM-Kermit, daß ich benutzt habe, um die Library von der Diskette auf den HP48 zu übertragen, es klappte tadellos.

Leider waren im Handbuch keinerlei Angaben darüber gemacht worden, wie das SETUP des I/O-Menüs auszusehen hat, bei mir war es natürlich prompt falsch eingestellt, Murphy läßt grüßen:

```
IR/wire:          wire
ASCII/BINARY:    ASCII
baud:            9600
parity:          none 0
checksum type:   3
translation code: 1
```

Installation

In einer auf der Diskette befindlichen Textdatei sind die Schritte für die Installation der Befehlslibrary kurz beschrieben, es reicht selbst für den Laien aus.

Schwierigkeiten kann es nur bei dem benötigten Speicherplatz geben, 20kByte sind eine stolze Größe, da müßte eigentlich durch RPL oder Assemblerprogrammierung

eine andere Größenordnung zu erreichen sein. Wenn man bedenkt, daß der HP48S keine Steckplätze hat und somit nur über 32kByte maximal verfügen kann, so kann man nur diese eine Library im Rechner geladen haben, für weitere Programme ist dann kein Platz mehr. Daten dürfen also in der HP-IL Schleife garnicht anfallen, sonst läuft der Rechner gleich über.

So, nach der langen Einleitung kommen wir nun zu den Befehlen, die ich der Einfachheit halber in der alphabetischen Reihenfolge des Handbuchs abhandle.

Die Befehle sind alle auf Softkeys erreichbar. Ich habe als Beispielkonfiguration einfach mein Digitalkassettenlaufwerk HP82161A und den Thermodrucker HP82162A als Einheit 1 und 2 an das Interface gehängt, siehe dazu Bild 1 auf der nächsten Seite.

ACCID

Dieser Befehl liest die Gerätenummer eines IL-Gerätes und legt diese im Stack 1: ab.

Das Digitalkassettenlaufwerk hat z.B. die Nummer 16, der Thermodrucker die Nummer 32.

```
2 ENTER ACCID ⇨ 1: 32
```

AUTOIO

wählt den Betriebsmodus AUTO, in dem das Interface automatisch zu druckerbezogenen Befehlen eine Druckerienheit (AACCID 32 - 47) oder für eine Massenspeicheroperation wie DIRECTORY nach einer Massenspeicherienheit (ACCID 16 - 31) sucht.

CLEARDEV

löscht das in Stackebene 1: spezifizierte Gerät. Steht dort eine 0, so werden alle Geräte der Schleife zurückgesetzt.

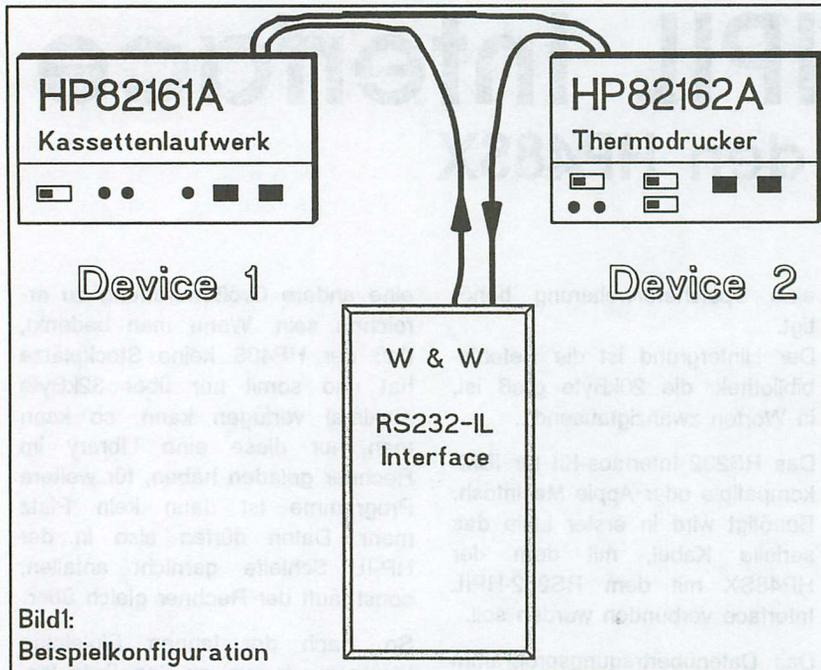


Bild1:
Beispielkonfiguration

DEVID

liest von dem in Stackebene 1: benannten Gerät den Identifikationsstring. Die in meinem Beispiel benutzten Geräte melden keinen String zurück, also nehmen wir einmal an, daß der Thinkjet anstatt des Thermodruckers das 2. Gerät unserer Schleife ist:

```
2 ENTER DEVID ⇨ 1:"HP2225B"
```

DEVSTAT

liest von der primären Einheit das Statusbyte. Hierin kann man z.B. bei IL-Thermodrucker erkennen, ob das Papier ausgegangen ist oder beim Kassettenlaufwerk, daß er noch "busy" ist, d.h. eventuell etwas sucht.

```
1 ENTER DEVSTAT ⇨ 1: #20d
```

Ein Blick in das Handbuch sagt uns, daß keine Kassette im Laufwerk ist...

DIRECTORY

Dies ist wohl der für die meisten interessanteste Befehl:

Er liest ein HP41/71/75-Verzeichnis von einem Massenspeicher, egal, ob Diskette oder Kassette.

Das Ergebnis ist eine Zeichenkette, die aus einer Anzahl Zeilen entsprechend der Anzahl der Verzeichniseinträge besteht. Sind also auf der Kassette 2 Dateien abgespei-

chert, so hat der Textstring auch nur 2 Zeilen (ohne Kopfzeile).

Es wird immer das komplette Verzeichnis gelesen, was unter Umständen bei vielen Verzeichniseinträgen auch mal eine halbe Stunde in Anspruch nimmt (120 Einträge). Der Text war dann über 5kByte groß, ein Anschauen mit EDIT überforderte den HP48SX deutlich, er war wieder für mehrere Minuten am Arbeiten.

Eine Anzeige der einzelnen Einträge (mindestens des Namens und des Typs) während der Übertragung der einzelnen Einträge, wie es beim HP41/71/75 der Fall ist, ist absolut notwendig, damit man bei erfolgreicher Suche nach einem Eintrag den Befehl abbrechen kann und nicht die lange Zeit warten muß, bis das gesamte Verzeichnis gelesen worden ist.

Warum das Lesen des Direktories so lange dauern muß ist mir aus technischer Sicht noch nicht so ganz klar geworden.

Überhaupt sind die Datenübertragungszeiten schlichtweg endlos, wenn man es nur mit dem HP41 mit IL-Modul vergleicht, vom HP71 ganz zu schweigen. Bei einer Bearbeitung größerer Dateien sollte man schon die Kaffeemaschine laufen haben.

Nun aber ein Beispiel: Ich die Va-

riablen 'ILpar' und 'IOPAR' auf einer leeren Kassette abgespeichert.

DIRECTORY ⇨

```
"----- LABEL 23e 08.09.91 16:26
IOPAR LIST 40h 08.09.91 16:28
ILpar LIST 36h 08.09.91 16:30"
```

Die erste Zeile ist die Kopfzeile, die das Datum der Initialisierung angibt, die 23e dürften die Anzahl der maximal verfügbaren Einträge im Verzeichnis sein. Das Handbuch schweigt sich zu diesem Thema aus, ebenso wie bislang jegliches Beispiel sowie alle Hintergrundinformationen zu den verwendeten Befehlen fehlten.

Hier besteht dringend Nachholbedarf!

Die Angabe 40h bedeutet, daß die Variable IOPAR im HP48 einen Platzbedarf von 40 Halbbytes hat, das sind rein mathematisch betrachtet 20 Bytes. Auf dem Massenspeicher belegt man aber trotzdem einen kompletten Sektor.

Die stärkste Seite dieses Befehls ist die Fähigkeit alle bislang verwendeten Formate zu lesen. Damit ist man nach allen Seiten offen.

DIRPART

liest den angegebenen Verzeichniseintrag oder auch von einem angegebenen Verzeichniseintrag an eine bestimmte Anzahl von Verzeichniseinträgen von dem Massenspeicher, das Ergebnis sieht dann aus wie das Beispiel oben bei DIRECTORY.

FINDDEV

Mit diesem Befehl kann man ein bestimmtes Gerät in der Schleife suchen lassen, das Ergebnis ist dessen Position innerhalb der Schleife:

```
16 ENTER FINDDEV ⇨ 1: 1
```

Wäre statt des Thermodruckers der Thinkjet angeschlossen, so könnte ich auch ihn explizit suchen lassen:

```
"HP2225B" ENTER FINDDEV ⇨
1: 2
```

INITIALIZE

Mit diesem Befehl kann man ein Massenspeichermedium formatieren, auch Disketten lassen sich komplett nutzen.

INSTR

Dieser Befehl ist das Pendant zu IND bzw. INA des HP41, mit dem man eine in 1: angegebene Anzahl Zeichen von einem Massenspeicher oder einer anderen beliebigen Datenquelle in der Interface-Loop einlesen kann:

```
12 ENTER INSTR ⇨
1: "Beispieltext"
```

```
"x" ENTER INSTR ⇨
1: "Beispielte"
```

Diesmal habe ich ein "Abbruchzeichen" angegeben, auch dies ist möglich.

LOCALDEV

schaltet das angegebene Gerät in den Local-Modus.

Einige über eine Tastatur bedienbare Geräte, z.B. ein Multimeter, schalten sich in den sogenannten Remote-Modus, sobald sie einen Befehl über ihre Schnittstelle erhalten haben. Ab diesem Zeitpunkt ist die Tastatur dieses Gerätes gesperrt, um eine Bedienung während der Programmierung durch einen externen Controller zu verhindern.

Um dieses Gerät wieder über seine Bedienelemente steuern zu können gibt es in der Regel eine Taste Local oder den eben erwähnten Befehl LOCALDEV.

In meiner Beispielschleife gibt es kein Gerät, das solche Bedienelemente hat. Deshalb hier an dieser Stelle kein Beispiel.

MANIO

stellt den Modus manual IO ein.

Dies bedeutet, daß der Controller nicht mehr automatisch bei Druckbefehlen nach einem Drucker sucht oder bei Massenspeicherbefehlen nach einem Massenspeicher (Device-ID 16-31). Alle Befehle richten sich dann an die mit SELECT gewählte primäre Einheit.

NLOOP

ergibt die Anzahl der in der Schleife angeschlossenen Geräte:

```
NLOOP ⇨ 1: 2
```

OUTSTR

sendet einen String aus Stackebe-

ne 1: im Gegensatz zu PR1IL unverändert an die primäre Einheit.

```
"Beispieltext" ENTER OUTSTR
```

OUTSTRLF

arbeitet genauso wie OUTSTR, nur daß hier an den String aus Stackebene 1: noch der in der Variablen PRTPAR spezifizierte Zeilenvorschub angehängt wird.

PR1IL

druckt den Inhalt eines Objektes aus.

Dabei werden alle Einstellungen beachtet, die auch für PR1L Anwendung finden: Übersetzungstabelle in TRANSIO, Systemflags etc.

PWRDOWN

Dieser Befehl schaltet alle Geräte, die die Eigenschaft besitzen sich selbst abschalten zu können in den Power Down Modus.

Das RS232-IL Interface schaltete sich selbst auch ab, was für den Accu-Betrieb wichtig ist, wenn man die Schleife für eine Weile totlegen will und erst nach ein paar Tagen wieder Meßwerte erfassen will.

Dem Handbuch entnahm ich nun, daß jeder Befehl, z.B. NLOOP meine im Power Down Modus befindlichen Geräte wieder anwerfen würde, ähnlich wie dies beim HP41 mittels des Befehls PWRUP möglich ist.

Die Ausführung dieses oder anderer Befehle brachte leider meine in Schlafstellung harrenden Geräte nicht wieder zu Leben, jeder Versuch endete mit der Fehlermeldung Loop timeout.

Mir blieb also nur das Wiederbeleben von Hand übrig, das Interface schaltete sich wundersamerweise wieder ein.

RCLSEL

leifert die IL-Adresse der primären Einheit in den Stack zurück.

READVAR

liest ein Objekt von einem Massenspeicher direkt in eine Variable mit demselben Namen wie angegeben.

```
'IOPAR' ENTER READVAR ⇨
```

Existiert diese Variable bereits, so wird diese nicht einfach überschrieben sondern es wird eine Variable mit demselben Namen.x angelegt, wobei x eine laufende Nummer von 1-9 ist. Liest man also eine Variable mehrmals ein, so werden immer neue Variablen angelegt.

Es lassen sich nicht nur HP48 Objekte eingelesen, es lassen sich auch HP41/71 Dateien einlesen.

Eine HP71 Textdatei z.B. wird in eine Liste von Zeichenketten umgewandelt, Datendateien in einen eindimensionalen Vektor. Man kann also seine alten Daten weiterverarbeiten, eine sehr gute Lösung.

Auch hier wieder der Wehrmuts-tropfen: Es dauert alles schier endlos lange.

REMOTEDEV

schaltet das in Stackebene 1: angegebene Gerät in seinen Remote Modus. Das z.B. Multimeter läßt sich nun nicht mehr über sein Bedienfeld steuern sondern nur noch mittels Befehlen über die IL-Schleife.

REMOVE

löscht die in 1: angegebene Datei egal welchen Typs (HP41/71/75 etc.), auch das Flag SECURE kann dabei bei Bedarf ignoriert werden.

Eine sehr gefährliche Eigenschaft der beiden eben beschriebenen Befehle ist ihre ähnliche Schreibweise, auf dem Softkey erscheinen nämlich nur die ersten 4 Buchstaben:

REMO

Man muß also schon genau wissen, was man gerade vorhat und welche Taste dafür zuständig ist.

RENAME

benennt eine Datei auf einem Massenspeichermedium um:

```
{'Alter Name' 'Neuer Name'}
ENTER RENAME
```

Auch hier läßt sich der Status SECURE ignorieren.

SECURE

sichert die in Stackebene 1: angegebene Datei auf einem Massenspeichermedium gegen unbeabsichtigtes Löschen.

SELECT

setzt die primäre Einheit neu. Dies ist das Gerät, das im Modus MA-NIO als einziges von z.B. Druckbefehlen angesprochen wird, ohne daß das Interface kontrolliert, ob dies wirklich ein Druckgerät ist.

Man möchte ja eventuell auch auf z.B. das IL-GPIO Interface ausdrucken, um einen Drucker mit paralleler Schnittstelle anzusteuern.

TRIGGERDEV

triggert z.B. das Multimeter jetzt eine Messung vorzunehmen. Ansonsten reagieren z.B. meine beiden Beispielgeräte nicht auf dieses Kommando.

UNSECURE

hebt den Schutz gegen Löschen einer Datei wieder auf, eigentlich logisch...

WRTVAR

schreibt die angegebene Variable auf ein Massenspeichermedium. Die Datei kann dabei gleich gesichert werden, was sich bei der langsamen Abarbeitung des Befehls auch dringend empfiehlt.

Die hiermit angelegte Datei kann nur mittels des Befehls READVAR wieder gelesen werden.

ILROMVER

erzeugt eine Copyrightmeldung mit einer IL-Grafik und zeigt die Version der verwendeten Software an, in meinem Fall die Version A.

RS232VER

fragt das Interface nach seiner Versionsnummer der internen Software:

RS232VER ⇨

1: "IL-CONVERTER: V1.0"

So, das waren alle Befehle. Das Sortiment ist im Grund genommen relativ komplett, vermisst habe ich nur die Möglichkeit das Interface aus seinem Modus Controller heraus zu holen, damit man z.B. vom HP41 aus so tun kann, als ob das Interface ein Zieldevice statt eines Druckers oder ähnliches ist. Damit ließ sich für viele die Anschaffung des Infrarotmoduls für eine direkte Datenübertragung ersparen.

Das mir vorliegende Handbuch kann auch nicht der Weisheit letzter Schluß sein, weder in Form noch Inhalt. Ich habe vor allem Details zu den einzelnen Kommandos vermisst, was schicken denn diese zum Interface bzw. was kann denn das Interface überhaupt, wenn man die

mitgelieferte Library nicht verwenden will ?

Ein Abspecken der Library wäre dringend nötig, man kann ja nicht ständig dieselbe von einem Rechner herunterladen. Mit 20kByte ist sie für einen permanenten Verbleib einfach zu groß. Eine Möglichkeit des Herausnehmens nicht benötigter Befehle wäre auch ein gangbarer Weg, nicht jeder benötigt LOCALDEV oder TRIGGER.

Im Handbuch vermisste ich außerdem mindestens ein Beispiel pro Befehl.

Bei dem stolzen Preis des Interfaces kann man eine in den genannten Punkten ergänzte Bedienungsanleitung schon erwarten, wie dies normalerweise bei W&W auch üblich ist. Die mir vorliegende hatte wohl eher den Charakter einer Vorabversion.

Um es auch von anderen Rechnern mit RS232 Schnittstelle steuern zu können benötigt man aber noch eine weitere Anleitung für die Fähigkeiten des Interfaces selbst, diese lag mir bislang nicht vor.

Insgesamt machte das Paket aber einen recht ordentlichen Eindruck und versagte bis auf das wieder Auffwecken der IL-Geräte einwandfrei.

mm

HILFE Umwandlungsprogramm TIFF ⇨ GROB gesucht

Selt Monaten bin ich auf der Suche nach einem Programm, daß es mir ermöglicht, auf dem PC erstellte Grafiken in Grob's umzuwandeln. Dabei habe ich auch ein paar Anwendungen gefunden, die meisten scheiterten jedoch an der Unmöglichkeit die Größe des Grob's bestimmen zu können.

Vor kurzem stieß ich auf ein Programm, das sich VGA2GROB nennt und laut VGA2GROB.DOC von Mallet Vicent (F) geschrieben wurde. Dieses Programm erlaubt es den oberen linken Teil des Bildschirms in ein Grob umzuwandeln, es stellte sich aber als zu umständlich heraus die gewünschte Grafik in den oberen linken Rand zu bewegen, um es dann übertragen zu können.

Für die Programmierer unter Euch hier eine kleine Anregung: Da ich sehr viel schreibe und meine Texte gerne mit Grafiken versehe, arbeite ich sehr oft

mit MS-WORD 5.0 und der dazu gelieferten Anwendung CAPTURE.COM. Dieses Programm erlaubt es einen beliebigen Bildschirmausschnitt zu "capturieren", d.h. als Datei auf Platte zu schreiben.

Die neue Anwendung könnte wie folgt aussehen:

1. Das Programm, welches Speicherresident bleibt, wird gestartet. (CAPTURE ermöglicht noch einige Parameter einzustellen).
2. Nun wird dafür gesorgt, daß die gewünschte Grafik auf den Bildschirm kommt.
3. Nach Drücken einer Tastenkombination (bei CAPTURE SHIFT+DRUCK) wird man aufgefordert dem GROB einen Namen zu geben und es erscheinen auf dem Bildschirm zwei waagrechte und zwei vertikale Linien, die sich mit den Cursortasten bewe-

gen lassen, um den gewünschten Ausschnitt und damit die Größe des Grob's zu definieren!!!

4. Das ganze schließt man mit RETURN ab und schon wäre das GROB erstellt.

Auf diese Weise wäre man "Formatungebunden" und es könnten sämtliche Grafikprogramme benutzt werden, um Grob's auf einem wirklich einfachen Weg erstellen zu können.

Leider wird mit dem von HP gelieferten Serial-Interface-Kit nur das Konvertierungsprogramm GROB2TIF beigelegt, warum nicht in beiden Richtungen?.

Für eine Lösung meines Problems wäre ich sehr dankbar und hoffe auf baldige Neuigkeiten.

Francisco Somoza (3664)
Markt Strasse 32
W6090 Rüsselsheim

BlackJack

Spielprogramm (ca. 8 kB mit allen Grafikteilen)

von Georg Hoppen

Das Programm ist "Shareware". Wer also dieses Programm länger als eine Woche benutzen möchte, sollte \$5 an folgende Adresse schicken, um die Spiellizenz zu erhalten:

"Ron Dippold
c/o Freezesoft
8318 Counterpane Lane
Juneau, Alaska 99801"

Vielleicht sollte ein Hinweis nicht fehlen, daß das Nichtübersenden der \$5 an den Autor erstens unfair und zweitens strafbar ist - die allgemeinen Regeln bei Shareware sollten beachtet werden, da dieser Vertriebsweg die ständige Weiterentwicklung von Programmen sichert ...

Zu den einzelnen Programmteilen:

Ich habe das Programm entgegen dem ausdrücklichen Begleittext minimal verändert. Zum einen habe ich den Einleitungsteil weggelassen (er folgt nachstehend) und zum anderen habe ich die englischen Texte in dem Programm durch entsprechende deutsche Texte ersetzt. Weggelassen wurden die Programme 'ABOUT' und 'AB2'; diese wiederholen eigentlich nur die Sharewareforderung und sind für das eigentliche Spiel nicht erforderlich.

'ABOUT'

```
< # 83 h # PDIM MBB PICT STO
PICT
{ # 0 h # 39 h } # 83 h # 7 h
BLANK
REPL 57 "(C)1990 Ron Dippold"
2 AB2 PICT { # 12 h # 4 h }
# 5 h # 31 h BLANK REPL
{ # 13 h # 5 h }
{ # 6F h # 33 h }
BOX 7 " 48 Blackjacj V1"
2 AB2 15 "--THIS IS SHAREWARE--"
1 AB2 21 "IF YOU KEEP THIS
YOU"
1 AB2 27 "MUST SEND $ 5 TO "
1 AB2 33 "--Ron Dippold JR.--"
1 AB2 39 " 8318 COUNTERPANE
LN"
1 AB2 45 "JUNEAU, AK 99801 "
1 AB2 { # 0 h # 0 h }
PVIEW 0 WAIT DROP PLAY >
```

'AB2'

```
< →GROB DUP SIZE DROP 133
SWAP - 2 / ROT
R→B 2 →LIST SWAP PICT 3
ROLLD REPL >
```

Die Veränderung der englischen Texte in deutsche betrifft folgende Programme: 'PLA2' ("Dealer busted!" wurde "Geber bankrott!"), 'PLA' ("Busted!" wurde "bankrott!"), 'QUIT' ("You had \$" wurde "Sie haben \$" "You have no money" wurde "Sie sind völlig blank" "You broke the bank" wurde "Bank wurde gesprengt"), 'BJC' ("DOUBLE BLACKJACK" wurde "Doppel Blackjack" "Your Blackjack" wurde "Ihr Blackjack" "Dealer Blackjack" wurde "Geber Blackjack"), 'WIN' ("You win" wurde "Sie gewinnen"), 'LOSE' ("You loose" wurde "Sie verlieren") und last but not least 'SHUFL' ("Shuffling" wurde "ich mische").

Eine Grafik (der Hauptteil - MBB) wurde ebenfalls dem deutschen Sprachgebrauch angepaßt, im oberen Kasten stand "You have : \$" nun steht dort "Guthaben: \$", im mittleren Kasten stand: "Place your Bet", dies wurde "Ihr Spieleinsatz". Wer also gerne die Originaltexte verwenden möchte, sollte diese wieder einsetzen. Der einfachste Weg für die Modifikation von MBB ist, den deutsche Code zu übernehmen (also GROB MBB abtippen) und dann nach 'MBB' EVAL PICT STO GRAPH mittels DOT+ und DOT- die Veränderungen vorzunehmen. Das Ergebnis mit PICT RCL in den Stack und dann wieder in 'MBB' abspeichern.

Spielregeln für diese Version von Blackjack sind wie folgt: Es wird mit 52 Karten gespielt, sind diese alle, wird neu gemischt (wenn schon einzelne Karten gezogen wurden, verfallen diese). Wer mehr als 21 Punkte hat, verliert sofort ohne daß der andere ziehen muß. Bildkarten zählen 10 Punkte, das Ass entweder 1 oder 11 Punkte. Es wird zunächst der Spieleinsatz

abgefragt (Standardwert sind \$50). Diese können über die Menütasten entsprechend verändert werden (maximal \$500). Nach dem Drücken von BET werden die Karten gezogen, mit HIT wird eine neue Karte verlangt, DBL verdoppelt den Einsatz (also auch die \$500) und der Spieler erhält nur noch eine einzige neue Karte - der Bankhalter kann beliebig viele dazu ziehen. STAND zeigt an, daß der erreichte Punktestand gelten soll.

Der Rest kann über das Spielen selbst ermittelt werden. Um weiterzuspielen ist jedesmal OKAY zu drücken, der Rechner prüft, ob und wieviel Geld noch vorhanden ist - bei Verlust des Startgeldes von \$1000 ist das Spiel zuende oder es kann nur noch der maximal vorhandene Betrag gesetzt werden. Mit QUIT kann das Spiel insgesamt verlassen werden (die Tasten sind sonst umfunktioniert ...). Das Spiel wird mit 'PLAY' gestartet, es werden ansonsten keine Menütasten benötigt.

```
485X B J H
< ERASE { 0 0 }
002 'PC' STO { } { }
    'PT' STO PBET 2
004 DEAL 3 SF 1 DEAL 3
    CF 2 DEAL 1 DEAL 11
006 CF
    IF 1 FC?
008 THEN PLA
    IF 1 FC?
010 THEN QUIT
    END
012 END
    > Bytes : 178
014 Checksum : # D73Eh
```

```
485X S H U F L
< "Ich mische" MSG
002 CDS OBJ→ DROP 1 200
    START RAND 52 * 1
004 + IP ROLL RAND 52 *
    1 + IP ROLLD
006 NEXT 52 →LIST
    'CDS' STO 52 'NC'
008 STO 1 CF PICT {
    # 0h # 0h } ROT
010 REPL
    > Bytes : 206
012 Checksum : # 651Dh
```

```
485X B → G R O B
< OBJ→ SWAP DROP 2
002 - 1 SWAP → s e
    * DEPTH ROLL s e
004 START DEPTH
    ROLL +
006 NEXT STR→
    >
008 > Bytes : 73
    Checksum : # 982h
```

```

485X PLAY
< RCLF 'FLGS' STO
002 -40 CF # 83h # 40h
PDIM ( # 0h # 0h )
004 PVIEW 50 'BET' STO
1 SF
006 IF NC
THEN 1 CF
008 END 1000 'M' STO
DO 1
010 IF FS?
THEN SHUFL
012 END BJH
UNTIL 2 FS?
014 END
> Bytes : 217,5
016 Checksum : # 3136h

485X DEAL
<
002 IF 1 FC?
THEN
004 IF NC
THEN → P
006 < 'CDS(NC)'
EVAL -1 'NC' STO+
008 DUP 'PC(P)' EVAL 8
* R+B P 1 - 29 *
010 R+B 2 →LIST OVER 3
IF FS?
012 "" 0 THEN DROP (
014 'PC(P)' EVAL 1 +
016 PC(P)' STO 1 →LIST
'PT(P)' EVAL SWAP +
018 'PT(P)' STO DROP
020 ELSE 1 SF DROP
END
022 END
> Bytes : 358
024 Checksum : # 6487h

485X BLNK
< PICT ( # 0h # 39h
002 ) # 83h # 7h BLANK
REPL MSG ,5 WAIT
004 PICT RCL SWAP 1 3
START DUP PICT (
006 # 0h # 0h ) ROT
REPL ,5 WAIT SWAP
008 NEXT DROP2
> Bytes : 170,5
010 Checksum : # C013h

485X SCOR
< 5 CF PT SWAP GET
002 OBJ → 0 SWAP 1 SWAP
START SWAP OBJ →
004 DROP2 "A23456789T"
SWAP POS
006 IF DUP NOT
THEN DROP 10
008 END
IF DUP 1 ==
010 THEN 5 SF
END +
012 NEXT
> Bytes : 128,5
014 Checksum : # BC1Ah

485X SHOWC
< → P
002 < # 0h P 1 - 29 *
R+B 2 →LIST 'PT(P)'
004 EVAL 1 GET 2 SF DC
2 CF
006 >
> Bytes : 113
008 Checksum : # 1D43h

485X PLA2
< PICT ( # 0h # 39h
002 ) # 83h # 7h BLANK
REPL 1 SHOWC 7 CF
004 DO 1 SCOR DUP 17
≥ SWAP DUP 7 ≥ 11 ≤
006 AND 5 FS? AND OR
IF
008 THEN 7 SF WHO
ELSE 1 DEAL 1
010 SCOR
IF 21 >
012 THEN
"Geber bankrott!"
014 BLNK WIN 7 SF
END
016 END
UNTIL 7 FS?
018 END
> Bytes : 280
020 Checksum : # CEBBh

485X PLA
<
002 IF BJC
THEN PICT ( # 0h
004 # 39h )
IF 'PC(2)' EVAL
006 3 > BET 2 * M > OR
THEN PBB 14 CF
008 ELSE PBB2 14 SF
END REPL ( # 0h
010 # 0h ) PVIEW 9 CF
DO 0 WAIT IP 15
012 CF
IF DUP 13 ==
014 14 FS? AND
THEN DROP 11
016 15 SF
END
IF DUP 16 ==
018 THEN DROP 9
020 SF PLA2
ELSE
022 IF DUP 11
==
024 THEN 2 DEAL
IF 1 FS?
026 THEN DROP
9 SF
ELSE 2
028 SCOR 5 CF
IF 21 ≤
030 THEN
032 DROP
IF 15
034 FS? THEN
036 9 SF PLA2
END
ELSE
038 DROP "Bankrott!"
040 BLNK LOSE 9 SF
END
END
ELSE DROP
044 END
UNTIL 9 FS?
046 END
> Bytes : 568,5
050 Checksum : # F6E2h

485X WHO
< 1 2
002 FOR I I SCOR
IF 5 FS?
004 THEN
IF DUP 12 <
006 THEN 10 +
END
END DUP PICT '
008 PC(I)' EVAL 8 * 10
+ R+B I 29 * 19 -
010 R+B 2 →LIST ROT 1
012 →GROB REPL
NEXT DUP2
IF ==
014 THEN DROP2
"Unentschieden" MSG
016 DROP
ELSE
018 IF >
THEN LOSE
020 ELSE WIN
END
END
022 END
END
024 > Bytes : 277
Checksum : # E0D3h

485X QUIT
< PICT ( # 0h # 39h
002 ) QPIC REPL
DO 0 WAIT IP
004 IF DUP 16 ==
THEN DROP
006 "Sie haben $" M
→STR + BYE
008 ELSE
IF 11 ==
010 THEN
IF M NOT
012 THEN
"Sie sind völlig bl
014 ank"
BYE
016 IF M
10000 ≥
018 THEN
"Bank wurde gespren
020 gt"
BYE
022 END
END 1
024 ELSE 0
END
026 END
UNTIL
028 END CLEAR
> Bytes : 287,5
030 Checksum : # 62EDh

485X BJC
< 8 CF 1 SCOR
002 IF 11 == 5 FS?C
AND
004 THEN 8 SF
END 2 SCOR
006 IF 11 == 5 FS?C
AND
008 THEN
IF 8 FS?
010 THEN 1 SHOWC
"Doppel Blackjack"
012 BLNK 0
ELSE 1 SHOWC
014 "Ihr Blackjack"
BLNK 11 SF WIN 0
016 END
ELSE
018 IF 8 FS?
THEN 1 SHOWC
020 "Geber hat Blackjac
k"
022 BLNK LOSE 0
ELSE 1
024 END
END
> Bytes : 314
026 Checksum : # 7E60h

485X BYE
< FLGS STOF MSG
002 PICT ( # 0h # 39h )
# 83h # 7h BLANK
004 REPL CLEAR 0 WAIT
DROP ERASE TEXT
006 KILL
> Bytes : 108,5
008 Checksum : # 1DBCh

485X WIN
< "Sie gewinnen"
002 MSG DROP BET
IF 11 FS?C
004 THEN 1,5 *
END
IF 15 FS?C
006 THEN 2 *
END 'M' STO+
008 > Bytes : 123,5
010 Checksum : # 60D5h

```


HP41 Utilities

von Dr. Martin Hochenegger

Bytes in der CCD Tastatur

HP41CY, CGD, DeskJet Drucker, (ThinkJet Drucker geht wegen des Rahmens nicht)

Das Gerüst habe ich schon im PRISMA 4/90/14ff vorgestellt, mit der Möglichkeit, die leeren Felder mit Tastenbelegungen auszufüllen. In diesem Rahmen sind nun die Bytes eingedruckt, die direkt mit der Tastatur (USER OFF/SHIFT/ entsprechende Taste) eingegeben werden können. Der Ausdruck erfolgt mit dem PC-8 Zeichensatz.

Bei den unten angegebenen Bytes sind die Druckzeichen in ASCII, PC-8 und ISO-German verschieden.

<u>BYTES</u>				
USER-OFF		ALPHA-ON		
97 a	98 b	99 c	100 d	101 e
15=SI	14=SO	10=LF	91 [93]
102 f	103 g	104 h	105 i	106 j
27=Ec	64 @	35 #	40 (41)
	107 k	108 l	109 m	SST
	APPEND			BST
110 n		111 o	112 p	CLA
:---		39 '	34 "	CLA
113 q	114 r	115 s	116 t	
95	7	8	9	
117 u	118 v	119 w	120 x	
38 &	4	5	6	
121 y	122 z	12=FF	33 !	
96 `	1	2	3	
31 ˇ	32=BLANK	59 ;	R/S	
92 \	123 {	Ø-BYTE	AVIEW	

	A	P	I
31		▼	
64	@	@	§
91	[[Ä
92	\	\	Ö
93]]	Ü
95			-
96	ˇ	ˇ	-
123	{	{	ä

Messung der Zeitdauer von Programmen

HP41, TIME, für PRGM 'T1' zusätzlich CCD-Modul

Im zu messenden Programm steht vor Beginn der Zeitmessung TIME/STOii, ein ein nicht benutztes Register. In diesem Fall wurde R10 gewählt. SIZE muß dann ≥ 11 sein! Am Ende der Zeitmessung XEQ'T1' oder 'T2'. Falls dies am Ende des zu messenden Programm ist, so kann man durch R/S, R/S in dieses zurückkehren.

```

01>LBL "T2"
02 TIME
03 RCL 10
04 HMS-
05 ENTER^
06 FIX 0
07 INT
08 CLA
09 ARCL X
10 "|H/"
11 X<>Y
12 FRC
13 E2
14 *
15 ENTER^
16 INT
17 ARCL X
18 "|M/"
19 X<>Y
20 FRC
21 E2
22 *
23 FIX 1
24 ARCL X
25 "|S"
26 PROMPT
27 END

```

Pitch & Points

HP41, Drucker (für ThinkJet nur PITCH, kein IL-Drucker)

Wenn man wissen will, welche Zeichengröße man für ein vorgegebenes Feld benötigt, so kann man dieses Programm verwenden.

Gegeben:

Länge des Schriftfeldes in mm L
Textlänge N
Höhe der Zeichen in mm H

Gesucht:

CPI (Pitch) Characters per Inch
Points inch/72

Die ausgegebenen Zahlen werden dann in die entsprechenden Steuersequenzen eingebaut.

```

01>LBL "PIPO"
02 FIX 1
03 "N="
04 PROMPT
05 "LAENGE="
06 PROMPT
07 /
08 25.4
09 *
10 "CPI="
11 ARCL X
12 PROMPT
13 "H="
14 PROMPT
15 2.8346
16 *
17 RND
18 "POINTS="
19 ARCL X
20 PROMPT
21 END

```

```

01>LBL "T1"
02 TIME
03 RCL 10
04 HMS-
05 CLA
06 ARCLI
07 "|H/"
08 FRC
09 E2
10 *
11 ARCLI
12 "|M/"
13 FRC
14 E2
15 *
16 FIX 1
17 ARCL X
18 "|S"
19 PROMPT
20 END

```

Dr. Martin Hochenegger
Heidelberger Landstraße 97
W6100 Darmstadt

T1

Zeile 1 von T1 (1-6) CCD-Barcodes Dr.M. Hochenegger
 Zeile 2 von T1 (7-12) CCD-Barcodes Dr.M. Hochenegger
 Zeile 3 von T1 (13-19) CCD-Barcodes Dr.M. Hochenegger
 Zeile 4 von T1 (20-20) CCD-Barcodes Dr.M. Hochenegger

T2

Zeile 1 von T2 (1-6) CCD-Barcodes Dr.M. Hochenegger
 Zeile 2 von T2 (7-14) CCD-Barcodes Dr.M. Hochenegger
 Zeile 3 von T2 (15-22) CCD-Barcodes Dr.M. Hochenegger
 Zeile 4 von T2 (23-27) CCD-Barcodes Dr.M. Hochenegger

PIPO

Zeile 1 von PIPO (1-3) CCD-Barcodes Dr.M. Hochenegger
 Zeile 2 von PIPO (4-8) CCD-Barcodes Dr.M. Hochenegger

Zeile 3 von PIPO (9-13) CCD-Barcodes Dr.M. Hochenegger
 Zeile 4 von PIPO (14-18) CCD-Barcodes Dr.M. Hochenegger
 Zeile 5 von PIPO (19-21) CCD-Barcodes Dr.M. Hochenegger

Für die geplagten Nichtcomputerbesitzer habe ich ein Dienstprogramm von mir beigefügt ('B-GROB'), um die ellenlangen GROB's einigermaßen bequem und relativ fehlerfrei in den Rechner zu bekommen. Die GROB's sollten in einen einheitlichen Abstand von je 19 Zeichen unterteilt werden (Bleistiftmarkierung) und dann über eine Liste mit Stringbegrenzungen "" eingegeben werden. Es empfiehlt sich, nach jeder Eingabe ein "Newspace" einzufügen, so hat man am schnellsten eine Kontrolle darüber, ob eventuell ein Zeichen zuwenig oder zuviel eingetippt wurde - für eine anschließende Fehlersuche hilft dies ungemein, wenn das Ergebnis vom Drucker protokolliert wird. Das Ergebnis ist ein Grafikobjekt im Stack, das in eine Variable gespeichert werden kann.

Wenn das Programm nicht ganz so funktioniert wie es sollte gibt es zwei mögliche Ursachen:

- a) die # Adressen sind nicht richtig abgetippt (Überprüfen) und
- b) es ist ein "BUG" im Programm, der "sporadisch" auftritt: Wenn der Bankhalter ziehen muß und es wird neu gemischt, gerät der Rechner ab und zu in eine "Endlosschleife" (erkennbar daran, daß die Sanduhr länger als 5-6 Sekunden erscheint ohne Textmeldung "ich mische"). Hier einfach den Rechner anhalten (ATTN) und neu starten. Ich habe

bisher noch nicht herausgefunden, warum dieser Fehler ab und zu auftritt - da dies aber selten der Fall ist, habe ich es dabei belassen.

Georg Hoppen

'MPIC2'

```
GROB 129 13 FFFFFFFFFF
FFFFFFFFFFFFFFFF10FF
FFFFFFFFFFFFFFFF
FFFFFFF10FFFFFFFFFFFF
FFFFFFFFFFFFFFFF10FFFFF
FFFFFFFFFFFFFFFF
FF10FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFF10FFFFFFFF
FFFFFFFFFFFFFFFFFFFF10
FFFFFFFFFFFFFFFFFFFF10
FFFFFFFFFFFFFFFFFFFF10
FFFF10FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF10
FFFFFFFFFFFFFFFF10FFFFF
10FFFFFFFFFFFFFFFF
FFFFFFFFFFFF10FFFFFFFF
FFFFFFFFFFFFFFFF10
```

'SUIT'

```
GROB 7 7 80C1E3F7F7C1E3
GROB 7 7 80C1A2F7A280C1
GROB 7 7 80412214224180
GROB 7 7 22559414224180
```

500C

Neues von der DESKJET-Familie

Es ist immer wieder erstaunlich, wie es Hewlett Packard nun schon zum dritten Mal schafft im unteren Preissegment Maßstäbe zu setzen, nicht nur für den Preis.

In letzter Zeit erreichten mich viele Fragen zu dem brandneuen DESKJET 500C, wobei das C für Color, also Farbe steht.

Ein echter Farbtintenstrahldrucker mit 300dpi (=Laser-Auflösung) für schlappe 1700.-DM, das war schon eine kleine Sensation.

Der Lieferumfang besteht auf den ersten Blick aus denselben Teilen wie beim 500 (ohne C). Halt, da ist ja noch ein sehr massiv erscheinendes Kästchen mit zwei Schächten, in die, Welch Zufall, die zwei Tintenpatronen hineinpassen, die sich ebenfalls im Karton befinden.

Wozu aber ein solcher Behälter?

Des Rätsels Lösung brachte dann der Blick unter die "Haube" unseres Druckers: Dort befindet sich, im

Fortsetzung auf Seite 84

Objekte suchen auf dem HP48

von Klaudius Chlebosch

Als frisch gebackener HP48SX Besitzer möchte ich auch einmal meinen Beitrag zum Informationsaustausch im PRISMA geben, hier in Form der kleinen Utility FIND zum Suchen von Objekten bzw. Variablen im hierarchischen Filesystem der beiden "großen" von HP:

Dankenswerterweise hat Hewlett Packard den HP48 mit einem hierarchischen Filesystem beglückt. Somit sollte es leicht möglich sein ein wenig Ordnung zu halten. Mir passiert es allerdings bisweilen, daß ich bestimmte Objekte nicht mehr wiederfinden kann, sprich nicht mehr weiß, wo ich bestimmte Objekte wiederfinden kann.

Dem hilft das Programm FIND ab: Es durchsucht alle *untergeordneten* Verzeichnisse ab dem aktuellen Verzeichnis und legt den Pfadnamen der gefundenen Objekte im Stack ab.

Wird ein solcher Pfad mit EVAL ausgewertet, dann befindet man sich im gesuchten Verzeichnis, in dem sich nun das Objekt aufrufen läßt.

FIND erwartet im Stack 1: den Namen des gesuchten Objektes.

FIND

```

< PATH 15 TVARS → search
  path dirs
  < dirs SIZE DUP
  IF
  THEN 1 SWAP
    FOR i 'dirs'
      i GET EVAL search
      FIND path EVAL
    NEXT
  ELSE DROP
  END
  VARS search
  IF POS
  THEN PATH
  END

```

>>

Assemblerprojekt

Ich möchte an dieser Stelle auch noch auf ein Assemblerprojekt eingehen, zu dem ich noch Kontakt zu Gleichgesinnten suche.

Die zwei Eigenschaften, die mich am HP41 am meisten gestört hatten sind beim HP48 ja beseitigt:

Es ist ohne zusätzlichen und teuren Hardwareaufwand möglich Assemblerprogramme zu entwickeln. Programme können auf einem Hostrechner assembliert und mittels KERMIT (binär) in den Taschenrechner heruntergeladen werden.

Einen solchen Crossassembler zu entwickeln ist mein Projekt.

Als Programmiersprache verwende ich ANSI-C. Meine Entwicklungsumgebung ist der ATARI-ST und TURBO-C 2.0, wobei ich besonderen Wert auf die Portierbarkeit der Ergebnisse lege.

Das entgeltige Produkt soll für CCD-Mitglieder kostenlos erhältlich sein, da ich keinerlei kommerzielles Interesse mit diesem Programm verfolge (welch gottlob doch noch existierende Ausnahme im Reiche der Hacker und Kraks: Kommentar von MM).

Der momentane Stand der

Entwicklung ist eine bereits funktionierende erste Testversion.

Leider steht mir als Grundlage der Arbeit nur das Buch 'Customize Your HP28' zur Verfügung. In diesem Buch sind leider keine Angaben über die Pseudo-OpCodes des HP71 IDS Assemblers enthalten, die wohl aus Kompatibilitätsgründen als Grundlage für einen selbstgeschriebenen Assembler dienen sollte.

Ich suche nun Gleichgesinnte mit hoffentlich etwas mehr Know-How bezüglich des HP48 Assemblers als ich, die mich mit Informationen über eventuell existierende Standards bei Assemblern für die Saturn-CPU versorgen können und auch in der Lage sind den Assembler zu testen.

Also Leute, seid nicht feige und lasst mich hinter den Baum. Ich hoffe baldigst das Telefon läuten zu hören um einen neuen Schub für meine Entwicklungsmotivation erhalten zu können.

Klaudius Chlebosch
Eschenweg 1
W7045 Nufringen

Tel.: 07032/82023 priv.
07031/620337 gesch.

FAHRPLAN

auf dem HP 48 SX

von Günter Schapka

Auslöser für dieses Programm war ein Grundprogramm von Markus Eißner (CCD 3203). Er ist auf die Bahn zur Heimfahrt angewiesen, hat ca. 15 Minuten zu Fuß zum Hauptbahnhof und kann sich als Student nicht generell auf eine Zeit festlegen.

Also:

Fahrplan abhängig von der aktuellen Zeit anzeigen. In seinem Grundprogramm waren das die nächsten 4 Züge inklusive Gleisnummer.

Ich habe das Programm so umgeschrieben und erweitert, daß im Fahrplan mit den Cursorstasten auf- und abgeblättert werden kann. Von den nächsten 5 angezeigten Zügen ist der obere invers dargestellt.

Drückt man ENTER, wird für den Zug der oberen Zeile ein Alarm mit 20 Minuten Vorlaufzeit (natürlich abhängig vom Weg zum Bahnhof anzupassen) und einem Wiederholungsintervall von 3 Minuten gesetzt. Dann kann man in Ruhe weiter arbeiten, bis man ans Losmarschieren erinnert wird.

Weiterhin habe ich die Auswahlmöglichkeit für 2 verschiedene Fahrpläne (Namen hier DZUG und FZUG) eingebaut, die "Arbeitsliste" ist in der Variablen 'GZUG' abgelegt.

Für den Fahrplan DZUG genügt ein ENTER bei der Input-Abfrage, drückt man zu dem D eine weitere Buchstaben-Taste, so wird FZUG ausgewählt.

Alarmer, bei denen die Vorlaufzeit vor der aktuellen liegt, werden gar nicht erst angenommen (Hinweis darauf ist die akustische Warnung), die Anzeige schiebt sofort den nächsten Zug nach oben und eine neue Entscheidung ist möglich.

Überschreiten des Datums ist be-

rücksichtigt, allerdings nicht wenn man rückwärts blättert, das ergäbe keinen Sinn. Beim Scrollen der Anzeige wird der einmal aufgebaute Bildschirm nur in den entsprechenden Teilbereichen für Abfahrtszeit und Gleisnummer überschrieben, das geht recht schnell.

Die Züge samt zugehörigen Gleisen standen ursprünglich in zwei Listen, jetzt zusammengefaßt in einer Liste aus doppelt belegten Listen, jeweils Zug / Gleis. Dadurch ist bei Fahrplan-Änderungen kein mühsames Abzählen der Position in den beiden Listen notwendig.

Beenden des Programms mit der Rückfeiltaste, dann bleibt kein unnötiger Datenmüll auf dem Stack zurück.

ZUG

(Flag -40=1 heißt immer Uhranzeige zum Fahrplan)

```

< RCLF → f
  < 2 FIX -40 SF "von:"
    { "D" { -1 0 } \Ga }
  INPUT
  IF "D" ==
    THEN DZUG
    ELSE FZUG
  END
  DUP 'GZUG' STO TIME
  → 1 t
  < 1 DUP SIZE DUP 's'
    STO SWAP OVER GET
    OBJ→ DROP2 t
    IF <
      THEN 1
      ELSE 2 /
    DUP
    DO DUP SWAP 'n'
      STO 1 SWAP GET
    OBJ→ DROP2 t
    IF >
      THEN
        2 / DUP n SWAP

```

```

-
ELSE
  2 / DUP n +
END
SWAP DUP ROT
SWAP .5
UNTIL ≤
END
END
0 RND 'n' STO 1 n
GET OBJ→ DROP2 t
IF <
  THEN 1 'n' STO+
END
CLLCD
"ZUG - ABFAHRZEITEN"
1 →GROB PICT STO
PICT RCL {#0d #14d}
GROB 14 6
77D141902790119077900000
REPL { #65d #14d }
GROB 19 6
E2E230122A00D2E210922220
EEEE10000000
REPL PICT STO ANZ
DO -1 WAIT DOK
UNTIL 51 ==
END
DEPTH DROPN
  { s n } PURGE 0 'GZUG'
STO CLLCD f STOF UPDIR
FF 2 MENU
  >
  >
  (Rücksprung in Ausgangsverzeichnis und Variablenmenü.)
ANZ
(x ist die Zeilennummer in der Anzeige)
< 1 CF 1 5
FOR x
  PICT #15d #4d #9d x
  * + 2 →LIST 2 FIX
  GZUG n x + 1 - DUP s
  IF >
    THEN s -

```

```

CASE 'x==1' 'n==1'
  AND
  THEN 1 SF
  END
  END
  GET OBJ → DROP SWAP
  DUP
  IF 10 >
  THEN " "
  ELSE " "
  END
  SWAP →STR SWAP + STD
  SWAP DUP →STR SWAP
  IF 10 <
  THEN " "
  ELSE ""
  END
  SWAP + + 1 22 SUB 2
  →GROB
  IF 1 x ==
  THEN NEG
  END
  REPL
  NEXT PICT RCL →LCD
  *
  
```

ALMS
(setzt Alarm)

```

< SWAP "ZUG: " SWAP 2
  FIX →STR + " Gls: " +
  ROT STD →STR + 8192
  180 * 4 →LIST STOALARM
  CLLCD "OK" 4 DISP 1500
  .2 BEEP 2 WAIT CLLCD
  51
  *
  
```

ALLMS
(erledigt Datum-Überschreitung und Vorlaufzeit für Alarm)
.2=20 Minuten Vorlauf

```

< DROP DATE
  IF 1 FS?
  THEN 1 DATE+
  END
  GZUG n GET OBJ → DROP
  SWAP DUP .2 HMS- HMS →
  →HMS DUP TIME
  IF <
  THEN
    IF 1 FC?C
    THEN 2500 .5 BEEP
    'n' INCR DROP ANZ
  ELSE ALMS
  END
  ELSE ALMS
  END *
  
```

Steuerung der Abstände für die immer konstant breite Anzeige.

1. Zeile inverse Darstellung !

DOK
(Tastaturabfrage) ENTER-Taste

```

< IP
  CASE DUP 51 ==
  THEN ALLMS END
  DUP 25 == Cursor aufwärts
  THEN 'n' INCR
  IF s >
  THEN n s - 'n' STO
  END
  ANZ
  END
  DUP 35 == Cursor abwärts
  THEN 'n' DECR
  IF 0 ==
  THEN s n - 'n' STO
  END
  ANZ
  END 55 == Rückpfeil=Ende
  THEN 51
  END
  END
  *
  
```

Variablen:

Fahrplanliste:
in Unterlisten jeweils Abfahrtszeit und Gleisnummer

```

GZUG
0
DZUG
{ { .39 6 } { 4.18 5 }
{ 5.09 4 } { 5.44 4 }
{ 6.04 5 } { 6.18 5 }
{ 6.5 6 } { 7.13 4 }
{ 7.23 5 } { 7.33 5 }
{ 7.57 5 } { 8.28 5 }
{ 8.52 6 } { 9.38 5 }
{ 10.3 5 } { 11.05 5 }
{ 12 5 } { 13 5 }
{ 13.33 5 } { 14.06 5 }
{ 15 6 } { 15.38 5 }
{ 15.58 5 } { 16.26 5 }
{ 16.45 4 } { 17.35 3 }
{ 18.36 5 } { 19.11 4 }
{ 20 5 } { 21.03 5 }
{ 22.24 5 } { 23.2 0 } }
FZUG
{ { .02 11 } { .37 15 }
{ 2.01 11 } { 4.18 5 }
{ 5.09 4 } { 5.44 4 }
  
```

```

{ 6.04 5 } { 7.23 5 }
{ 7.33 5 } { 8.28 5 }
{ 8.52 6 } { 10.3 5 }
{ 11.05 5 } { 13 5 }
{ 13.33 5 } { 14.06 5 }
{ 15 6 } { 15.58 5 }
{ 16.26 5 } { 17.35 3 }
{ 18.36 5 } { 19.11 4 }
{ 20 5 } { 22.24 5 }
{ 23.2 0 } { 23.45 16 }
}
  
```

Günter Schapka
Rebusgasse 11
DW6100 Darmstadt

Fortsetzung von Seite 81

Gegensatz zum Paintjet, nur eine einzige Halterung für eine Tintenpatrone.

Somit muß man eine der beiden Patronen irgendwo und irgendwie aufbewahren, so, daß sie nicht austrocknet. Genau dafür sorgt das oben erwähnte Schächtelchen, ein genauer Blick hinein zeigt die Gummidichtungen gegen das Austrocknen der Tinte.

Die schwarze Patrone ist die alte geblieben, die wir seit den Zeiten des ersten DESKJET kennen, natürlich inzwischen wischfest.

Die farbige Patrone enthält drei Fräben (gelb, cyan und magenta), aus den in 20%-Schritten alle Mischfarben erzeugt werden können, die Arbeit kann man aber getrost seinem neuen Freund überlassen.

Die Textfonts, deren saubere Gestaltung so manchen Laserdrucker-Fan erblassen ließ, sind allerdings nicht in Farbe ausdrückbar, der 500C druckt nur Grafikdaten in Farbe.

Skalierbare Windows-Schriften werden natürlich auch farbig gedruckt (diese sind reine Grafikdaten), natürlich auch schwarz mit der Farbpatrone; der Tintenverbrauch ist aber immens gegenüber der schwarzen Patrone.

Die Druckgeschwindigkeit wird mit der schwarzen Patrone mit 3 Seiten pro Minute angegeben, mit der Farbpatrone wähle man den Sprachgebrauch "4 Minuten pro Seite"; es ist also auch eine

Fortsetzung auf Seite 95

Dieser Artikel erscheint leider nicht ganz zeitgerecht, es ist aber schließlich jedes Jahr Ostern.



Andreas Stwiorok gibt in dem Buch "Sterne und Weltraum", 2/91, S.114, Verlag Sterne und Welt-

raum, München, ein BASIC-Programm zur Berechnung des Osterdatums ohne jegliche Zeitbeschränkung an.

Die dabei verwendete Originalliteratur, sonst un bequem erreichbar, ist dort zitiert und der wesentliche Teil davon auch abgedruckt.

Dieses Programm in den HP71B BASIC-Dialekt übersetzt läßt den verwendeten Algorithmus sehr gut erkennen.

Die ebenso vorgelegte HP48-Version ist mit den Kalenderprogrammen DAT und JD (PRISMA 5-6/90) zusammen im Verzeichnis CAL abgelegt.

Diese drei Programme sind dann in der Variable CST zusammengefaßt. Dazu wird die Liste { DAT JD ODAT } in die Ebene 1: gebracht und dann der Befehl MENU ausgeführt. Im abgedruckten Input- und Ergebnisdisplay ist diese Menübeschriftung zu erkennen.

Das 71er und das 48er Programm laufen in etwa gleich schnell.

Osterdatum

von Dr.G.Heilmann



```

ODAT
* STD
"Osterdatum
für das Jahr:"
" INPUT OBJ → DUP DUP2 7
MOD DUP + SWAP 4 MOD +
DUP + SWAP 19 MOD 3 PICK
1583
IF <
THEN 5 SF DUP 19 * 15
+ 30 MOD DUP DUP 6 * 6 +
5 ROLL + 7 MOD +
ELSE 3 PICK 100 / IP
DUP DUP 4 / IP - DUP ROT
3 / IP - 15 + 30 MOD 3
PICK 19 * + 30 MOD DUP
DUP 6 * 6 ROLL + 4 ROLL
4 + 7 MOD + 7 MOD +
END DUP 10
IF <
THEN 22 + 6 SF OC
END 9 -
IF 5 FS?C
THEN OC
END DUP 26
IF ==
THEN DROP 19
END SWAP ROT → d a
*
IF 'd#28 AND a≤10'
THEN DUP2 OC
END
* DUP 25
IF ==
THEN DROP 18
END DUP2 OC
*

```

```

OC
* SWAP ROT DROP2
"Ostern:" "
" +
" +
SWAP →STR + "." +
IF 6 FS?C
THEN "März "
ELSE "April "
END + SWAP →STR +
CLLCD 3 DISP 7 FREEZE
HALT
*

```

```

2: # 52315d
1: 130

```

```

R44 HALT 4 PRG
[ HOME CAL ]

```

```

Osterdatum
für das Jahr:

```

```

1931

```

```

DAT JC ODAT

```

```

Ostern:

```

```

5.April 1931

```

```

DAT JC ODAT

```

Dr.G.Heilmann

Obernhofen Straße 15

5408 Seelbach

```

10 ! ODAT. Osterdatum nach C.F.Gauß.
20 DESTROY ALL @ INTEGER A,D,E,P,Q,Y @ DIM F$(5)
30 F$="März" @ INPUT "Jahr?";Y @ A=MOD(Y,19) @ P=2*MOD(Y,4)+4*MOD(Y,7)
40 IF Y<1583 THEN SFLAG 1 @ GOSUB 110 ELSE GOSUB 120
50 IF E<10 THEN E=E+22 @ GOTO 100
60 E=E-9 @ F$="April" @ IF FLAG(1,0) THEN 100
70 IF E=26 THEN E=19
80 IF D=28 AND A>10 THEN 90 ELSE 100
90 IF E=25 THEN E=18
100 PRINT USING "'Ostern: ',K,'.',5A,' ',K";E,F$,Y @ END
110 D=MOD(19*A+15,30) @ E=D+MOD(P+6*D+6,7) @ RETURN
120 E=IP(Y/100) @ Q=E-IP(E/4)
130 D=MOD(19*A+MOD(Q-IP(E/3)+15,30),30) @ E=D+MOD(P+6*D+MOD(Q+4,7),7) @ RETURN

```



Wischfeste Tinte...

beim DESKJET ist ja nun endlich Wirklichkeit geworden.

Unter der gleichen HP-Nummer wie bisher wird nun ausschließlich diese neue Tinte verkauft.

Der Ausdruck "wischfeste Tinte" ist

dabei von Hewlett Packard selbst aufgebracht worden. Es handelt sich dabei nämlich in erster Linie nicht um wasserfeste Tinte, die kann es natürlich nicht geben, sonst wäre sie nicht flüssig...

Die Tinte soll vor allem Textmarkern und den so häufigen feuchten Fingern widerstehen.

Das dies tatsächlich funktioniert habe ich selbst ausprobiert. Mir scheint die jetzige Tinte auch ein klein wenig schwarzer zu sein als die Alte, das könnte aber subjektiv bedingt zu sein. mm

Der neue HP32S II

Der HP32S hat einen Nachfolger bekommen.

Neu hinzugekommen ist vor allem der Zahlentyp komplex sowie die Möglichkeit numerische Integrationen vorzunehmen.

Der eingebaute Gleichungslöser wurde erheblich erweitert und anwenderfreundlicher gestaltet.

Der "alte" HP32S ist im 32SII vollständig enthalten, neu hinzugekommen sind folgende Features:

Brüche

Gemeine Brüche können eingege-

ben, angezeigt und manipuliert werden.

Gleichungslöser

Algebraische Gleichungen und Formeln können direkt eingegeben und gelöst werden. Die Formeln kann man in dieser Form auch in Programmen verwenden; dabei erscheinen die Gleichungen in einer Programmzeile.

Einheitenkonvertierungen

Zusätzlich zu den bereits beim HP32S vorhandenen Konvertierungen stehen dem Anwender Funktio-

nen zur Massen-, Längen-, Volumen- und Temperaturkonvertierung zur Verfügung.

Weitere Funktionen

Mit dem HP32SII lassen sich direkt Populations-Standardabweichung sowie die x-te Wurzel aus y berechnen.

Eine zweite Shift-Taste ermöglicht ein schnelleres Zugreifen auf viele der eingebauten Funktionen.

Quelle:
HP PC-Info 6/91
mm

Alarmsystem für den HP71B ALMCCD1

Korrekturen

von Wulf-Thorsten Gerdts

Bug mit dem HP71B, Version 1BBBB

Aufgrund meines Artikels im PRISMA 5-6/90 habe ich eine ganze Reihe von Zuschriften bekommen. Erhebliche Probleme beklagen Besitzer des HP71B mit der Betriebssystemversion 1BBBB:

Der Rechner gibt keine Töne, sondern nur unwilliges Geknacke von sich, auch bleiben die Alarmtexte nur sehr kurz in der Anzeige.

Da ich einen HP71B mit der Betriebssystemversion 2CDCC besitze hatte ich bislang keine Chance diesen Fehler frühzeitig zu erkennen.

Für Insider:

ALMCCD1 verändert den VECTOR, um bei jedem Tastendruck ST 12 zu setzen, damit jeder Tastendruck den Alarmton beendet.

In der Interrupt-Routine wird dann mit A=IN geprüft, ob eine Taste gedrückt ist. Wichtig sind dabei die Bits IN(0) bis IN(13), Tastaturspalte 1-14 und IN(15) für ATTN.

IN(14) hat keine mir bekannte besondere Bedeutung, könnte aber eventuell von irgendwelchen Modulen gesetzt werden.

Aus irgendwelchen Gründen setzt der HP71/1BBBB aber im Interrupt das Bit 14 des IN-Registers, der HP71/2CDCC tut das nämlich nicht. Ich lasse ALMCCD1 in seiner Interruptroutine im Gegensatz zu vorher jetzt das IN-Bit 14 ignorieren, schon funktionierte alles prächtig.

Korrektur PRISMA 5-6/90

Beim Einlesen des ASCII-Files des HP71 Listings von ALMCCD sind alle ">" und "<" unter den Tisch gefallen, da der Ventura-Publisher diese als Steuerzeichen-Prefix verwendet.

Wir bitten vielmals um Entschuldigung für dieses Maleur, die meisten Anwender hatten es ja ganz schnell gemerkt. Wer sich das Programm von Henry Schimmer hatte schicken lassen hatte natürlich keine Probleme.

Aus bislang noch nicht zu ermittelnden Gründen war das unvollständige Listing zu allem Unglück auch noch doppelt abgedruckt, warum dies passieren konnte war bislang nicht zu klären.

Listing:

```

LEX 'ALMCCD1'
ID #E1
MSG 0
POLL PH
ENTRY WCC
CHAR #D
KEY 'ALM'
TOKEN #F0
ENDTXT
ATYP EQU #E220
TBL GOSUB TBL1
  NIBASC 'ONACKCAT'
  NIBASC 'CLREXCRE'
  NIBASC 'LSETOFF'
  NIBASC 'GET'
TBL1 C=RSTK
CDOEX
R3=C
P= 3
CC RTNCC
DP CDOEX
C+P+1
CDOEX
P= 5
RTNCC
PS GOSBVL #3064
GOSBVL #3FB8
GOSUB TBL

```

Wulf-Thorsten Gerdts
Bunzlauer Weg 18
W3002 Wedemark 1

```

B=0 A
PS1 B=B+1 A
C=DATO W
CSL W
?C=0 B
GOYES PSERR
CSR W
?A=C WP
GOYES PS2
GOSUB DP
GONC PS1
PS2 C=R3
CD0EX
CD1EX
C+P+1
CD1EX
C=B A
GOSBVL #2D28
C=C-1 A
GOSBVL #2426
REL(3) CC
REL(3) FIXP
REL(3) CC
REL(3) AA
REL(3) CC
REL(3) PSS
REL(3) PSS
REL(3) CC
REL(3) GPS
FIXP GOVLNG #2A6E
AA GOSBVL #3FD9
?ST=0 0
GOYES PX
PSERR GOVLNG #2E35
PX GOVLNG #3172
PSS GOSUB FIXP
GOSBVL #36CD
GOSBVL #379D
GOSBVL #36CD
GOVLNG #3A03
DC GOSUB TBL
A=0 A
A=DAT1 1
A=A-1 A
B=A A
DC1 C=DATO W
A=A-1 A
GOC DCE
GOSUB DP
GONC DC1
DCE A=R3
AD0EX
D0=D0- 2
P=P+1
LCASC ' '
P=P+1
GOSBVL #5423
D1=D1+ 1
A=B A
GOVLNG #5470
GPS GOSBVL #369D
LCHEX F2
?A#C B
GOYES V5
GOSBVL #2CEB
CR3EX
C=0 W
LCHEX 111121
CR3EX
V0 GOSUB V1
P= 0
C=R3
C=C-1 A
?C=0 P
GOYES V2
?ST=1 3
GOYES V3
GONC V4
V2 ?ST=0 3
GOYES V3
V4 CSR W
R3=C
?C=0 W
GOYES V8
LCHEX F1
?A#C B
GOYES V6
GOSBVL #2CEB
GONC V0
V8 GOSBVL #2A7A
GOC V6
V5 GOVLNG #2E2B
V1 CD1EX
D1=C
R2=C
GOSBVL #3FD9
RTNC
?ST=1 0
GOYES V3
?XM=0
GOYES V9
V3 C=R2
D1=C
ST=1 4
GOVLNG #2E66
V6 GOTO PX
V9 RTNCC
GSUXE GOTO AFN1
GSUX GOSBVL #F178
?A#0 W
GOYES GSUX1
GOSUB GG0
GOSUBL BU
GONC GSUXE
D1=D1- 5
D1=D1- 12
C=DAT1 B
ST=C
A=0 W
?ST=1 2
GOYES GSUX2
A=A+1 S
GSUX2 GOTO GNIX
GSUX1 ?A#0 S
GOYES GSUX3
GOTO GSUXZ
GSUX3 GOSUB GG0
GOSUBL B2
GONC GSUXE
A=0 W
GSUX4 CD1EX
?C>=D A
GOYES GSUX5
CD1EX
A=A+1 W
GOSUBL CN
?ST=0 1
GOYES GSUX6
GOSUBL NXA
GONC GSUX4
GSUX6 GOSBVL #1B322
GSUX5 GOTO GNIX1
V9Y RTNCC
AFNF C=RSTK
AFN1 P= 0
LCHEX 39
GOTO BC
SBSU GOSBVL #F178
SBSU1 GOSBVL #1B223
B=A W
GOSUBL B2
GONC AFNF
SBS1 CD1EX
?C>=D A
GOYES V9Y
CD1EX
B=B-1 A
?B=0 A
RTNYES
GOSUBL NXA
GONC SBS1
GSU GOTO GSUX
GSUXZ GOSUB SBSU1
GOC GGET
GOSUB GG0
C=0 W
P= 0
LCHEX 91
A=C W
ASRC
GOTO GNIX
GGET CD1EX
D1=(5) #2F6CF
DAT1=C A
GOSUB GG0
A=DAT1 W
GOSBVL #ED21
A=0 M
ASR W
GOSUB GG2
GOSUB GG1
C=DAT1 B
C=C+C B
CD0EX
GOSUB GS4
GONC GNOR
D0=D0- 10
D1=D1+ 10
GNOR D0=D0- 14
CD0EX
CSR B
R2=C
C=R3
GOSBVL #ED2C
CD1EX
CD0EX
AD1EX
R1=A
GOSBVL #1A460
D1=A
A=R2
GB10 C=DATO B
?A=0 B
GOYES GB20
A=A-1 A
GOSBVL #18504
D0=D0+ 2
GONC GB10
GB20 ST=1 0
P= 0
GOSBVL #181B7
C=R3
CD0EX
GOSBVL #F23C
A=DAT1 W
GOSUB GSTO
GOSUB GG1
GOSUB GS4
A=0 W
GONC GNA
C=0 W
C=DAT1 B
GOSUB CL
D1=D1+ 2
C=DAT1 B
R1=C
C=0 W
D1=D1+ 2
C=DAT1 6
GOSUB GS11
R2=C
C=A W
GOSBVL #ECB4
GOSUB CL1
R1=C
C=R2
SETHEX
GOSUB GS31
GOSBVL #1B322
P= 0
LCHEX 010
A=A-C X
LCHEX 01
?A#C X
GOYES NZM
P= 14
LCHEX 8
?A=C P
GOYES NZM
LCHEX 2
?A>C P
GOYES NZM1
LCHEX 5
NZM1 A=A-C P
NZM P= 0
?ST=0 2

```

```

GOYES GNA
A=A-1 S
GNA GOSUB GG2
GOSUB GS1
C=A W
GOSBVL #13335
C=0 W
C=D B
GOSUB CL
C=B B
GOSBVL #ED40
A=A+C W
P= 0
LCHEX 08
GOSUB GS2
GOSUB GS1
A=0 W
R1=A
GOSUB GS31
P= 0
LCHEX 04
GOSUB GS2
GOSUB GG1
B=0 W
B=B+1 S
A=0 W
C=DAT1 B
C=C+C B
GONC GNIC
A=A+B S
GNIC B=B+B S
GOSUBL CN
?ST=0 1
GOYES GNIP
A=A+B S
GNIP B=B+B S
?ST=0 0
GOYES GNIX
A=A+B S
GNIX ASRC
GNIX1 GOSUB GG2
GOTO WES
GG1 A=DAT0 B
P= 0
LCHEX F1
?A=C B
GOYES GGO
C=RSTK
GOTO WES
GG0 D0=D0+ 2
GOSBVL #FA35
GOSBVL #F186
CD1EX
D1=C
GOSBVL #ED3D
CD0EX
R3=C
GOSBVL #F7B0
D1=(5) #2F6CF
C=DAT1 A
CD1EX
C=0 W
RTN
GS2 C=0 XS

GOSBVL #1B322
A=A-C X
GG2 SETHEX
C=R3
CD0EX
GOSBVL #ED2C
CD1EX
C=A W
GOSBVL #F216
GSTO GOVLNG #F5F8
GS1 GOSUB GG1
C=DAT1 12
GOSUB CS1
GS11 GOVLNG #13229
GS31 GOSBVL #13252
GOSUB CL1
GS3 GOSUB CL
C=B B
GOSUB CL
C=D B
A=C W
RTN
GS4 D1=D1+ 13
C=DAT1 A
D1=D1+ 1
ST=C
?ST=1 3
RTNYES
RTN
CL1 C=R1
CL CSL W
CSL W
C=A B
RTN
REL(5) DC
REL(5) PS
WCC C=0 A
C=DAT0 1
D0=D0+ 1
B=C A
GOSBVL #2426
REL(3) EXT
REL(3) ALA
REL(3) AG
REL(3) W5V
REL(3) ACV
REL(3) W4V
REL(3) W2
REL(3) W1
REL(3) ALA
REL(3) GSU
W5V GOLONG W5
ACV GOLONG AC
W4V GOLONG W4
ALA GOSUB BU
GONC ALAE
D1=D1- 12
D1=D1- 5
C=DAT1 A
ST=C
ST=0 2
B=B-1 A
?B=0 P
GOYES ALA1

ST=1 2
ALA1 C=ST
DAT1=C A
ALAE GOTO COE
IV P= 0
LCHEX 36
GOTO BC
AG GOSUB SBSU
GONC IV
GOSUB AGO
WES GOTO WEX
BU SETHEX
P= 0
LCASC ' STPPA'
D1=(5) #2F558
A=DAT1 A
AD1EX
BU1 A=DAT1 W
?A=C W
GOYES BUE
?A#0 B
GOYES BU2
BUX RTNCC
BU2 D1=D1+ 16
D1=D1+ 16
A=DAT1 A
CD1EX
C=C+A A
CD1EX
GONC BU1
BUE D1=D1+ 16
A=0 A
A=DAT1 4
LC(5) ATYP
?A=C A
GOYES BUE1
C=0 A
RTN
BUE1 D1=D1+ 16
A=DAT1 A
D1=D1+ 5
C=0 A
RTNSC
ST D0=(5) #2F8C5
C=DAT0 W
CR2EX
DAT0=C W
D0=D0+ 16
ST1 C=DAT0 W
CR1EX
DAT0=C W
D0=D0+ 16
C=DAT0 W
CR0EX
DAT0=C W
RTN
INP GOSBVL #E90C
D1=D1+ 16
RTN
HMSB GOSBVL #F186
GOSUB INP
C=0 A
LCHEX 4
GOSUB RJA

GOSUB CS
ACEX W
GOSBVL #13274
R3=A
RTN
ARJ SETDEC
A=A+C X
RJU GOVLNG #12AE2
RJA GOSUB ARJ
RJA2 ACEX W
RJA1 B=0 W
D=0 W
D=C B
RTN
CS GOSUB CS1
CS2 B=C B
CS1 CSR W
CSR W
RTN
W2 GOSUB HMSB
CD1EX
R2=C
GOSUB SNA3
GOSUB CS1
CSRB
A=R2
AD1EX
A=DAT1 W
R2=A
GOSUB JPHR
GOTO PL2
W1 GOSUB HMSB
C=DAT1 A
A=0 W
C=C+1 P
GOC DATBL
GOSUB INP
DATBL R1=A
C=DAT1 W
R2=C
GOSUB JPH
PL2 SETHEX
C=DAT1 W
C=C+1 P
GOC TEXT
D1=D1+ 16
TEXT GOSBVL #BD31
SETHEX
C=R3
B=A A
BSRB
B=B+C W
R0=A
CD1EX
R1=C
C=C+A A
CD1EX
C=0 A
LCHEX BE
?C>A A
GOYES TOK
LCHEX 25
BC GOVLNG #9393
TOK D1=D1+ 14

```

```

C=0 W
P= 12
C=DAT1 P
C=C+B W
R3=C
EXT GOSUB ST
GOSUB BU
GOC TOK1
R2=C
?C#0 A
GOYES NNV
LCHEX 3F
GONC BC
NNV C=0 A
LCHEX 25
D=0 S
GOSBVL #1AF01
CON(5) #84C4
GONC CRF1
TFE GOVLNG #944D
CRF1 C=R1
D1=C
C=R2
DAT1=C W
D1=D1+ 16
LC(5) ATYP
DAT1=C A
TOK1 GOSUB BU
AD1EX
D1=A
D1=D1- 16
D1=D1- 16
D1=D1- 5
B=0 W
C=R3
C=C+C B
B=C B
CD1EX
GOSBVL #1AF01
CON(5) #133C
A=R0
GOSUB ST
C=A A
CR0EX
B=0 W
B=C B
AD1EX
C=R3
DAT1=C 14
D1=D1+ 14
C=R2
?C=0 W
GOYES CO5
DAT1=C 10
D1=D1+ 10
CO5 A=R1
CD1EX
GOSBVL #1308
C=R0
CD1EX
GOSUB POS
COE GOSUB P1
WEX GOLONG W4E
POS CD1EX

D0=C
GOSUB B2
C=DAT0 M
CSR M
CSR M
B=C M
POS1 CD1EX
D1=C
?C>=D A
GOYES POS2
C=DAT1 M
CSR M
CSR M
?C>B M
GOYES POS2
GOSUB NXA
GONC POS1
NXA C=0 A
C=DAT1 B
C=C+C B
AD1EX
A=A+C A
AD1EX
RTNCC
POS1 A=0 W
A=DAT0 B
A=A+A B
CD1EX
D1=C
C=C-A A
GOSBVL #ED3D
C=A B
D=C W
AD0EX
CD1EX
?C<A A
GOYES POS3
B=C A
B=B-A A
B=B-1 A
GOSUB SHI1
C=A A
GOSUB SHI
C=A A
A=A+1 A
GOTO POS4
POS3 ACEX A
C=C+D A
ACEX A
B=A A
B=B-C A
B=B-1 A
D1=C
GOSUB SHI
C=A A
GOSUB SHI1
CD1EX
A=C A
C=C+1 A
POS4 GOSBVL #1B41B
D0=C
GOSUB SHI
C=DAT0 S
GOSBVL #1308

GOSBVL #1B432
D0=C
GOSBVL #1B418
DAT0=C S
D=D-1 B
?D#0 B
GOYES POS4
D1=A
?C>A A
RTNYES
C=D W
GOSBVL #ED2C
D1=C
RTN
SHI1 C=C-1 A
SHI GOVLNG #1B435
JPH A=R1
GOSUB RJU
?A=0 B
GOYES JU
ACEX W
D=C W
GOSUB G
C=D W
A=C W
JU D0=(5) #2F8C5
DAT0=A A
A=R1
C=0 W
P= 0
LCHEX 8
GOSUB ARJ
SETHEX
?A#0 A
GOYES X1
CD1EX
RSTK=C
GOSUB SNA3
D0=(5) #2F8C5
GOSUB CS1
CSR B
GOSUB GS11
C=RSTK
CD1EX
GOTO X2
X1 GOSUB RJA2
P= 3
A=0 W
A=C WP
P= 0
GOSBVL #ED2F
GOSUB CS2
D=C B
GOSBVL #13304
X2 C=A W
D=C W
GOSUB KEIN
GOSUB MAL
JPHR A=R3
C=A+C W
CSL W
CSL W
P= 0
LCHEX 7

R3=C
C=0 W
CR2EX
A=C W
C=C+1 P
GONC X8
X81 RTNCC
X8 ?A=0 W
GOYES X81
CR3EX
P= 8
?A=0 S
GOYES X4
P= 12
X4 CPEX 13
LCHEX C
CR3EX
LCHEX 010
GOSUB ARJ
SETHEX
C=A W
GOSBVL #1B42C
B=0 W
GOSUB CS2
BSL W
BSL W
B=B+C B
?C=0 B
GOYES JD8
D=C A
GOSUB G
C=D A
B=C B
JD8 C=B W
R2=C
C=A W
GOSUB RJA1
GOSUB CS
A=0 W
ACEX B
CSR W
C=0 M
SETDEC
C=C+C A
A=A+C A
CSR A
C=C+C A
A=A+C A
GOSBVL #13274
GOSBVL #ED40
A=R2
C=C+A W
R2=C
RTNCC
MAL C=0 W
P= 1
SETHEX
LCHEX 1518
GOVLNG #ECBB
KEIN A=DAT0 B
?A=0 B
GOYES KX
KEIX P= 0
LCHEX 9

```

?A#C P	C=ST	GOYES ARE	A=R1
GOYES K1	RTN	C=DAT1 A	?A<C W
K2 D=D-1 W	WG1 C=ST	ST=C	GOYES NDOK
GOSUB D7B	ACEX P	ST=1 7	A=DAT0 B
P= 0	ST=C	C=ST	P= 0
?C#0 P	LCHEX 1	DAT1=C A	LCHEX 9
GOYES K2U	?A=C P	GOTO AFI	?A=C P
LCHEX 7	GOYES WGEN	ARE ST=0 5	GOYES NDER
K2U A=DAT0 B	C=C+1 A	C=ST	GOTO NEURE
ASR A	?A=C P	GOSBVL #1B41E	NDER LCHEX 93A80
GOSUB WG	GOYES WGEN	DAT1=C W	A=0 W
?A#C P	C=A P	CD1EX	A=C A
GOYES K2	RTN	RSTK=C	GOTO MIKO
KX GOTO KF	WGEN C=ST	GOSUB SNA3	NDOK CD1EX
K1 ?A#0 P	ACEX P	GOSUB CS1	RSTK=C
GOYES K3	RTN	B=C W	GOSUB SNA3
K4 GOSUB D7A	LC7 C=0 W	BSRB	B=C W
P= 0	P= 0	C=RSTK	C=RSTK
LCHEX 3	LCHEX 7	D1=C	RSTK=C
?B#C P	RTN	D0=C	CD1EX
GOYES K4	D7A D=D+1 W	D0=D0+ 14	C=0 W
D=D-1 W	D7B GOSUB KF	A=B W	D1=D1+ 2
K3 GOSUB D7A	A=A+1 W	R1=A	C=DAT1 10
P= 0	DIV7 GOSUB LC7	NEURE C=0 W	GOSUB SNA
?C#0 P	GOVLNG #EC7B	C=DAT1 12	C=RSTK
GOYES K5	G P= 0	GOSUB CS1	CD1EX
LCHEX 7	SETDEC	D=C W	GOSUB POS
K5 A=DAT0 B	DSRC	C=0 W	AF1 CD1EX
ASR A	?D=0 P	C=DAT0 10	R0=C
GOSUB WG	GOYES GR	GOSBVL #ED2F	GOSUB B2
?A#C P	LCHEX 8	C=C+D W	R1=C
GOYES K3	?D#C P	GOSBVL #13229	AF1 CD1EX
A=DAT0 B	GOYES GH	D1=D1+ 2	D1=C
LCHEX 8	DSL C	DAT1=C 10	A=R1
?A=C P	?D>=C P	D1=D1- 2	?C>=A A
GOYES KF	RTNYES	C=A W	GOYES AF3
?A=0 P	?D=0 P	GOSBVL #13335	C=0 W
GOYES KF	RTNYES	SETDEC	C=DAT1 14
K7 C=D W	GR C=RSTK	C=0 W	GOSBVL #1B438
K9 R0=C	GOVLNG #E920	D0=D0+ 2	ST=C
GOSBVL #13335	GH ?D>C P	C=DAT0 B	?ST=0 5
GOSUB KF	GOYES GR	D0=D0- 2	GOYES AF3
SETDEC	LCHEX 5	B=B+C W	C=DAT1 A
A=A-1 W	?D>C P	P= 0	C=C+C B
GOSUB DIV7	GOYES GH1	LCHEX 12	GONC AF2
SETHEX	LCHEX 2	MTE ?B<C B	GOSUB NKA
A=A+1 W	GH1 D=D+C P	GOYES MOK	GONC AF1
C=DAT0 B	DSL C	B=B-C B	AF3 P= 0
P= 0	LCHEX 6	A=A+1 A	LCHEX C4
?C=A P	?D=C P	GOTO MTE	GOSBVL #13601
GOYES K8	GOYES GR	MOK GOSBVL #13304	AF2 CD1EX
A=R0	C=C+1 A	D=C W	A=R1
GOSUB LC7	?D=C P	A=DAT0 B	?C<A A
C=C+A W	GOYES GR	?A=0 B	GOYES AF21
GONC K9	RTN	GOYES ND	C=R0
K8 C=R0	AGO C=DAT1 W	GOSUB KEIX	AF21 A=R0
D=C W	CD1EX	ND GOSUB KF	CD1EX
KF C=D W	R0=C	GOSUB MAL	AD0EX
A=C W	CD1EX	MIKO D1=D1+ 2	RTN
RTN	GOSBVL #1B438	C=0 W	SNA D0=(5) #2F755
WG ST=C	ST=C	C=DAT1 10	A=0 W
LCHEX 8	?ST=0 5	C=C+A W	A=DAT0 12
?A=C P	RTNYES	DAT1=C 10	CSL W
GOYES WG1	?ST=1 7	D1=D1- 2	CSL W

```

C=C+C W
?A<C W
GOYES SNA1
ACEX W
SNA1 ?A>B W
GOYES SNA2
ACEX W
SNA2 DAT0=A 12
SNA3 GOVLNG #125B2
B2 C=RSTK
D=C A
GOSUB BUL
C=D A
RSTK=C
RTNNC
D1=D1- 5
B3 CD1EX
D1=C
C=C+A A
D1=D1+ 5
D=C A
RTNSC
PH ABEX A
?A=0 B
GOYES C3
GOSBVL #23E3
CON(2) #F9
REL(3) P1
CON(2) #FC
REL(3) Y7
CON(2) #FE
REL(3) Y7
CON(2) #FD
REL(3) DS
CON(2) #2D
REL(3) FTY
CON(2) #2E
REL(3) LI
CON(2) 0
ABEX A
RTNSXM
C3 C=R3
D1=C
A=R2
D1=D1- 14
CD1EX
?A>C A
GOYES C6
D1=C
R3=C
LCASC ' ALM:1G'
DAT1=C 14
C6 RTNSXM
LI ABEX A
LC(5) ATYP
?A=C A
GOYES LI2
RTNSXM
LI2 D1=D1+ 16
D1=D1+ 16
CD1EX
B=C A
GOSUBL W5RA1
C=B A

D1=C
A=DAT1 A
GOSUB B3
GOSBVL #ED3D
CD1EX
R3=C
A=0 W
R1=A
A=A-1 A
R2=A
A=DAT0 B
LCHEX F1
B=C A
?A#C B
GOYES LI7
D0=D0+ 1
A=DAT0 A
ASR A
R1=A
D0=D0+ 5
C=DAT0 B
?B#C B
GOYES LI8
D0=D0+ 2
A=DAT0 4
LI8 GOSBVL #1B2D2
R2=A
LI7 A=R1
GOSBVL #1B2D2
C=R2
GOSBVL #ED3D
C=A A
GOSBVL #ED3D
R2=C
C=R3
CD1EX
LI3 CD1EX
A=R3
GOSBVL #ED0A
?C>=A A
GOYES ZZE
CD1EX
C=R3
CD1EX
D1=C
R3=C
C=R2
C=C+1 A
R2=C
A=C A
GOSBVL #ED2C
B=C A
GOSBVL #ED2C
?A<B A
GOYES LI4
?A>C A
GOYES ZZE
GOSUB ZZEIG
GOSUB ZZEIG
GOSUB CN
?ST=0 3
GOYES LI4
D1=D1+ 14
GOSUBL JHI

GOSUB ZZEIG
LI4 GOSUB NXA
GONC LI3
ZZEIG GOSUBL CRLF
GOSBVL #76AD
C=R3
CD1EX
RTNC
C=RSTK
ZZE GOSUBL W5RA
XM=0
RTNCC
FTY ABEX A
LC(5) ATYP
?A=C A
GOYES FTY1
RTNSXM
FTY1 GOSUB FTY2
BSS 3
CON(2) 5
NIBASC 'APPT '
CON(1) 1
CON(4) ATYP
NIBHEX FF
FTY2 C=RSTK
CD1EX
A=0 S
A=A+1 S
RTNCC
ST2 D0=(5) #2F901
GOLONG ST1
P1 GOSUB BUL
GOC P1J
P1E RTNSXM
P1J D1=D1- 5
D1=D1- 12
C=DAT1 A
ST=C
?ST=1 2
GOYES P1E
GOSUB ST2
D0=(2) #41
P= 1
C5 C=RSTK
DAT0=C A
D0=D0+ 16
P=P-1
GONC C5
C=D A
DAT0=C A
LBA GOSUB SNA3
B=C W
BSR W
BSR W
BSRB
GOSUB B2
LBO CD1EX
D1=C
?C>=D A
GOYES PE
GOSUB CN
?ST=0 1
GOYES LB2X
GOSUB NXA

GONC LB0
LB2X C=0 W
C=DAT1 12
CSR W
CSR W
?B<C W
GOYES LB3X
GOSUB SAN
GOLONG W8
SAN LCHEX C4
SAN1 GOVLNG #135FA
W8Z D1=D1+ 13
C=DAT1 A
ST=C
ST=1 0
ST=1 1
C=ST
DAT1=C A
D1=D1- 13
?ST=0 3
GOYES LB5X
?ST=1 2
GOYES LB5X
GOSUB AGO
LB5X GOTO LBA
LB3X BSL W
BSL W
B=B+B W
GOSUB SNA
PE GOSUB DSX
GOSUB ST2
D0=(2) #61
C=DAT0 A
D=C A
P= 1
C4 D0=D0- 16
C=DAT0 A
RSTK=C
P=P-1
GONC C4
RTNSXM
W4E GOTO WEND
W4 GOSUB WIPE
GOSUB B2
GONC W4E
PRA1 CD1EX
?C>=D A
GOYES W4E
PRN CD1EX
A=0 A
A=DAT1 B
GOSUB CN
?ST=1 0
GOYES PRAG
GOSUB NXL
GONC PRA1
PRAG D1=D1+ 13
ST=0 0
C=ST
DAT1=C P
D1=D1+ 1
A=A+A B
AD0EX
D0=D0- 14

```

```

?ST=0 3          IRT1 D0=(4) #F43C      A=C B          AD C=R0
GOYES PRANR      DAT0=C A          GOSBVL #12917  CDOEX
D1=D1+ 10        P= 0          P GOTO DSE     CD1EX
D0=D0- 10        LCHEX F          UA C=RSTK      R3=C
PRANR A=DAT1 B   OUT=CS          R3=C          D1=C
LCASC '1'       RTN          LCHEX CD       ?C>=D A
?A#C B          CN2 C=R3          GOSBVL #13601  GOYES Q
GOYES PRA3      CN3 CD1EX        LCHEX CE       GOSUB CN
D1=D1+ 2        CN D1=D1+ 13       GOSUB SAN1     D1=D1+ 14
D0=D0- 2        C=DAT1 A          C=0 A          ?ST=0 3
A=0 W           D1=D1- 13       LCHEX E        GOYES AD1
AD0EX          ST=C          B=C A          D1=D1+ 10
ASRB           RTN          LCHEX 809     AD1 C=R1
P= 2           DS C=D A          GOSBVL #1197D  ADOEX
GOSBVL #9723   R2=C          GONC UA1       A=A+C A
GOSBVL #2296   DSX GOSUB B2       LCHEX          ADOEX
GOSUB VNT      GONC DSE          OD4358454D4C41 AD1EX
LCHEX 20000    DS1 CD1EX         DAT1=C 14     A=A+C A
PRA4 C=C-1 A   ?C>=D A          UA1 C=R3       AD1EX
GOC PRA5       GOYES DSE1        RSTK=C         GOSBVL #1B1C7
?ST=0 12      CD1EX           GOTO P         GOC AK
GOYES PRA4     C=DAT1 A        CLEAR GOSUB BUL NES GOSUB CLEAR
PRA5 GOSUB IRT ST=C          D1=D1- 16     GOSBVL #1AF01
PRAA GOTO W4E ?ST=1 7        D1=D1- 16     CON(5) #133C
PRA3 AD1EX     GOYES DS2        D1=D1- 5      GOSUB AFI
CD0EX          GOSUB CN         A=R3          GOTO PRAA
B=C A          ?ST=1 1        D0=A          AK GOSUB NX1
D0=(5) #2F480 GOYES DS3        C=0 A         GONC AD
CD0EX          DS2 GOSUB NXL   C=DAT0 B      Q P= 0
GOSBVL #1308   GONC DS1         C=C+C B      LCHEX 0036
CD0EX          DSE1 LCHEX C4    A=A+C A      P= 9
A=DAT0 B       GOSBVL #13601   C=C A        GOSBVL #93BC
LCASC '>'      GONC DSE         B=C A        WEND SETHEX
?A#C B         DS3 GOSUB SAN    CD1EX        GOVLNG #8A48
GOYES PRA6     DSE C=R2         RTN          AM GOSBVL #1B223
LCASC ' '      D=C A          ACFN GOLONG AFN1 B=A A
DAT0=C B       Y7E RTNSXM     AC GOSBVL #F186 C=R3
PRA6 GOSBVL #14C8A Y7 C=D A      CD1EX        D=C A
GOVLNG #2620   R2=C          R0=C         C=R2
VNT GOSUB VNT1 R3=A          R1=A        CD1EX
INTOFF        GOSUB P1        GOSUB B2     AM1 CD1EX
A=0 W         GOSUB BUL        GONC ACFN    R3=C
A=IN         GONC DSE        R3=C        ?C>=D A
ASRC         D1=D1- 5        CD1EX        GOYES Q
A=A+A A     D1=D1- 12       R2=C        GOSUB NX2
A=A+A XS    C=DAT1 B      C=R0        B=B-1 A
?A=0 W      ST=C          CD1EX        ?B#0 A
GOYES I2    ?ST=1 2        C=DAT1 S    GOYES AM1
ST=1 12     GOYES DSE        A=R1        GONC NES
C=0 A       GOSUB B2        C=C+1 S     TIMH D0=(5) #2F8C5
I1 C=C+1 X   Y71 CD1EX      GOC AY       P= 0
GONC I1      ?C>=D A        GOTO AM     LCASC '00.00.00'
I2 INTON     GOYES P          NX1 C=R3    GOSUB XSPC
GOVLNG #DB   GOSUB CN3        NX2 CD1EX   LCASC ':00:00'
VNT1 ST=0 12 ?ST=1 0        NXL GOLONG NXA XSPC DAT0=C W
D0=(5) #2F43C GOYES Y72      BUL GOLONG BU D0=D0+ 16
C=DAT0 A     GOSUB NXL     AY GOSBVL #BD31 SPC LCASC ' '
D0=(4) #F971 GONC Y71      R1=A        DAT0=C W
DAT0=C A     Y72 AR3EX    R0=C        D0=D0+ 2
C=RSTK       A=A+1 A      C=R3        RTN
GOTO IRT1    A=A+1 B          U4 D0=(5) #2F8E5
IRT D0=(5) #2F971 GOC UA      D=C A      U3 CDEX W
C=DAT0 A     A=0 W          C=R2        GOSUB U2
ST=1 12     LCHEX 35        CD1EX       C=B A

```

```

GOSUB U2
C=A A
GOSUB U2
CDEX W
RTN
U2 DAT0=C 1
D0=D0- 2
CSR W
DAT0=C 1
D0=D0- 4
RTN
WIPE P= 0
D1=(5) #2F480
C=0 A
LCHEX D8
GOVLNG #1B0AF
ZEIG GOSUB TIMH
GOSUB CN2
?ST=0 0
GOYES ZEI1
LCASC 'x'
D0=D0- 2
DAT0=C B
D0=D0+ 2
ZEI1 LCASC ' '
?ST=0 1
GOYES ZEI2
LCASC 'P'
A=DAT1 B
A=A+A B
GONC ZEI2
LCASC 'a'
ZEI2 DAT0=C A
D0=D0+ 4
C=0 A
P= 3
CPEX 3
D1=D1+ 12
C=DAT1 XS
LCASC 'N'
?ST=0 3
GOYES ZEI4
LCASC 'R'
?ST=0 2
GOYES ZEI4
LCASC 'A'
ZEI4 DAT0=C A
D0=D0+ 4
C=0 A
GOSUB SPC
C=R3
CD1EX
C=0 W
D1=D1+ 2
C=DAT1 10
GOSBVL #130E5
GOSUB U4
GOSBVL #1B418
A=C B
GOSUBL CS
D=C B
GOSUB RNSU
C=R3
CD1EX
C=0 A
C=DAT1 B
C=C+C B
CD0EX
D1=D1+ 13
D0=D0- 14
C=DAT1 A
ST=C
D1=D1+ 1
?ST=0 3
GOYES ZEI5
D1=D1+ 10
D0=D0- 10
ZEI5 A=0 W
AD0EX
ASRB
GOTO ZSP
SEI? CD1EX
SE? C=R2
GOSBVL #ED2C
D=C W
CD1EX
D1=C
?C<D A
RTNYES
RTN
W5RA1 P= 0
LCHEX 4
D1=(5) #2F6C1
DAT1=C 1
W5RA GOSUBL BU
RTNNC
D1=D1- 12
D1=D1- 5
C=DAT1 S
CD1EX
D1=(5) #2F6C1
A=DAT1 S
DAT1=C S
CD1EX
DAT1=A S
RTNSC
QUIT GOSUB W5RA
GOSBVL #14C8A
WOUT GOLONG COE
W5 P= 0
GOSBVL #18534
GOSUB W5RA1
GONC QUIT
GOSUBL B2
CD1EX
R3=C
?C>=D A
GOYES QUIT
CDEX A
GOSBVL #ED3D
CDEX A
R2=C
CD1EX
W5B GOSUB SE?
GONC W51
AD1EX
D1=A
GOSUB CN
?ST=0 1
GOYES W51
R3=A
GOSUB NXL
GONC W5B
W51 GOSUB ZEIG
W52 GOSUB CLN
W53 GOSBVL #CF7
GOSBVL #152BA
GONC TA1
LCHEX 32
?B=C B
GOYES TA2
C=C+1 A
?B=C B
GOYES TA2
TA1 GOSBVL #212E
GOSBVL #10EE
GOC W53
TA2 A=B A
ST=0 0
GOSBVL #23E3
CON(2) 51
REL(3) DWN
CON(2) 163
REL(3) FDWN
CON(2) 50
REL(3) UP
CON(2) 162
REL(3) FUP
CON(2) 99
REL(3) QUIT
CON(2) 43
REL(3) QUIT
CON(2) 5
REL(3) TIM
CON(2) 4
REL(3) REPI
CON(2) 86
REL(3) XRU
CON(2) 91
REL(3) DEL
CON(2) 71
REL(3) AGOW
CON(2) 73
REL(3) RST
CON(2) 46
REL(3) RUN
CON(2) 0
DWN1 GOTO W51
FDWN ST=1 0
DWN GOSUB NX1
GOSUB SE?
GONC DWN1
CD1EX
R3=C
?ST=0 0
GOYES DWN1
GONC FDWN
UP C=R2
A=R3
B=A A
UP1 ?C=B A
GOYES DWN1
R3=C
GOSUB NX2
CD1EX
GONC UP1
FUP C=R2
R3=C
FUP1 GOTO W51
TIM GOSUB TIMH
C=0 W
DAT0=C W
GOSBVL #130DB
GOSUB U4
GOSBVL #1B418
A=C B
GOSUBL CS
D=C B
GOSUB U3
P= 0
D1=(5) #2F8C5
GOSBVL #15147
TIM1 GOTO W51
REPI GOSUB CN2
D1=D1+ 14
?ST=0 3
GOYES TIM1
GOSUB JHI
GOTO W52
JHI P= 0
D0=(5) #2F8C5
LCASC '0+00'
DAT0=C W
D0=D0+ 16
LCASC ':00:00+0'
DAT0=C W
D0=D0+ 16
LCHEX 003030
DAT0=C 6
A=0 W
A=DAT1 B
?A#0 B
GOYES RN3
GOTO RN2
RN3 ASR W
GOSUB RN1
NIBASC 'SaSoMoD1'
NIBASC 'MiDoFrWe'
RN1 C=RSTK
A=A-1 A
A=A+A A
A=A+A A
A=A+C A
AD0EX
C=DAT0 A
D0=(5) #2F8C5
DAT0=C A
A=0 A
A=DAT1 1
GOSUB RN4
NIBASC '12345+-'
RN4 LCHEX 6
?A<C P
GOYES RN5
A=A-1 A
A=A-1 A

```

```

RN5 A=A+A A      ?ST=0 0      C=R3      LCHEX E00
C=RSTK           GOYES XRUI      GOLONG PRN  GOSUB TUT
C=C+A A         ST=0 0           W8 D1=D1+ 12 T59 GOTO OUT
CD0EX          GONC XRUI2        P= 0      T6 LCHEX 4
C=DAT0 B       XRUI ST=1 0      A=0 A     T61 ST=C
D0=(5) #2F8C9  XRUI C=ST      A=DAT1 P  C=0 W
DAT0=C B       DAT1=C A        D1=D1- 12 LCHEX 60
RN2 D0=(5) #2F8CF D1=D1+ 2      GOTO W51  R0=C
D1=D1+ 2      DEL C=R2        ?C<A P   ST=0 11
C=DAT1 A      R4=C           GOYES OUT T62 B=0 W
GOSUB U2      GOSUB CLEAR      R1=A     B=B-1 P
D1=D1+ 2      GOSBVL #1AF01     GOSUBL VNT C=R0
C=0 W         CON(5) #133C     W81 C=R1  GOSBVL #EBEB
C=DAT1 6      C=R2           REL(3) OUT P= 0
SETHX        CD1EX          REL(3) T1  GOSUB TUTE
GOSBVL #13229 D1=D1+ 16      REL(3) T2  GOC T59
D=C W        D1=D1+ 16      REL(3) T3  A=R0
GOSBVL #ECAF D1=D1+ 16      REL(3) T4  ?ST=1 11
SETHX        C=DAT1 A        REL(3) T5  GOYES T63
D0=(5) #2F8D5 AD1EX          REL(3) T6  A=A-1 A
GOSUB U2     D1=A           REL(3) T2  LCHEX 10
CDEX W       C=C+A A        REL(3) T4  GONC T64
GOSBVL #13252 D1=D1+ 5      REL(3) T6  T63 A=A+1 A
D0=(5) #2F8E7 CD1EX          T1 GOSBVL #EC5A T64 R0=A
RNSU GOSUB U3 A=R4           OUT GOSUBL IRT ?A#C B
P= 0         C=A S          GOLONG W8Z  GOYES T62
D1=(5) #2F8C5 A=0 A        R2=C         T2 LCHEX 100 ?ST=1 11
ZSP2 D1=D1+ 2 GOSUB SE1?     D=C A       GOYES T65
A=A+1 A      GOC DEL1       LCHEX 800  ST=1 11
C=DAT1 B     GOTO QUIT      GOSUB TUT  GONC T62
?C#0 B       DEL1 C=R3      GOTO WT     T65 C=ST
GOYES ZSP2   GOSUB SE1?     T31 P= 0   C=C-1 P
D1=(2) #C5   GOC RST2       LCHEX 400  GONC T61
ZSP D0=A     C=R2           GOSUB T33  WT P= 0
P= 0         R3=C           P= 0       A=R1
GOSBVL #18542 GOTO FDWN       LCHEX 300  LCHEX 10007
AD0EX       AGOW GOSUB CN2  T33 D=C A  ?A<C P
ST=1 4      ?ST=0 1       LCHEX 200  GOYES T59
C=R2        GOYES RST2     TUT GOSUB TUTE WT4 GOSUB TUTE
R0=C        GOSUBL AGO     RTNC       GOC T59
C=RSTK     GOSUB SE?      C=0 M     C=C-1 A
CD0EX      GONC RST2      D=0 M     GONC WT4
GOSBVL #17E15 CD1EX          GOVLNG #EB40 GOTO W81
C=R0        GOC RST3      T3 GOSUB T31 TUTE ACEX W
R2=C        RST GOSUB CN2  GOSUB T31 D0=(5) #2E3FF
CD0EX       D1=D1+ 13     GOSUB T31 C=DAT0 A
RSTK=C      ?ST=0 3       T32 GOTO OUT CSTEX
RTN         GOYES RST2    T4 LCHEX 40 ?ST=1 2
CLN D0=(5) #2F478 ST=1 1      GOYES TUTE1 TUTE1 CSTEX
C=DAT0 A    C=ST         GOSBVL #EC5A ACEX W
CSTEX      DAT1=C A      GOSUB TUTE RTNC
ST=1 9     D1=D1- 13     GOC T32   ?ST=1 12
CSTEX      GOSUBL AGO    C=ST      RTNYES
DAT0=C A    CD0EX        GONC T41  RTN
CRLF C=R2   RST3 R3=C     GOTO WT    T5 LCHEX 0B0
CD0EX       RST2 GOTO W51  D=C A
GOSBVL #17DC1 C=R2       LCHEX E00
C=R2        GOSUB CN2    GOSUB TUT
CD0EX       ?ST#1 0      P= 0
R2=C        GOYES RST2   LCHEX A00
RTN         GOSUB W5RA   D=C A
XRUI GOSUB CN2 D1=D1+ 13  GOSUB WIPE

```



Alternatives Schnittstellenpaket

Fortsetzung von Seite 84

Geschwindigkeitsfrage, welche Patrone man verwendet.

Wenn man sich diese doch nicht unwesentlichen Einschränkungen ansieht, so erkennt man recht schnell die eigentlich Anwendung dieses neuen Gerätes: Leute, die normalerweise Texte und Grafiken lautlos und gestochen scharf zu Papier bringen möchten, gelegentlich aber eine farbige Seite benötigen, und sei es nur eine Hardcopy des Bildschirms.

Für diese seltenen Fälle ist der Tausch der Patrone gerade noch zumutbar, es sind wirklich nur zwei Handgriffe, fast idiotensicher, ohne jetzt empfindlichen Mitbürgern auf den Schlips treten zu wollen...

Der 500C unterstützt nicht mehr die EPSON FX80 Emulationskassette.

Die normalen Schriftkassetten des DESKJET/DESKJET + sowie 500 können aber ohne Einschränkungen weiter verwendet werden.

Der 500C kann auch spezielle Overheadfolien bedrucken, das Ergebnis ist aber natürlich nicht so gut wie das eines Plotters, aber immerhin...

Es gibt inzwischen auch einen neuen Druckkopf (HP51626A) mit doppelt soviel Tinte bei gleichen Abmessungen wie der alte. Durch eine Veränderung im Innern der Patrone wurde mehr Platz für die Tinte geschaffen, auch wieder ein Beitrag zu mehr Umweltschutz durch weniger Müll.

In dieser Hinsicht ist HP immer wieder richtungweisend. mm

Ich habe in Darmstadt ein Produkt entdeckt, daß ich Euch kurz vorstellen möchte, weil es sehr interessant ist.

Es geht um ein Schnittstellenset für den HP 48SX zum IBM PC, das aus Software und Hardware (Kabel mit 9 oder wahlweise 25 poligen DSUB) besteht. Über die Software möchte ich näher berichten.

DataConverter in der Version 3.0 präsentiert sich u.a. mit folgenden Merkmalen:

Maussteuerung, Pulldownmenüs, mehrere Fenster gleichzeitig offen, Hotkeys usw.

Doch was kann die Software?

Es wird damit leicht, Programme für den HP 48 zu entwickeln. Außerdem kann man sich GROBs im Grafikmodus anschauen, sie in TIFF-Dateien konvertieren und derlei Dinge mehr.

Die Entwicklung von Programmen geht folgendermaßen vor sich:

Man kann einen Quelltext neu erstellen oder auch einen vom HP holen und editieren.

Mit einem eingebauten Editor (Fremdeditorbenutzung möglich) verändert man den Text und überträgt ihn per Knopfdruck zum HP.

Auf dem HP läßt man dann das Programm laufen, so wie man es normalerweise auch tun würde.

Ist ein Fehler drin, kann man diesen beheben (auf dem PC) und kann dann den Quelltext erneut übertragen. Das geht erheblich schneller, als auf dem HP den Editor aufzurufen und dort zu korrigieren trotz der Übertragung!

Außerdem kann man auf dem PC wesentlich mehr Zeilen gleichzeitig sehen als in den Anzeige des HP.

Man kann beim Editieren entweder

den Quelltext verwenden, den man nach dem Übertragen vom HP bekommen hat (mit allen ersetzten Zeichen). Oder man kann den Quelltext konvertieren, so daß z.B. wieder "«" statt "1/2" und "»" statt "†" im Text steht, was den Zeichen des HP entspricht. Diese Konvertierungsmöglichkeit ist sehr hilfreich, da man die gleichen Zeichen sieht, wie auf dem HP.

Die Dokumentation (39 kB auf Diskette) gibt nicht nur darüber ausführlich Auskunft, sondern auch über alle anderen Funktionen, über die ich aber nicht näher berichten will.

Eine Besonderheit besteht in der Möglichkeit, mit eigenen Programmen (sogenannten Usermodulen) den Funktionsumfang zu erweitern. Als Usermodul wird z.B. ein Programm zur Erzeugung von Postscript - Dateien aus GROBs mitgeliefert.

Insgesamt halte ich den Preis von 99 DM für angemessen, da das Erstellen von Programmen für den HP damit schneller von statten geht als mit dem "Interface Kit" von Hewlett Packard.

Bei letzterem muß man dauernd zwischen Editor und Kermit springen, was nun wirklich kein Spaß ist. Wenn es um das Weiterbearbeiten von Daten geht, ist man bei dem "Interface Kit" ganz aufgeschmissen.

Zu beziehen ist das ganze von:

EDV und Ind.elektronik

Richard Kreppner

Postfach 11 08 17

6100 Darmstadt 11

Tel.: 06151/318277

Fax.: 06151/316729

Stephan Werner
In den Eichgärten 43
W6338 Rechtenbach

Verkaufe:

HP 28S DM 300.-
 Thermodrucker 82240A DM 200.-
 HP 41CX plus Magnetkartenleser
 nur zusammen DM 600.-
 HP 48SX plus Handbuch DM 500.-
 Edgar Mauntz, Tel.: 06031/3430
 Mo-Fr ab 17 Uhr.

Plotter HP 7475

mit HP-IB Interface, neuwertig.
 Detlev Mainx, Tel.: 07024/3759.

Verkaufe gegen Gebot:

HP 41CV mit W&W Turbo
 Serial No 2222S42379
 W&W Ramboc 32 kByte
 W&WRambox 64 kByte
 Magnetkartenleser HP 82104A
 Serial No 2444S80583
 Thermo Printer HP 82143A
 Serial No 2042S40110
 Ladegerät für Thermo Printer
 Akku für Hp 41 8 Stck
 Ladegerät für HP 41 Akkus
 Time-Modul
 CCD-Modul
 Math I-Modul
 X-Function-Modul
 X-Memory-Modul HP 82181A 2 Stck
 Diverse Magnetkarten, teilweise beschrieben, teilw. neu
 Handbücher zu o.g. Positionen
 Diverse Literatur zum HP 41 aus dem
 Heldermann Verlag Berlin
 Software aus dem PCC-Modul
 Dr.-Ing. Peter Wutsdorff, Tassilostraße 4,
 6143 Lorsch, Tel.: 06251/51978.

Verkaufe:

HP 86B, Rechner komplett mit Speichererweiterung 128 k HP 82909A, Diskettenlaufwerk HP 9121D, Drucker HP 82906A, ROM-Drawer HP 82936A, Assembler-ROM HP 00087-15007, System-Monitor HP 82928A, Matrix-ROM's HP 00087-15004, GPIO Interface HP 829400A, I/O ROM HP 00087-15003, Advanced Programming ROM's HP 00087-15005 und diverse Software.
 Preis: Verhandlungssache.

für **HP 41**: Thermodrucker HP 82143A, DM 150.-.

Alles mit Handbüchern und in sehr gutem Zustand.

Peter Fütterer, Postfach 1145, 7740 Triberg, 07722/4749.

Verkaufe

Hardware: 41CX; Magnetkartenleser; Barcodeleser; Drucker 82143A; CCD-ROM; Advantage-ROM; 140 Magnetkarten; Akkus und Ladegeräte für Rechner, Drucker.

Literatur: Synthetic Programming on the HP41C; Best of PRISMA; Das HP-IL-System.

Alles in 1A+ Zustand! Nur komplett und ohne Diskussionen **DM 650.-**.

Werner Marschik, Alter Weg 16, 5910 Kreuztal 2, Tel. 02732/12305 nach 20 Uhr.

Verkaufe HP 48SX

serieller Anschluß an IBM, Gleichungslöser Bibliothek. Wenig gebraucht. Preis VS. Telefon 030/745 00 03 nach 19 Uhr.

Zu Verkaufen:

Serie 70:
 HP 71B, neuw. 500.-
 IL-Modul 82401A, neu 100.-
 Magnetkartenleser 82400A, neu 50.-
 HP 41 Translator Pac 82490A, neu 75.-
 Finance Pac 82482A, neu 50.-
 Tastenfeldschablonen f. HP 75, 5St. 20.-
 Serie 40:
 HP 48SX Version E, neu 600.-
 Infrarotdrucker 82240B, neu 150.-
 HP 41CV, gebr. 200.-
 Thermodrucker 82143A, neuw. 200.-
 IL-Modul 82160A, neu 120.-
 Time-Modul 82182A, neu 40.-
 Quad-RAM 82170A, neu 40.-
 Extended I/O 82183A, gebr. 50.-
 Plotter-Modul 82184A, gebr. 50.-
 Mathe-Modul 00041-14023, gebr. 25.-
 Statistik-Modul 00041-15008, neu 40.-
 Finanz-Modul 00041-15014, neu 40.-
 Games Pac 00041-15022, neu 30.-
 Securities Pac 00041-15026, neu 40.-
 Real Estate Pac 00041-15016, neu 40.-
 HP-IL:
 Cassettenlaufwerk 82161A, gebr. 250.-
 ThinkJet 2225B, gebr. 350.-
 Thermodrucker 82162A, gebr. 200.-
 Video-Interface 82163B, gebr. 100.-
 Thomas Mareis, Cranachstraße 1,
 8000 München 40.

Biete für HP 41 zum Verkauf:

Barcodelesestift HP 82163A (DM 100)
 Quad-Modul HP 82170A (DM 20)
 Drucker ohne Akku HP 82143A (DM 50)
 17 Rollen Thermopapier (DM 20)

Bücher: W. Meschede: Plotten und Drucken auf dem HP 41 Thermodrucker; W.C. Wickes: Synthetische Programmierung auf dem HP 41 (2. erw. Auflage) pro Buch DM 18.-.

Gerhard Deltau, In der Harth 11, 6342 Haiger 13, Tel.: 02774/3749 vorzugsweise nach 20.30 Uhr.

Biete an:

HP 75D mit 64k-Expansion Pod HP 82718A, opt. Lesestift und eingebautem US-Modem (Handbücher englisch) DM 600.-
 8k-RAM-Modul für HP 75 HP 82700A DM 120.-
 30 Blank Cards für HP 75/HP 71 DM 50.-
 Surveying Pack 00075-15012 für HP 75 DM 60.-
 Quad-RAM für HP 41C (mit eingebauter Lithiumbatterie) DM 40.-
 Uwe Berg, Tel.: 06174/623 89.

Verkaufe:

HP 71B DM 500
 HP Digital Casette Drive 82161A DM 400
 Video-Interface und Netzgerät

HP-IL 80-Column HP 82163 DM 250
 Bildschirm Zenith ZVM 1220-EA DM 120
 HP ThinkJet 2225BB DM 300

Module:

Text-Editor 5061-7240 DM 80
 Mathematics DM 80
 HP-IL Interface HP 82401A DM 120

Bücher:

HP 71 Referenzhandbuch
 HP 71 Benutzerhandbuch
 HP 71 Software Developers Handbook
 HP 71 Hardware Internal Design Spezifikation
 HP 71 Hardware Internal Design Vol I-III A+B Gesamt: DM 150
 Bei Kauf aller Posten Gesamt: DM 1800

Angebote an: Dr. Michael Wuttke, Am Nordbahnhof 81, 6100 Darmstadt, Tel.: 06151/71 68 61 nach 19 Uhr.

Suche HP 71 mit VER\$ ≠ 1BBBB

Forth-Modul
 CMT-Eprom-Programmer
 CMT-Eproms
 CMT-RAM-Box
 Dr. Hanspeter Waldmann, Schwendier Weg 19, 7959 Kirchberg, Tel. 07354/8729

Suche für HP 41C ein Extended I/O

ROM HP 82183A, VB DM 100.-.
 P. Jochen, 7410 Reutlingen, Heilbronner Straße 240, Tel. 07121/63 01 63.

Verkaufe

37 Stk. PRISMA Ausgaben 9/83 - 5/88, nur komplett, VHB DM 165
 HP 110 VHB DM 555

Suche günstigen "Zweitreechner" HP 10/11/41C/28C o.ä. Tel.: 07268/1697.

HP 35 neuwertig, gegen Gebot zu verkaufen. Tel. 0208/89 02 17.

Verkaufe Drucker HP 2225B mit IL-Anschluß komplett, zusätzlich: Staubschutzhaube, neuer Druckkopf, ca. 1500 Blatt Spezialpapier DM 600.-
 Horst Klittich, Ersinger Straße 31, 7530 Pforzheim, Tel. 07231/424 30.

Suche HP IL-RS Schnittstelle. Tel.: 030/621 65 15.

Kostenlos abzugeben 1 x HP 85, 1 x HP86 mit Diskettenlaufwerken und Handbüchern. Tel. 06171/786 51 ab 19.30 Uhr

Verkaufe CCD-Module für HP 41, DM 80.-. Suche CMT-RAM-Disc für HP Interface Loop.

Christoph Klug, Leibnizstraße 19, 3200 Hildesheim, Tel. 05121/324 78.

Suche: HP 41CX mit zwei eingebauten X-Memories (max. 300 DM) oder Doppel-X-Memory (max. 90 DM). Außerdem HP 32S, auch ohne Anleitung (max. 40 DM) und IR-Drucker 82240B (max. 120 DM). Tel. 05302/5378 ab 18 Uhr oder Wochenende.

Verkaufe:

HP 75c + 8k Erweit. DM 200

HP 41-71 Translator DM 70
 HP 48SX DM 550
 Alle Geräte mit Handbüchern.
 Günther Vollmer, 07472/8245 ab 18 Uhr.

HP 48SX DM 550
HP 71B DM 300
HP 28S DM 250
HP 125 mit kleinem Fehler DM 150
128 kB RAM-Card für Vectra (HP 82215A) DM 200
32 kB RAM-Card für Vectra (HP 82214A) beide RAM's orig. Verp. DM 70
Monitor HP 35721B (monochrom) DM 200
ThinkJet-Printer (Centronics) für 9 Volt DC mit Hp-Netzteil DM 350
SIMM (Single Inline Memory Module) 256 kB, 80 ns, 9 Bit DM 20
Thermopapier 57 mm 6 Rollen à 25 m orig. Verp. (HP 82045) 8 Mon. alt DM 25
Data Cassette für Laufwerk HP 82161, unbenutzt, orig. verschlossen DM 10
 Gysbert Hagemann, Alter Weg 1, 6653 Blieskastel 2, Tel. 06842/2805.

Biete an Hardware:

Lederkoffer (Corvallis) NP 800.- DM
 Barcode Lesestift
 IL-Drucker
 IL-Modul
 5 m Kabel-Verl. Kabel 2 x
 5 Cassetten für Cassettdrive incl. Programmen
 3 Tastaturschablonen
 12 Rollen Thermopapier
 Magnetkartenleser
 ca. 150 Magnetkarten zum Teil incl. Programmen
 HP 41-Overlaykit
 Financial Modul
 HP DIN A4 Drucker HP 82905B mit IL-Schnittst. (evtl. defekt)

Software Programme:
 FB-Heizung
 Wärmebedarfsrechnung
 Rohrnetzdimensionierung
 Brötje-Heizkörperberechnung
 Buderus-Heizkörperberechnung
 Schäffer-Heizkörperberechnung
 Einkommensteuerberechnung Corvallis
 ca. 50 sonstige kleine Programme
 + viel Literatur.

Obige Hard- und Software - Abgabe nur komplett Komplettpreis DM 500.-
 Bernd Altenhoff
 Pagenstecherstraße 34, 4500 Osnabrück, Tel. 0541/628 26/220 62.

Suche zur HP 41CX-Peripherie zu kaufen: HP 7470A/Opt 003 Plotter/Manual
 HP 82153A D Barcodeleser/Manual
 CMT HP-IL RAM-Disk mit RS232-Schnittstelle, 130 kB bevorzugt, Manual.
 Horst Schöffeld, Mellenseestraße 14, 1136 Berlin, Tel. 00372/511 02 51.

Rarität zu verkaufen
 HP 65 erster programmierbarer Taschenrechner aus dem Jahre 1975, sehr guter Zustand.
 Walter Winkler, Talweg 132, CH-8610 Uster/Schweiz, Tel. 0041/1/941 50 04.

Suche: HP 75 Module Math, I/O und Wandmodul; HP Modulsimulator 16k incl. Software; HP 75 original Tragetasche; evtl. Programmsammlungen mit Magnetkarten; z.B. HP 75 Assembler auf Magnetkarten + Cass. und Originalhandbuch. Suche Blank Cards und HP 75 Schablonen für die Tastatur. Preise VS.
 Suche auch Fotokopien von alten Prisma-Artikeln über HP 75 Tests und mit HP 75 Software. Gegen Unkostenerstattung.

Biete: HP 75 Forth Compiler bzw. Interpreter, Editor, Assembler, Debugger und Disassembler incl. Source; lauffähig mit und ohne RAM-Modul, 1 Kassette + original Handbuch + Lehrbuch DM 170.-

Biete HP 82718A 64k Ramfloppy für HP 75C/D nutzbar für Daten und Programmspeicherung, batteriegepuffert, resetfest, 300 Baud Modem eingebaut ohne FTZ bzw. ZZF (Betrieb am Postnetz strafbar!) NP über 2000 DM, jetzt 590 DM.

Peter Habicht, Lindenstraße 24, 6363 Echzell 1, Tel. 06008/7235 ab 13 Uhr.

Verkaufe:

Compaq-16-VGA-Grafikkarte
 1 Jahr alt, NP DM 1100.- für DM 700.-
HP 71B mit IL-Schnittstelle, Mathe-Modul, Kartenleser komplett DM 750.-
HP DeskJet RAM-Erweiterung
 256 kB DM 350.-
 Tel. 0431/32 80 32.

Verkaufe HP 71B

HP 71B + IL-Modul DM 550
 Kartenleser 82400A DM 155
 Texteditor 7240 DM 105
 71/41 Translator 7269 DM 105
 IL-Modul 82401A DM 105
 Tatzl: Prakt. Anwendungen mit dem HP 71 B DM 15
 J. Gruss: Textverarbeitung für HP 71 + Disk 3 1/2" DM 50
 Diskettenlaufwerk 3 1/2" 9114A + Akku und Netzteil DM 495
 4 Netzadapter 82066 Stück DM 15

Für HP 41C
 2 Speichermodule 82106A (je 64 Register zusätzlich) je DM 15
 -- Alles mit original Dokumentation, Handbücher, etc. --
 Peter Klaiber, Dahlegrund 10, 2100 Hamburg 90, Tel. 040/760 73 66, abends.

Zu verkaufen HP 71B Turbo mit Translator für HP 41, mit Magnetkartenleser, 32 k Modul, HP-IL 82401A Modul, wie neu DM 800

HP Diskette 9114B auch für Batteriebetrieb DM 600
 Display 8x40 von W&W DM 200
 Ladegerät für Diskette W&W DM 40
 HP Video Interface 92198 DM 150
 HP Thermodrucker Infrarot 82240 DM 50
 HP Netzadapt. f. Drucker 82241AB DM 20
 HP Netzadapter für HP 71B DM 20
 alle Bücher dazu, plus Nachnahme
 Gesamtpreis DM 1880

M. Rogos, Röntgenstraße 90, 7200 Tuttlingen, Tel. 07461/3796 ab 18.30 Uhr.

Aus meinem Vorrührgerätebestand biete ich folgende Geräte an:

HP 45711FD MS-DOS Computer: HP Portable Plus, 512 kB Arbeitsspeicher, 384 kB RAM-Disk, IL-Interface, RS232 Interface, ROM/EPROM-Halter mit div. Software (z.B. WordPerfect 5.0, Norton Utilities 4.0 jeweils im EPROM). Baujahr 1987, Akku im Juli erneuert, Preisvorstellung 2000.- DM.

HP 9114A IL-Diskettenlaufwerk, Baujahr 1984, Akku im August 1991 erneuert. Preisvorstellung 300.- DM.

HP 9114B IL-Diskettenlaufwerk, Baujahr 1986, Akku im September 1991 erneuert, Preisvorstellung 500.- DM.

HP 9114B IL-Diskettenlaufwerk, Baujahr 1986, unbenutzt originalverpackt, Preisvorstellung 800.- DM.

HP 82161A IL-Kassettenlaufwerk, Baujahr 1985, Akku neu, Preisvorstellung 300.- DM.

HP 2225 HP ThinkJet Tintenstrahldrucker mit IL-Interface, Baujahr 1986, Akku im Januar 1991 erneuert, Preisvorstellung 400.- DM.

Jörg B. Warmuth, Heron Computer, Tel. 030/791 20 73 (Mo-Fr 10-18, Sa 10-14 Uhr).

Impressum

Titel:
 PRISMA

Herausgeber:
 CCD - Computerclub Deutschland e.V.
 Postfach 11 04 11
 6000 Frankfurt am Main 1

Verantwortlicher Redakteur:
 Alf-Norman Tietze (ant)

Redaktion:
 Michael Krockner (mik)
 Martin Meyer (mm)
 Dieter Wolf (dw)

Herstellung:
 CCD e.V.

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

Druck und Weiterverarbeitung:
 Reha Werkstatt Rödelheim
 Biedenkopfer Weg 40a
 6000 Frankfurt am Main 90

Anzeigenpreisliste:
 Es gilt unsere Anzeigenpreisliste 3 vom Juni 1987.

Erscheinungsweise:
 PRISMA erscheint jeden 2. Monat.

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.

SERVICELLEISTUNGEN

BEST OF PRISMA

Schutzgebühr: 30,- DM

Nachsendedienst PRISMA

Schutzgebühr: 5,- DM pro Heft für Jahrgänge 1982-86
10,- DM pro Heft für Jahrgänge ab 1987

Inhaltsverzeichnis PRISMA

Schutzgebühr: 3,- DM in Briefmarken

Programmbibliothek HP-71

Die bislang in PRISMA erschienenen Programme können durch Einsenden eines geeigneten Datenträgers (3,5" Diskette, Digitalkassette oder Magnetkarte) und eines SAFU angefordert werden.

MS-DOS Inhaltsverzeichnis

Kann durch das Einsenden einer formatierten 360 kB oder 1,2 MB 5,25"-Diskette oder einer formatierten 720 kB oder 1,44 MB 3,5"-Diskette und einem SAFU angefordert werden.

ATARI Inhaltsverzeichnis

Kann durch das Einsenden einer 3,5"-Diskette + SAFU bei Werner Müller angefordert werden.

UPLE

Das UPLE-Verzeichnis mit der Kurzbeschreibung der einzelnen Programme sowie den Bezugsbedingungen kann gegen Einsenden von DM 10,- in Briefmarken angefordert werden.

Programme aus BEST OF PRISMA

- Eine Kopie der Programme von BEST OF PRISMA auf Kassette erfordert das Beilegen einer Leerkassette und eines SAFU.
- Für Barcodes von BEST OF PRISMA-Programmen gibt es folgendes Verfahren:
Schickt eine Liste mit den Namen und der Seitenangabe (der Barcodeseiten) an die Clubadresse, pro Barcode-Seite legt bitte 40 Pf., plus 2,40 DM für das Verschicken, in Briefmarken bei. Die Liste der verfügbaren Programme ist in Heft 3/88 auf der Seite 35 abgedruckt, sie kann gegen einen SAFU angefordert werden.

Der Bezug sämtlicher Clubleistungen erfolgt über die Clubadresse, soweit dies nicht anders angegeben ist, oder telefonisch bei Dieter Wolf:

(069) 76 59 12

Die eventuell anfallenden Unkostenbeiträge können als Verrechnungsscheck beigelegt werden, Bargeld ist aus Sicherheitsgründen nicht zu empfehlen; ist dies nicht der Fall, so wird Rechnung gestellt, dies macht die Sache natürlich nicht unbedingt einfacher, bzw. schneller.

Formvorschriften für Schreiben an die Clubadresse gibt es keine; das Schreiben kann durchaus handschriftlich verfasst sein, ein normaler Sterblicher sollte es noch lesen können. Vor allem den Absender und die Mitgliedsnummer deutlich schreiben!
(SAFU = Selbst Adressierter FreiuSchlag)

CLUBADRESSEN

1. Vorsitzender

N.N.

2. Vorsitzender

Alf-Norman Tietze (1909),
Sossenheimer Mühlgasse 10,
6000 Frankfurt 80, ☎ (069) 34 62 40, GEO1:A.N.TIETZE

Schatzmeister

Dieter Wolf (1734),
Pützerstraße 29, 6000 Frankfurt 90,
☎ (069) 76 59 12, GEO1:D.WOLF

1. Beisitzer

Norbert Resch (2739),
Tsingtauerstraße 69, 8000 München 82

2. Beisitzer

Werner Dworak (607),
Allewind 51, 7900 Ulm,
☎ (07304) 32 74, GEO1:W.DWORAK

MS-DOS Service / Beirat

Alexander Wolf (3303),
Pützerstraße 29, 6000 Frankfurt 90,
☎ (069) 76 59 12

ATARI Service / Beirat

Dr. Werner Müller (1865),
Schallstraße 6, 5000 Köln 41,
☎ (0221) 40 23 55, MBK1:W.MUELLER

Regionalgruppe Berlin

Jörg Warmuth (79), Wartburgstraße 17, 1000 Berlin 62

Regionalgruppe Hamburg

Alfred Czaya (2225), An der Bahn 1, 2061 Sülfeld,
☎ (040) 43 36 68 (Mo.-Do. abends)
Horst Ziegler (1361), Schüslerweg 18b, 2100 Hamburg 90,
☎ (040) 79 05 67 2

Regionalgruppe Karlsruhe / Beirat

Stefan Schwall (1695), Rappenwörthstraße 42,
7500 Karlsruhe 21, ☎ (0721) 57 67 56, GEO1:S.SCHWALL

Regionalgruppe Rheinland/Ruhrgebiet

Jochen Haas (2874), Roßstraße 27, 5000 Köln 30,
☎ (0221) 51 98 70

Regionalgruppe München

Victor Lecoq (2246), Seumestraße 8, 8000 München 70,
☎ (089) 78 93 79

Regionalgruppe Rhein-Main

Andreas Eschmann (2289), Lahnstraße 2, 6906 Raunheim,
☎ (06142) 46 64 2

Beirat

Wolfgang Fritz (125), Welfenstraße 35, 7500 Karlsruhe

Beirat

Peter Kemmerling (2466), Danziger Straße 17, 4030 Ratingen

Beirat

Martin Meyer (1000), Kelkheimer Straße 20, 6232 Bad Soden 1

E-Technik

Werner Meschede (2670), Sorpestr. 4, 5788 Siedlingshausen

Grabau GR7 Interface

Holger von Stillfried (2641), Am Langdiek 13, 2000 Hamburg 61

Hardware 41

Winfried Maschke (413), Ursulakloster 4, 5000 Köln 1,
☎ (0221) 13 12 97

HP-71 Assembler (LEX-Files)

Matthias Rabe (2062), Teichsiede 13, 4800 Bielefeld,
GEO1:M.RABE

Mathematik

Andreas Wolpers (349), Steinstraße 15, 7500 Karlsruhe

Naturwissenschaften

Thor Gehrman (3423), Hobeuken 18, 4322 Spockhövel 2,
☎ (02339) 39 63

Programmbibliothek HP-71

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

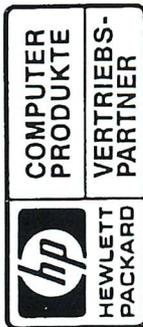
"Clubadresse"

CCD e.V., Postf. 11 04 11, 6000 Frankfurt 1, ☎ (069) 76 59 12

Sonderpreise für CCD Mitglieder



**HEWLETT
PACKARD**



HP 28 SD Taschenrechner
HP 48 SX Taschenrechner

DM 285,-
DM 598,-

HP 48SX Zubehör

Serieller Anschluß an IBM
Serieller Anschluß an MAC
Gleichungslöser Bibliothek
32 KByte RAM Erweiterung
128 KByte RAM Erweiterung
Programmer's Manual

DM 95,-
DM 95,-
DM 158,-
DM 127,-
DM 323,-
DM 38,-

Hewlett Packard Drucker

DeskJet 500 Tintenstrahldrucker 948,- DM
LaserJet IIIIP 2298,- DM



Zahlungsbedingungen: gegen Vorauskasse oder per Nachnahme jeweils zzgl. Versandkosten



H&G EDV Vertriebs GmbH

Münsterstraße 1 • 5300 - Bonn 1

☎ 0228/72 90 8-27/40 • FAX: 0228/72 90 838

Ihre Ansprechpartner:
Herr Endler
Herr Tubic

HP-41/71 RESTPOSTEN!!!

ALLGEMEINES ZUBEHÖR

Serie 10 Worksheets (50er Pack)	DM	20,-
Overlays (5er Pack)	DM	15,-
Hartlederhülle HP-Serie 10 Rechner	DM	60,-
Teakholz Tischständer	DM	60,-
Metallständer	DM	15,-
Holzständer für Aufklapprechner (HP-28 etc.)	DM	40,-
Ständer für HP-41, 21, 22, 32 u.ä. aus Acryl oder Holz	DM	30,-

SPEICHERERWEITERUNGEN FÜR DEN HP-41

Umbau von HP-41CX auf C4 TURBO 64k (ab # 2549520801)	DM	599,-
--	----	-------

MODULE FÜR DEN HP-41

HP X-Memory-Modul (82181A)	DM	150,-
HP X-Function-Modul (82180AD)	DM	150,-
HP Time-Modul (82182A)	DM	150,-
HP X-I/O-Modul (82183A)	DM	150,-
Grafik-ROM (16k-Modul für Geschäfts- u. Funktionsgrafiken)	DM	499,-

HP-41 ZUBEHÖR

Metall-Klappständer zur Montage unter dem HP-41	DM	15,-
Holzständer für HP-41	DM	15,-
Weiche HP-41 Echtlederhülle (schwarz)	DM	30,-
Hartlederhülle für HP-41 (offene Ausführung)	DM	99,-
HP-41 Overlays (Tastaturschablonen, schwarz, 10er Pack)	DM	25,-
HP-41 Staubschutzoverlay	DM	10,-
Zigarettenanzünderadapter für HP-Geräte	DM	59,-

SPEICHERERWEITERUNGEN FÜR DEN HP-71

CMT-CR-64 (64kByte Speichererweiterungsmodul)	DM	730,-
---	----	-------

HP-THINKJET IS-ANSCHLUSS (Vorführgerät)	DM	999,-
---	----	-------

CARD READER HP 71B	DM	345,-
--------------------	----	-------

EPROM-MODULE FÜR DEN HP-71

CMT-71-32KE (32kByte EPROM-Modul)	DM	299,-
-----------------------------------	----	-------

I/O-GERÄTE

SERIALPIUS II (Gehäuse mit 32k RAM und Zusatzdisplay)	DM	1.459,-
SERIALPIUS II (Gehäuse mit 129k RAM und Zusatzdisplay)	DM	1.899,-

SONSTIGES ZUBEHÖR

CMT-71 Wasserschutzgehäuse	DM	299,-
HP-71 Hartlederhülle	DM	110,-
HP-71 Holzständer	DM	55,-
HP-71 Staubschutzoverlay	DM	55,-
Firmware HP-II MODEM für HP-41 und HP-71	DM	700,-

BÜCHER

loux/Coffif: An easy course in using the HP-28C	DM	15,-
loux/Coffif: An easy course in using the HP-28S	DM	15,-
R. Hübner: Anwenderprogramme zum HP-28C/S	DM	15,-
Jarett: Erweiterte Funktionen des HP-41 leicht gemacht	DM	25,-
Kraus: Der HP-41C/CV in Handwerk und Industrie	DM	2,-
Kraus: Optimales Programmieren mit dem HP-41	DM	15,-
Meschede: Plotten u. Drucken mit dem HP-41 Thermodrucker	DM	5,-
Prankel: Grafik mit dem HP-41	DM	5,-
Saul: The Chemistry Collection - Seven HP-41 Programs	DM	28,-
Synthetic Quick Reference Card	DM	2,-
Synthetic Quick Reference Guide	DM	10,-
Wickes: Synthetisches Programmieren auf dem HP-41	DM	15,-
Horn: HP-71-BASIC made easy	DM	18,-

MC 5/2

Rechner u. Zubehör auf Anfrage

LIEFERUNG SO LANGE VORRAT REICHT!

HP-48SX PAKETANGEBOTE!!

HP-48SX + Mathematikmodul

HP-48SX + Mechanical Engineering Pac

HP-48SX + General Chemistry Pac

HP-48SX + Electrical Engineering Pac

Jedes Paket DM 899,-

Einzelmodule ohne HP-48SX DM 299,-

Alle Module mit ausführlichem deutschen Handbuch.

Weiteres Zubehör auf Anfrage!

HOTLINE: 02202/42021