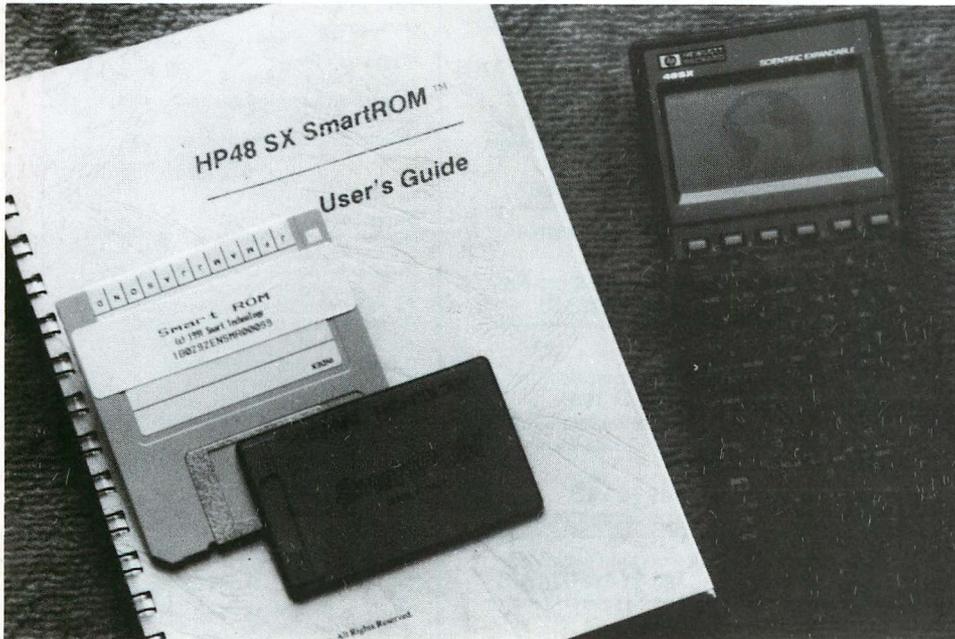


# PARISMA

Computerclub Deutschland e.V. • Postfach 11 04 11 • D-6000 Frankfurt am Main 1

1992

Nr. 1



Zwei interessante Produkte präsentiert die Italienische Firma SMART Technologies: Eine Steckkarte mit neuen Funktionen für einen breiten Anwenderkreis sowie einen Konverter von RS232 des HP48 auf Centronics Parallel für all diejenigen, deren Drucker keine zusätzliche serielle Schnittstelle (wie alle HP-Drucker seit DESKJET) haben. Im nächsten Heft wird dazu ein Testbericht erscheinen. mm

## Magazin

**Protokoll der Mitgliederversammlung**

## Taschenrechner

**RAM-/ROM-Editor 42S  
Master Mind**

## 41er

**Bemessung von Regenüberlaufbecken**

**Belastungsglieder und Volleinspannmomente**

**HP41 Utilities**

## 48SX

**Kabelsalat**

**Suche nach Objekt-Typen**

**AOS->UPN**

**Programmsammlung**

**Übersicht Pas de deux 1-7**

**Workspace-Ergänzung**

**Schnelle Telefonliste**

**Alarmer zurückholen**

**Orte aller Körper**

**Sommer-/Winterzeit  
automatisch stellen**

**Buchbesprechung**

**HP48 noch effektiver nutzen**

**Terminplanung**

## 71er

**Datumsberechnungen für die  
Astronomie**

**Centronics-Schnittstelle**

### Verkaufe:

Magnetkartenleser HP 82104 A mit ca. 300 Magnetkarten und Zubehör  
DM 300.-

### Suche:

Programm zum Berechnen vermaschter Rohrnetze nach dem Hardy-Cross-Verfahren für HP-41 oder HP-48

Jürgen Mang, Lindener Straße 7, 6751 Krickenbach, Tel.: 06307/525

### Verkaufe komplett:

- HP 41CX
- + Doppel X-Memory
- + CCD-Modul
- +MATH-Modul
- + 2 CCD-Akku-Sätze
- + CCD-Ladegerät
- Magnetkartenleser
- + 20 Karten
- Barcodelesestift
- IL-Modul
- IL-Thermodrucker
- + Ladegerät
- IL-Kassettenlaufwerk
- + Ladegerät
- + 3 Kassetten
- sonstiges Zubehör

Preis: DM 1.000.-

Dipl.-Ing. Peter Schuster, Bahnhofstraße 59, 6345 Eschenburg 4, Tel.: 02770/681

### Verkaufe:

HP 48SX, serieller Anschluß an IBM, Gleichungslöser Bibliothek, wenig gebraucht, Preis VS.

Telefon: 030/745 00 03 (nach 19 Uhr).

### Suche:

Tischrechner HP 97  
Magnetkartenleser für HP 41  
IR-Modul für HP 41

Bernhard Platz, Neue Straße 26, 7042 Aidlingen, Tel.: 07031/14-2044 (Mo-Fr 8.30 - 17.00 Uhr).

### Verkaufe für Serie 70:

|                            |       |
|----------------------------|-------|
| HP 71B                     | 500.- |
| IL Modul 82 401 A          | 150.- |
| Kassettenlaufwerk 82 161 A | 300.- |
| Thermodrucker 82 1162 A    | 200.- |
| Videointerface 82 163 B    | 80.-  |
| 32k RAM-Erweiterung        | 200.- |
| 4k RAM-Erweiterung         | 80.-  |
| Mathe Modul                | 90.-  |
| Text Modul                 | 80.-  |
| HP-IL Converter 82 166 A   | 80.-  |

Klaus Bendfeldt, Königsberger Straße 25, 5920 Bad Berleburg, Tel.: 02751/2114.

### Suche:

IL-Modul HP 82 160 A, IL-Kabel.

H. Migura, Tel.: 0211/71 35 17

### Verkaufe:

HP-IL Cassettenlaufwerk HP 82 161 A  
DM 250.-

HP-IL Videointerface PAC SCREEN zum Anschluß von HP-IL Rechnern an Monitor (einschließlich Monitor -berstein-)  
DM 250.-

Dipl.-Ing. Ulrich Kulle, Memeler Straße 26, 3000 Hannover 51

**Suche** HP 71 Speichererweiterung 64 KByte als Einschubmodul.

E. Hartmann, Steinbergstraße 14, 8064 Altomünster, Tel. 08254/8581.

### Suche:

HP-75 Module Math, datacom und HP-Modulsimulator 16k incl. Software.  
HP-75 original Tragetasche.  
Evtl. Programmsammlungen mit Magnetkarten, z.B. HP-75 Assembler auf Magnetkarten + Cassette und Originalhandbuch.

Suche Blank Cards und HP-75 Schablonen für die Tastatur.

Preise VS.

Suche CCD-Programme und vor allem LEX-Files auf Magnetkarten, z.B. nasty, Physcons, CCDBASIC, Peekpoke, Peepook, LIF1UTIL, wrap, derive und disassembler (die beiden letzten in BASIC).

Biete bis zu DM 25 pro LEX Magnetkarte.

### Biete:

HP-75 Forth Compiler bzw. Interpreter, Editor, Assembler, Debugger und Disassembler incl. Source; lauffähig mit und ohne RAM-Modul. 1 Cassette + original Handbuch + Lehrbuch DM 149.-.

Peter Habicht, Lindenstraße 24, W-6363 Echzell 1, Tel.: 06008/7235 ab 13 Uhr.

### Verkaufe:

|                                    |          |
|------------------------------------|----------|
| HP 97 Calculator (Tischrechner)    | DM 600   |
| HP-IL/IB-Interface (HP 82 169 A)   | DM 300   |
| HP 85                              | DM 600   |
| HP 85 B                            | DM 1.000 |
| HP IB-Interface (HP 32 937 A)      | DM 400   |
| HP 16k Memory-Module (HP 82 903 A) | DM 300   |

|                                   |        |
|-----------------------------------|--------|
| 2 X HP MassStorage ROM            | DM 200 |
| HP Input-Output ROM               | DM 200 |
| HP Plotter-Printer ROM            | DM 200 |
| HP Matrix ROM                     | DM 200 |
| HP Advanced Progr. ROM            | DM 200 |
| HP-Diskettenlaufwerk 9121         | DM 200 |
| HP-Nadeldrucker (wie Epson FX80+) | DM 200 |

Statikprogramme für HP 85 auf Anfrage.

Alle Geräte sind funktionsfähig, gut erhalten und mit den entsprechenden Kabeln und Handbüchern versehen.

W. Brettschneider, Tel.: 030/792 50 97.

**Suche** HP-97 oder HP-41 mit Kartenleser. Tel.: 07156/33 268.

Verkaufe Magnetkarten, DM -.50 pro Karte. Martin Meyer, Tel.: 06196/23 150

### Zu verkaufen:

2 x HP-ThinkJet mit HP-IL Interface, mit Akku und Ladegerät, VB DM 490.-  
HP 9114 B 3 1/2" Diskettenlaufwerk mit Akku und Ladegerät, HP-IL Interface, VB DM 590.-.

Telefon gesch. 06172/49 70 56  
privat 06003/7806 Hans Schramm.

**Suche** HP 41CX mit eckigen Display-Ecken (alte Version). C. Klug 05121/32 478.

### Verkaufe:

HP 48SX mit deutschem Handbuch DM 480.-. Telefon 07472/8245 ab 18 Uhr.

Suche **HP 82 166 A** IL-Converter.

C. Klug 05121/32 478.

### Zu verkaufen:

Serie 110:

- HP-110 Portable, Accu neu, dt  
DM800.-

Serie 70:

|  |          |
|--|----------|
| - HP-71B, neuw.                        | DM 100.- |
| - IL-Modul 82401A, neu                 | DM 50.-  |
| - HP-41 Translator Pac 82490A, neu     | DM 75.-  |
| - Finance Pac 82482A, neu              | DM 50.-  |
| - Tastenfeldschablonen f. HP-75, 5 St. | DM 20.-  |

Serie 40:

|  |          |
|--|----------|
| - Thermodrucker 82143A, neuw.          | DM 200.- |
| - Time-Modul 82182A, neu               | DM 40.-  |
| - Quad-RAM 82170A, neu                 | DM 40.-  |
| - Plotter-Modul 82184A, gebr.          | DM 50.-  |
| - Games Pac 00041-15022, neu           | 30.-     |
| - Securities Pac 00041-15026, neu      | DM 40.-  |
| - Real Estate Pac 00041-15016, neu     | DM 40.-  |
| - Magnetkarten f. 82104A, unben. 40St. | 10.-     |

HP-IL:

|  |          |
|--|----------|
| - Cassettenlaufwerk 82161A, neuw.          | DM 300.- |
| - ThinkJet 2225B, gebr.                    | DM 250.- |
| - Diskettenlaufwerk 9114A, gebr.           | DM 380.- |
| - Diskettenlaufwerk 9114B, gebr., Accu neu | DM 450.- |

Thomas Mareis, Cranachstraße 1, 8000 München 40, Tel.: 089/129 56 65.

### 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 die Preisliste 3 vom Juni 1987.

**Erscheinungsweise:**

PRISMA erscheint jeden 2. Monat.

**Auflage:** 1700

**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 abschnittsweise - nur mit schriftlicher Genehmigung des CCD.

**Inhalt**

**Magazin**

Protokoll der Mitgliederversammlung 4

**Taschenrechner**

RAM/ROM Editor beim HP42S 20

Master Mind 43

**41er**

Bemessung von Regenüberlaufbecken 32

Belastungsglieder und Volleinspannmomente 35

HP41 Utilities 39

**48SX**

Kabelsalat 5

Suche nach Objekt-Typen 12

AOS -> UPN 13

Neues aus Corvallis, USA 14

Programmsammlung Übersicht über Pas de deux 1-7 16

Workspace-Ergänzung 17

Schnelle Telefonliste 18

Alarmer zurückholen 19

Orte aller Körper 21

Sommer-/Winterzeit automatisch stellen 24

Buchbesprechung HP48 noch effektiver nutzen 52

Terminplanung 53

**71er**

Datumsberechnungen für die Astronomie 25

Centronics-Schnittstelle 29

Clubbörse 2

Serviceleistungen 59

Inhalt 3

Impressum 2

# Wieviele Ausgaben

kommen '92 heraus? Dies ist die häufig gestellte Frage am Telefon, wenn Mitglieder anrufen. Wenn wir das wüßten!

Die Redaktion besteht im Grunde nur noch aus Martin Meyer, Alf-Norman ist beruflich sehr ausgelastet und hat folglich nur noch sehr wenig Zeit für PRISMA, Michael Krocker und ich selbst unterstützen zwar Martin, soweit dies in unseren Kräften steht, aber die Hauptlast hängt an Martin.

Darin ist auch begründet, daß wir keine Regelmäßigkeit in die Erscheinungsweise von PRISMA bringen können. Das zur Verfügung stehende Material wird in Form gebracht, danach ist es jedesmal ein Kraftakt, den Umbruch zu machen und die Druckvorlagen für PRISMA zu erstellen. Dann ist jedesmal eine Erholungspause fällig, schließlich sind alle Beteiligten an PRISMA nebenbei tätig.

Das einzige was wir versprechen können ist: PRISMA wird weiterhin mit geballten Informationen zu den Taschencomputern von HP erscheinen. Dabei ist der bisherige Trend zu relativ dicken Heften nicht zu vermeiden, siehe oben.

Trotz der zu seltenen Erscheinungsweise ist die Menge der vermittelten Information größer geworden. Dies deshalb, weil durch die Verschiebung der Beiträge zum HP48 bei gleicher Seitenzahl die Informationsdichte steigt. 48er-Listings nehmen nunmal weniger Platz im Heft ein als 41er-Listings plus Barcodes.

Die Hefte, die '91 zuzuordnen sind, hatten einen Gesamtumfang von 244 Seiten, davon rund 230 redaktionelle Seiten, obwohl nur drei Ausgaben erschienen sind. Dazu kommt die Informationsverdichtung die schon angesprochen wurde.

Dies nährt bei mir die Hoffnung, die Mehrheit der Mitglieder des CCD möge denken wie der Anrufer, der sinngemäß sagte: "Es ist für mich nicht so wichtig wie oft PRISMA erscheint, Hauptsache PRISMA erscheint."

Dies soll nun keineswegs als Ausrede dienen, um den auch für uns unbefriedigenden Zustand der zu seltenen Erscheinungsweise auf ewige Zeiten festzuschreiben. Im Moment bleibt nur nichts anderes übrig.

Happy computing  
Euer Dieter Wolf

NB.: Es ist schon ein seltsames Gefühl das Protokoll der Mitgliederversammlung abzutippen und dabei nochmal den Ablauf nachzulesen. Dies war die letzte Zusammenkunft mit Gerhard Link. Die Erinnerung an ihn und seinen tragischer Tod läßt mich innehalten ob der jetzt so bizarr erscheinenden Formulierungen im Protokoll.

# Protokoll der Mitglieder- versammlung des Computer- club Deutschland e.V. vom 07.12.1991

von Rechtsanwalt Ludwig Widl

Ort: Intercity Restaurant Frankfurt,  
Hauptbahnhof Frankfurt

Beginn: 11.15 Uhr

Tagesordnung:

1. Begrüßung durch den Vorstand
2. Feststellung der Beschlußfähigkeit und anderer Formalitäten
3. Bericht des Vorstandes
4. Bericht des Beirats
5. Bericht der Kassenprüfer
6. Entlastung des Vorstandes
7. Neuwahl des Beirates
8. Haushaltsplan 1991
9. Prisma
10. Anträge
  - a. Antrag auf Satzungsänderung
  - b. Antrag betreffend Mitgliederversammlung 1992
  - c. Sonstige Anträge
11. Verschiedenes

Zu 1.: Die erschienenen Mitglieder wurden vom 1. Vorsitzenden des Vereins, Herrn Gerhard Link, begrüßt.

Zu 2.: Die Beschlußfähigkeit wurde festgestellt auf der Grundlage, daß die vorangegangene ordentliche Mitgliederversammlung vom 6. April 1991 gemäß § 14 III der Satzung nicht beschlußfähig war, gemäß § 14 III S. 2 und S. 3 der Satzung deshalb eine erneute Mitgliederversammlung einzuberufen war, welche ohne Rücksicht auf die Zahl der erschienenen Mitglieder beschlußfähig ist. Hierauf wurde in der Ladung im Heft Prisma März/Juni 1991 satzungsgemäß hingewiesen.

Auf Antrag des Vorstandes wird die

Tagesordnung dahingehend in ihrem Ablauf geändert, daß Tagesordnungspunkt 10. vor Tagesordnungspunkt 8. behandelt wird.

Zu 3.: Der 1. Vorsitzende berichtete über das Vereinsgeschehen und die Entwicklung des Vereins seit der letzten Mitgliederversammlung, sowie über die Möglichkeiten künftiger Entwicklung. Er berichtete über eine rege Beteiligung an den Vorstandssitzungen. Der Schatzmeister Dieter Wolf berichtete über die finanzielle Situation des Vereins, welche als ausgeglichen dargestellt wurde.

Zu 4.: Das Beiratsmitglied Martin Meyer berichtete über die Arbeit des Beirats. Dieser hat an den Vorstandssitzungen regelmäßig teilgenommen; die wesentliche Beiratsarbeit erfolgt in den Regionalgruppen.

Zu 5.: Der Kassenprüfer Schülle berichtet, daß der weitere gewählte Kassenprüfer Bluhm nicht mitgewirkt hat. An seiner Stelle hat das Beiratsmitglied Martin Meyer an der Kassenprüfung teilgenommen. Der Kassenprüfer bescheinigte dem Vorstand korrekte Erledigung der finanziellen Angelegenheiten und beantragte ohne Einschränkung die Entlastung des Vorstandes für das Jahr 1990.

Zu 6.: Dem Vorstand wurde einstimmig Entlastung erteilt.

Zu 7.: Zu Beiräten wurden gewählt die Mitglieder Stefan Schwall, Peter Kemmerling, Alexander Wolf, Georg Hoppen, Martin Meyer, Wolfgang Fritz und Werner Müller im Block bei zwei Enthaltungen und keiner Gegenstimme.

Zu 8.: Der Haushaltsplan für das Jahr

1991 wurde einstimmig genehmigt.

Zu 9.: Erscheinungsform und Inhalt der Vereinszeitschrift Prisma wurden besprochen.

Zu 10.: a. Auf Antrag des Vorstandes wurde § 14 III der geltenden Satzung neu gefaßt und beschlossen wie folgt:

"Die Mitgliederversammlung ist beschlußfähig, wenn mindestens 30 Vereinsmitglieder anwesend sind. Bei Beschlußunfähigkeit kann der Vorstand frühestens eine Stunde nach Beginn der Mitgliederversammlung diese für sofort am gleichen Ort mündlich nochmals einberufen. Diese dann stattfindende Mitgliederversammlung ist ohne Rücksicht auf die Zahl der erschienenen Mitglieder beschlußfähig. In der jeweiligen ordnungsgemäßen Einberufung der Mitgliederversammlung ist darauf hinzuweisen, daß im Falle der Beschlußunfähigkeit sie sofortige Einberufung einer sodann beschlußfähigen Mitgliederversammlung erfolgen kann."

Die Satzungsänderung wurde einstimmig beschlossen.

b. Auf Vorschlag des Vorstandes wurde von der Mitgliederversammlung bei vier Enthaltungen ohne einer Gegenstimme beschlossen:

"Die Mitgliederversammlung vom 7. Dezember 1991 ersetzt die ordentliche Mitgliederversammlung für das Jahr 1992. Die satzungsgemäßen Rechenschaftsberichte für das Jahr 1991 sind in der Mitgliederversammlung des Jahres 1993 nachzuholen."

c. Sonstige Anträge wurden nicht gestellt.

# Kabelsalat

## Die PC-Diät mit nur 48SX Kalorien

Ralf Pfeifer (116)

Wer außer dem 48SX noch einen Computer hat, kann seine Daten dort sichern. Damit die Übertragung funktioniert, bietet der 48SX eine Reihe von Funktionen an, die erheblich besser dokumentiert wurden, als die Software, die dem Übertragungskabel von HP beiliegt. Wer hier irgendwo Probleme hat, beginnt am Besten mit der Liste IOPAR. Zunächst löscht man sie, führt HOME aus und startet mit der blauen (!) Vortaste und I/O eine Übertragung, die man mit ON sofort wieder unterbricht. Dieser Vorgang hat eine geeignete IOPAR-Liste erzeugt. Wichtig: Die Systemlisten wie IOPAR oder PPAR sollten immer im HOME-Verzeichnis stehen, damit man in jedem Verzeichnis die eingebauten Funktionen verwenden kann. Mit dem Menü I/O SETUP braucht man danach nur noch den Übertragungsweg (wire) und die Übertragungsart (für die Übertragung zum PC empfiehlt HP den ASCII-Modus) wählen.

Auf dem PC muß man Kermit starten und einstellen. Beim ersten Anlauf klappt dies aber sicher nicht. Zunächst muß man Kermit auf den richtigen Ausgang (Schnittstelle) und auf 9600 Baud einstellen, denn bei mir blockiert COM1 immer eine Ratte (manche nennen dieses Biest auch liebevoll Maus) und Kermit startet nur mit 2400 Baud. Ohne den entscheidenden Tip des Michael Ritter von der 48-Gruppe an der FH Köln, wäre die folgende Anleitung nie entstanden: Sie verwendet den Uralt-Zeileneditor EDLIN, der eigentlich nur den Vorteil hat, daß er auf jedem DOS verfügbar ist; [ENTER] in eckigen Klammern bezeichnet die Eingabetaste (CR, Wagenrücklauf) auf dem PC, die in aller Regel zwar ENTER macht aber nicht so heißt.

Wir erzeugen nun eine Datei mit dem Namen K.BAT, bevor wir aber starten, bleibt noch die Frage, wohin K.BAT

soll. Mein Vorschlag: Zunächst führen wir:

```
TYPE AUTOEXEC.BAT
```

aus, denn dort befindet sich garantiert eine Zeile mit dem Befehl PATH. Diesem wiederum folgen die Namen einiger Verzeichnisse (z.B. C:\DOS oder C:\DRDOS ist sicher dabei). Da der PC alle angegebenen Verzeichnisse durchsucht, kann man Kermit von überall starten, denn dieses liest und schreibt nämlich in das gerade gewählte Verzeichnis, auch wenn sich K.BAT oder KERMIT.EXE nicht darin aufhält. Wir beginnen mit dem Verzeichniswechsel und dann mit

```
CD \DOS (ENTER)
EDLIN K.BAT (ENTER)
```

Der PC sollte jetzt "Neue Datei" anzeigen, eine andere Meldung, wie "Ende der Eingabedatei", deutet darauf hin, daß eine Datei K.BAT bereits existiert weshalb man in diesem Falle sofort mit

```
Q (ENTER) J (ENTER)
```

abbricht, und mit einem neuen Namen (z.B. KER.BAT o.ä.) nochmal von vorne beginnt.

```
I (ENTER)
```

schaltet in den Einfügemodus, dann kommt die eigentlich wichtige Zeile:

```
C:\HP\KERMIT SET PORT 2, SET
BAUD 9600, STAY (ENTER)
```

Mit Strg und C (oder Ctrl und C auf der engl. Tastatur) beendet man das Einfügen. Jetzt kann man es sich nocheinmal überlegen: Falls etwas nicht richtig war, kann man alles mit

```
Q (ENTER) J (ENTER)
```

abbrechen oder die kleine Batch-Datei mit

```
E (ENTER)
```

speichern.

Vor KERMIT muß der Pfad stehen (bei mir befindet es sich auf dem Laufwerk

C im Verzeichnis HP während K.BAT anderswo steht). Zwischen KERMIT und STAY dürfen noch weitere Einstellungen vorgenommen werden, die man sauber mit Komma trennen muß. Dabei sollte man allerdings auf eine sinnvolle Reihenfolge achten, es macht nämlich wenig Sinn, im o.g. Beispiel zuerst die Schnittstelle mit 9600 Baud einzustellen und sie dann zu wechseln! Zum richtigen Start ruft man nur noch K [ENTER] auf (.BAT braucht man nicht).

Manchmal bricht die Übertragung vom Computer mit der Meldung *Invalid Name* ab. Das Problem ist der Punkt (.), den der HP-48SX nicht in Namen erlaubt, wenn man Zahlen mit Dezimalkomma (wie in den meisten europäischen Ländern) und nicht mit Dezimalpunkt anzeigen läßt. Das ist in der Regel dann der Fall, wenn man vom Computer eine Datei übertragen will, die nicht nur einen Namen, sondern auch eine Namenserverweiterung hat, z.B.

```
FISH.GRO
```

Abhilfe: Entweder schaltet man den 48er auf den Dezimalpunkt um

```
(orange Vortaste) MODES
(orange Vortaste) PREV
```

und die Menütaste FM, drücken (das kleine Quadrat im Menülabel verschwindet), oder man startet die Übertragung auf dem Rechner mit RECN, und wählt einen erlaubten Namen, z.B. nur FISH, hier also entweder "FISH" RECN oder 'FISH' RECN.

Es könnte allerdings auch sein, daß der Name Kleinbuchstaben enthielt, die der 48 zwar unterscheidet, der PC aber nicht. Beispiel: Ein Programm soll die Funktion SIN des HP-48 erweitern. Unter diesem Namen kann es der 48 nicht speichern, weil Programme nicht die Namen von eingebauten Funktionen haben dürfen. Also speichert man es unter Sin. Schickt man

Sin zum PC, dann speichert dieser es unter SIN, da der ja nicht zwischen groß und klein unterscheidet. Holt man SIN später auf den 48 zurück, dann kommt es als (verbotenes) SIN an, weshalb man auch hier die Übertragung mit RECN durchführen sollte.

Möglicherweise hat Kermit aber auch Probleme mit dem Namen: Als ich ein Programm mit dem Namen CON übertragen wollte drehte Kermit auf dem PC durch und schrieb mir das GROB auf den Monitor !? (Siehe die Anmerkung der Redaktion am Schluß des Artikels).

Im Paket mit dem Kabel zum 48SX liegen einige praktische Programme bei. Eines davon ist GROB2TIF.EXE, welches GROB's vom HP-48SX in TIFF-Dateien umwandelt. Diesen wiederum bescheinigt die Bedienungsanleitung zu GROB2TIF.EXE, daß sie von vielen Textverarbeitungen, DTP und Grafikprogrammen verstanden wird. Die Frage ist also nur noch, wie kann man das nutzen? Mein Programm PRP, welches ich zum Programmausdruck auf dem IR-Drucker erstellt habe, konnte ich nun so umschreiben, daß die Ausgabe des Programms in ein GROB erfolgt. Das große Problem bei der Funktion →GROB besteht ja darin, daß es in einem String den LF (=LineFeed, Zeilenumbruch, entspricht 10 CHR oder blaue Vortaste und Dezimalpunkt) einfach ignoriert, und den String in ein einzeiliges GROB verwandelt. Um das zu vermeiden, habe ich folgende Strategie gewählt:

1. Den String in kleinere Strings zersägen. Jeder dieser kleineren Strings beginnt nach einem LF und hört vor dem nächsten auf. Die LF-Zeichen werden ganz herausgeworfen, denn sie erscheinen im GROB als unerwünschte Quadrate.
2. Ein großes GROB mit BLANK erzeugen, welches den String später aufnimmt. Jeder dieser kleinen Strings erhält im späteren großen GROB eine Zeile (klar, denn das LF-Zeichen hätte das ja auch getan, wenn man den String gedruckt hätte). Aus der Anzahl der kleinen Strings und der Größe der Buchstaben ergibt sich dann die Höhe des großen GROBs, die Breite legt dagegen der längste String fest.

3. Die kleinen Strings ins große GROB einfügen. Dazu verwandelt man die kleinen Strings zunächst mit 1→GROB, 2→GROB oder 3→GROB (je nach gewünschter Zeichenhöhe) in kleine GROBs, welche man dann mit REPL ins große GROB einfügt.

## Listings in Texten

Das Programm SPRG (Send Program) verwendet die gleiche Darstellung für Programme wie PRP. Das erscheint ziemlich unübersichtlich, allerdings erfolgt der Zeilenumbruch wie in der Anzeige des HP-48 (ausgenommen längere Strings), so daß die Suche nach falschen Befehlen im Programm mit Hilfe des Listings sehr einfach wird.

Durch die Übertragung der GROBs gibt es auch keine Probleme mit dem Zeichensatz, die Zeichendarstellung hat zwar die Qualität von Bauklötzchen (es wird die Punktmatrix des HP-48 verwendet) aber Sonderzeichen kommen korrekt zu Papier.

Bedienung: Das Programm SPRG erwartet in Ebene 1

1. den Namen eines Programms oder
2. den Namen eines Verzeichnisses aus dem alle Programme einzeln als GROBs übertragen werden sollen. SPRG ignoriert alles, was kein Programm ist und durchsucht auch Unterverzeichnisse.
3. SPRG nimmt auch eine Liste, welche Namen von Programmen und/oder Verzeichnissen enthält.

Um alle Variablen im Rechner auszu-  
drucken, könnte man

```
HOME VARS SPRG
```

ausführen.

Dabei sollte man folgendes beachten: Auf dem HP-48 dürfen Programme Namen mit bis zu 127 Zeichen haben.

Auf einem PC erlaubt das DOS nur Namen mit max. 8 Zeichen plus drei Zeichen für eine Namenserverlängerung (hier wählt SPRG immer .GRO), ein längerer Name wird daher vom Kermit des Computers nach dem 8. Zeichen abgeschnitten. Außerdem numeriert Kermit gleiche Namen durch, wer also zwei Verzeichnisse drucken läßt, die verschiedene Programme enthalten, braucht sich über die angehängten Ziffern auf dem PC nicht wundern.

```

PFEIFER WED 15.01.92 14:06:48
48SX SPRG
< ( ) + RCLF SWAP
002 # 38040300A0180FF2h
STOF 1 1 3 PICK
004 SIZE
START GETI
006 CASE DUP VTYPE
15 SAME
008 THEN EVAL 8
TVARS 15 TVARS +
010 IF DUP SIZE
THEN SPRG
ELSE DROP
END UPDIR
012
014 END DUP VTYPE
8 *
016 THEN DROP
END "48SX "
018 OVER + "PFEIFER "
DATE TIME TSTR + 2
→LIST OVER RCL DUP
" Bytes : " + SWAP
022 BYTES 10 CHR +
"Checksum : " +
024 SWAP + + 1001
DO DUP 2 MOD
026 NOT OVER " "
IFTE " " + ROT + 2
028 OVER 10 CHR POS DUP
CASE NOT
030 THEN DROP
OVER SIZE
032 END DUP
26 >
034 THEN DROP
OVER SIZE SUB 1 24
036 END 1 -
END SUB
038 LASTARG + OVER SIZE
SUB SWAP 4 ROLL +
040 ROT ROT SWAP 1 +
OVER SIZE
042 UNTIL NOT
END DROP2
044 LIST→ # 8Fh OVER 9
* R→B BLANK 0 R→B
046 DUP 2 →LIST 4 ROLL
1 →GROB REPL ( # 0h
048 # 7h ) 4 ROLL 3
→GROB REPL 2 ROT 1
050 -
FOR j # 0h j
052 9 * R→B 2 →LIST ROT
2 →GROB REPL
054 NEXT → g
< " .GRO" +
056 'g' SWAP 2 →LIST 1
→LIST SEND CLOSEIO
058 *
END
060 NEXT DROP2 STOF
» Bytes : 700,5
062 Checksum : # A9E1h

```

Hier hat sich SPRG selbst als GROB an den PC geschickt. Die TIFF-Datei wurde auf ca. 70 % verkleinert.

Schließlich haben PC und 48 nicht den gleichen Zeichensatz, um hier Ärger zu vermeiden, sollte das 48SX-Programm einen Namen haben, der nur die Zeichen 32127 aus dem ASCII-Satz enthält (Die Tabelle befindet sich in Teil 2, Anhang C des HP48 Handbuchs).

Auf der Seite des 48SX regelt SPRG alles weitgehend automatisch: Der ASCII-Modus und das Kabel als

Schnittstelle werden gewählt und der 48SX beginnt automatisch mit der Übertragung.

**Vorsicht:** Größen, die nicht von Flags gesteuert werden (z.B. Baud in der Liste IOPAR) verändert SPRG nicht.

Die Arbeit auf dem PC ist viel aufwendiger, daher ein Beispiel:

Das Programm SPRG soll sich selbst als GROB an den PC schicken. Dazu geben wir folgendes auf dem HP-48 ein:

```
'SPRG' ENTER SPRG
```

dann starten wir auf dem PC das Übertragungsprogramm mit K [ENTER] (die oben erstellte Batch-Datei).

Der Kermit-Befehl SERVER veranlaßt den PC das Ohr auf das Kabel zu legen um dem 48SX zu lauschen. Jetzt kann man eine Pause machen um ein paar Brötchen zu vertilgen, es dauert halt ein bißchen, vereinzelt piept der PC, aber das sollte kein Grund zur Beunruhigung sein. Wenn der 48SX fertig ist, kann man durch gleichzeitiges Drücken von Strg und C eine weitere Empfängerin des PC verhüten. Wer nur ein Programm übertragen möchte, kann statt SERVER auch RECEIVE oder kurz R eingeben, dann kommt es sofort nach der Übertragung zum temporären "Kermitus", dem Ableben der Verbindung zwischen 48SX und PC.

Egal ob man sich für R oder SERVER entschieden hat, verläßt man Kermit endgültig mit:

```
EXIT (ENTER)
```

ein. Wie man sich leicht mit dem Befehl DIR (diesen DOS-Befehl versteht auch Kermit) überzeugen kann ist auf dem PC eine neue Datei mit dem Namen SPRG.GRO entstanden (sofern der Name nicht bereits existiert hat).

Die weitere Bearbeitung der GROBs auf dem PC geht dann so weiter:

```
GROB2TIF.EXE < SPRG.GRO >
SPRG.TIF (ENTER)
```

Das Programm GROB2TIF.EXE braucht zwei verschiedene Namen, der eine (SPRG.GRO) ist die Quelle und SPRG.TIF das Ziel, in welchem die TIFF-Datei SPRG steht. Die Endung .TIF sollte man immer verwenden, damit sowohl Benutzer als auch Software diese Datei als brauchbare Grafikdatei erkennen. Das Verfahren

scheint zwar ein wenig umständlich, aber wer die Listings mit dem Infrarotdrucker erzeugt und nachträglich in den Text einklebt erhält trotz größerem Zeitaufwand eine schlechtere Qualität.

Hier allerdings noch ein paar Nachteile: GROB2TIF.EXE stellt TIFF-Dateien nur in einer Größe her. Wer also Listings verkleinern oder vergrößern möchte, braucht dafür also ein cleveres Text-, DTP oder Grafikprogramm. Außerdem erzeugt SPRG im 48SX sehr große Grafikobjekte, die ohne ausreichenden Platz zum Abbruch wegen Speichermangel führen, so braucht SPRG als GROB im 48SX ca. 10 kBytes, und zur einwandfreien Übertragung braucht man etwa den doppelte Platz. Damit SPRG im gesamten RAM wirken kann, sollte es im HOME-Verzeichnis stehen.

Der besondere Trick bei den Programmen PRP und SPRG war der, daß man ein Programm auf zwei Arten in einen String verwandelt werden kann: Entweder mit dem Befehl →STR oder indem man einen String addiert (hier ist es ein leerer String, d.h. ohne Zeichen, aber er kann beliebige Zeichen enthalten), also durch:

```
"" + (der String muß kein Zeichen
enthalten)
```

Bei Programmen bewirkt die letztere Methode die bekannte Darstellung durch PRP und SPRG. Die Funktion →STR hingegen erzeugt eine ganz andere Darstellung. Außerdem verwandelt diese Funktion alle Objekte in Strings (Strings, die mit →STR behandelt werden, bleiben unverändert). Hier eine kleine Tabelle der unterschiedlichen Wirkungen:

| Objekttyp             | "" +   | →STR  |
|-----------------------|--|---|
| Programm, Verzeichnis | Zeilenumbruch nach max. 19 Zeichen (ausgenommen Strings) | Zeilenumbruch nur vor strukturieren Befehlen (FOR, DO, END, STEP, THEN, «...) |
| 'Namen'               | werden ohne ' in String verwandelt                       | ' auch im String enthalten  |
| Vektoren              | Umwandlung mit Zeilenumbrüche n                          | Umwandlung ohne Zeilenumbrüche  |

Besonders reizvoll finde ich dabei die Umwandlung von Feldern (Arrays) in

Strings, da hier der 48SX die gewohnten Formen der Felder erhält, und sie nicht wie beim Ausdruck mit dem IR-Drucker zerpfückt.

```

PFEIFER WED 15.01.92 14:03:53
48SX TXTGROB
< { 6 9 11 } OVER
002 GET → n h
    * →STR 10 CHR + 0
004 { } ROT
    DO DUP 10 CHR
006 POS 1 SWAP 1 - SUB
    LASTARG + 1 + OVER
008 SIZE SUB 4 ROLLD
    DUP ROT + 4 ROLLD
010 SIZE MAX ROT ROT
    UNTIL DUP SIZE
012 NOT
    END DROP LIST→
014 DUP h * SWAP 2 +
    ROLL 6 * 1 + R→B
016 OVER 1 + R→B BLANK
    0 R→B SWAP 1 4 ROLL
018 FOR j ROT n
    →GROB ROT OVER SIZE
020 DROP > LASTARG IFTE
    ROT ROT # 1h j R→B
022 2 →LIST SWAP REPL h
    STEP 0 R→B DUP
024 2 →LIST ROT
    IF n 1 SAME
    THEN 3 PICK
026 SIZE SWAP DROP 2
028 →LIST SUB
    ELSE DROP2
030 END
    *
032 * Bytes : 373
    Checksum : # 113Dh
    
```

### Strings in Grobs

Das Programm TXTGROB verhält sich ähnlich wie die eingebaute Funktion →GROB und erwartet folgende Eingaben: Ein beliebiges Objekt in Ebene 2 und eine 1, 2 oder 3 in Ebene 1, um die Zeichengröße festzulegen. Im Unterschied zu →GROB verwandelt es mit →STR das Objekt in einen String (Strings bleiben unverändert) und bricht es an den Newline-Befehlen (10 CHR) um, außerdem ist die Zeichengröße 0 in Ebene 1 unzulässig. Das Programm baut den Text so ins GROB ein, daß der Rand rundherum ein Pixel breit ist. Für die (proportionale) Zeichengröße 1 reduziert das Programm anschließend noch einmal die Größe des GROBs.

Dieses Programm kann man natürlich auch verwenden, wenn man mit dem PC gar nichts zu tun haben will, sondern nur im LCD einen Text invers darstellen möchte, zum Beispiel folgenden Zweizeiler:

```
"Kriegt der Rechner einen Schock,
singt Heinz Schenk im Blauen Bock"
```

```
1 TXTGROB NEG →LCD
```

### Beliebige Objekte im PC

Ein Programm, welches beliebige Strings in grafische Objekte verwandelt, stellt **SOBJ** (Send OBJekt) dar. Es nimmt das Objekt, verwandelt es mit **TXTGROB** und gibt es dann an den PC weiter. Dabei ist die 2 in Zeile 4 von **SOBJ** dafür verantwortlich, daß ausschließlich die Zeichengröße 2 verwendet wird (kann man aber ändern).

Die von **SOBJ** erwartete Eingabe entspricht der des Befehls **SEND**. In Ebene 1 erwartet **SOBJ**:

1. Einen Namen. Das gespeicherte Objekt wird als **GROB** an den PC geschickt.

```

PFEIFER WED 15.01.92 14:05:21
48SX SOBJ
* { } + RCLF SWAP
002 # 38040300A0180FF2h
  STOF
004 * 2 TXTGROB → g
  * ".GRO" + 'g'
006 SWAP 2 →LIST 1
  →LIST
008 IFERR SEND
  THEN DROP
010 END CLOSEIO
  »
012 * → P
  * LIST→ 1 SWAP
014 START ( ) +
  LIST→
016 CASE DUP 2 >
  OVER NOT OR
018 THEN DROPN
  END 2 SAME
020 THEN SWAP
  IFERR RCL
022 THEN TEXT
  END P
024 EVAL
  END DUP
026 IFERR RCL
  THEN DROP2
028 ELSE P EVAL
  END
030 END
  NEXT
032 * STOF
  * Bytes : 260
034 Checksum : # 705Bh
    
```

2. Eine Liste mit Namen. Diese Liste arbeitet der **48SX** in umgekehrter Reihenfolge ab, und schickt die Variablen an den PC.

Statt den Namen in 1. und 2. darf man auch eine Liste eingeben, die an erster Stelle entweder den Namen oder (und das ist mehr als **SEND** kann) das Objekt enthält, und an zweiter Stelle den neuen Namen, unter dem es auf dem PC abgelegt werden soll. **SOBJ** ignoriert alle unzulässigen Eingaben.

Beispiel: Mit '**SOBJ**' **SOBJ** schickt sich das Programm selbst an den PC. Mit

```
{ ABC SOBJ }
```

landen **ABC** und **SOBJ** auf dem PC, mit

```
{{ (12) XY } DEF }
```

bringt man den Vektor unter dem Namen **XY** auf den PC und die Variable **DEF** wird unter ihrem eigenen Namen verschickt.

```
{{ SPRG UVW } }
```

verschickt **SPRG** als **UVW**, aber Vorsicht: Hier muß doppelt **ge{}** werden, sonst schickt **SOBJ** sowohl **SPRG** als auch **UVW** (sofern vorhanden) als einzelne Objekte an den PC. Alles klar?

Um die Dateien auf dem PC besser zu identifizieren, hängt **SOBJ** immer ein **.GRO** als Dateinamenserweiterung an!

Um den PC empfangsbereit zu machen, und um nach der Übertragung die **GROBs** in **TIFF**-Dateien zu verwandeln, verfährt man wie bei **SPRG**.

Im Display gönnt der **48SX** den Zeichen nur 8 Pixel Höhe. Das reicht allerdings nicht, wenn die Unterlängen der Kleinbuchstaben mit den Groß-

```

[[ ,006112 ,01 206,2 ]
[ ,01 6,9828 129,2 ]
[ ,015 13,036 87,98 ]
[ ,02 17,513 67,01 ]
[ ,03 24,1 45,67 ]
[ ,04 28,983 34,8 ]
[ ,05 32,898 28,19 ]
[ ,06 36,183 23,74 ]
[ ,08 41,534 18,1 ]
[ ,1 45,833 14,67 ]]
    
```

Hier hat **SOBJ** die Dampftafel aus **Pas de deux 6 (PRISMA 2/3-91)** verwandelt. Die Abbildung ist auf ca. 70% verkleinert.

buchstaben der nächsten Zeile verschmelzen, weshalb alle Programme hier pro Zeile 9 Pixel vergeben um die Lesbarkeit verbessern.

Außerdem kann der **SEND**-Befehl eine Anweisung

der Form:

```
{{ GROB ABC } } SEND
```

nicht verarbeiten, denn in diesen Listen sind nur 'Namen' von Variablen erlaubt (eine sinnlose Einschränkung), weshalb sich meine Program-

me mit einem kleinen Trick aus der Klemme helfen:

das **GROB** wird für die Dauer der Übertragung in einer lokalen Variablen zwischengespeichert (**GROB** in Ebene 1, sein Name in Ebene 2):

```
→g « ".GRO" + 'g' SWAP
2 →LIST 1 →LIST SEND »
```

Hier noch ein anderes Anwendungsbeispiel für **GROB2TIF.EXE**: Mit der Funktion **ASN** kann man die Funktion **LCD→** auf jede beliebige Taste zuweisen. So ist es möglich, jede Darstellung im Stack sofort als **GROB** zu sichern. Wenn ein Programm eine neue Menüzeile mit **TMENU** anlegt, kann man so das **LCD→** kopieren, mit **SEND** an den PC übertragen und als **TIFF**-Datei in eine Textverarbeitung übernehmen.

### Bibliotheken

Durch den 2. Teil des Handbuches geistern seltsame Objekte, die sog. Bibliotheken. Mit deren Hilfe kann man beliebige Variable (Programme, Konstanten u.s.w.) zu einem kleinen Paket schnüren, welches mit folgenden Eigenarten aufwartet:

1. Die Variablen (z.B. Programme) lassen sich nicht verändern und nicht einsehen.
2. Die Variablen lassen sich nicht trennen, wichtige Teile kann man nicht herauslösen.
3. Variable können unsichtbar werden, so daß diese nur noch von den Programmen dieser Bibliothek genutzt werden können.
4. Autorenhinweis, Versionsnummer und unverbindliche Preisempfehlung können so ebenfalls dauerhaft eingebaut werden.

Schneller sind solche Bibliotheken allerdings kaum und auf dem **48er** lassen sich diese leider nicht erzeugen. Notwendige Voraussetzung ist nicht nur das Kabel und Kermit, sondern auch ein weiteres Programm für den PC, welches nicht auf dem normalen Vertriebsweg erhältlich ist. Es heißt **USRLIB.EXE** (Version 2.2 liegt mir vor, 2.3 soll es geben), läuft auf dem PC und verwandelt dort ein normales Verzeichnis in eine Bibliothek. Das Programm stammt von HP, außerdem ist seine Nutzung und Weitergabe kostenlos, solange man damit kein Geld verdient, allerdings gibt es außer der

Bedienungsanleitung keine weitere Unterstützung. Ich versuche dieses Programm über die MS-DOS-Gruppe in Umlauf zu bringen (s.u.), man bekommt es allerdings auch über die HP-Mailbox in den USA.

Der Ablauf ist recht einfach:

0. Bevor etwas in die Hose geht, erst 'mal alles im Rechner auf dem PC in Sicherheit bringen. Um alles unter dem Namen HP48 zu speichern wirft man auf dem PC Kermit an und befiehlt dem 48er:

```
:!O: "HP48.BAK" ARCHIVE
```

Die Endung .BAK (=Back-up) hindert die Editierfunktionen mancher PC-Helferlein an dieser Versicherungskopie herumzudoktern.

1. Ein Verzeichnis im RAM zusammenstellen, welches alle Variablen enthält, die später in der Bibliothek erscheinen sollen. Da allerdings eine Bibliothek keine Verzeichnisstruktur enthalten kann, sollte man in dem Verzeichnis keine weiteren Unterverzeichnisse anlegen.
2. Austesten, ob alles funktioniert und die Variablen in der richtigen Reihenfolge anordnen.
3. Jetzt erzeugt man in dem Verzeichnis mit den eigenen Variablen (meist Programme) einige weitere Variable, denen eine besondere Bedeutung zukommt, und die alle mit dem Dollarzeichen beginnen. Unbedingt erforderlich sind die unter a und b genannten:
  - a. Die Bibliothek braucht eine Nummer. Dabei hat man die Auswahl von 02047 (#0#7FFh). Allerdings sollte man sich auf den Bereich von 769 bis 1792 beschränken, und hier wiederum den Bereich von 1537 bis 1792 bevorzugen, da alle anderen Bereiche reserviert oder bereits belegt wurden. Diese Nummer speichert man (dezimal oder binär) als
 

```
nnnn '$ROMID' STO
```
  - b. Die Bibliothek braucht einen Namen. Bis zu fünf Zeichen erscheinen später dann auf der Menütaste, bis zu 22 Zeichen lassen sich mit REVIEW erkennen (Space erlaubt).

Dazu speichert man einen String:

```
"Name" '$TITLE'
```

- c. Möglich ist auch ein Programm, welches die Bibliothek beim ersten Laden z.B. einbindet. Es darf aber in keinem Falle zu einem Error führen und es darf nichts vom Stack nehmen (da das Programm nach einem Systemhalt startet, ist da auch nichts). Das Programm speichert man dann mit
 

```
« ... » '$CONFIG' STO
```

Die Variable '\$CONFIG' muß man als versteckte Variable angeben, sonst erscheint sie später in der Bibliothek.

- d. Etwas undurchsichtig finde ich die Aufgabe der Variablen '\$MESSAGE'. Zunächst soll man in verschiedene Variable Strings speichern. Dann nimmt man eine Liste mit den Namen dieser Variablen und speichert sie in \$MESSAGE, damit daraus später eine Meldungstabelle werden kann. Es geht aber auch ohne.
- e. Man kann Variable in der Bibliothek verstecken, z.B. Unterprogramme, die für sich alleine ziemlich sinnlos bleiben, die aber von mehreren Programmen der Bibliothek gebraucht werden. Dazu kann man die Variablen \$VISIBLE (für die Sichtbaren) und \$HIDDEN (für die Unsichtbaren) verwenden. Falls beide vorliegen, beachtet der PC nur \$HIDDEN, die man wegen \$CONFIG ohnehin anlegen sollte, Variable die nicht in \$HIDDEN auftauchen bleiben sichtbar, so daß \$VISIBLE eigentlich überflüssig ist. Um Variable zu verstecken, sammelt man deren Namen in einer Liste und speichert sie:
 

```
{ ... } '$HIDDEN' STO
```
- f. Möglicherweise soll die Bibliothek veränderliche Daten enthalten. Nehmen wir z.B. eine Sammlung von Steuerberatungsprogrammen. Die darin enthaltenen Steuersätze, Tabellen u.s.w. haben in der Regel nur ein begrenztes Halt-

barkeitsdatum. Man kann nun das Zahlenmaterial ebenfalls in die Bibliothek übernehmen. Sobald sich die Bibliothek installiert, legt sie aber die Zahlen im RAM als Variable an. Die internen Programme greifen ebenfalls auf das RAM zu, weil dort die Daten regelmäßig aktualisiert werden können. Die Liste mit den Namen der Variablen speichert man mit:

```
{ ... } '$VARS' STO
```

4. Nun schaltet man den 48er auf binäre Übertragung und schickt das Verzeichnis mit Kermit an den PC.
5. Auf dem PC startet man das Programm USRLIB.EXE mit den Namen des Verzeichnisses und einem Namen, der das Ergebnis (also die Bibliothek) aufnehmen kann:

```
USRLIB STEUER TEUER
```

Man kann an USRLIB noch einige Parameter anhängen, die werden aber durch die unter Punkt 3 angelegten Variablen abgedeckt. Die Namen von Quellverzeichnis und Zielbibliothek sollten verschieden sein, danach kann man TEUER mit Kermit an den 48er zurückschicken Fertig!

HP weist auf einige Schwierigkeiten mit Bibliotheken hin, so kann z.B.

```
'A' 5 GETI
```

nicht ausgeführt werden (wichtig, wenn eine Variable in der Bibliothek eine Liste enthält, die von anderen Programmen genutzt werden soll). Eine Abhilfe ist nicht nötig, denn wer so programmiert, braucht Bibliotheken um seine Programme vor der spöttelnden Weltöffentlichkeit zu verstecken. Als Abhilfe funktioniert nämlich

```
A 5 GETI
```

viel besser, schneller und kürzer.

Da sich Objekte in Bibliotheken wie Programme im ROM verhalten, kann man natürlich Befehle wie STO und PUT vergessen, wenn sie etwas in der fertigen Bibliothek verändern sollen.

Auch wenn die Programme vorher einwandfrei liefen kann es sein, daß die von USRLIB.EXE erzeugte Bibliothek nicht richtig geht. HP empfiehlt

daher vollständiges Austesten der fertigen Bibliothek und lehnt natürlich jede Garantie und jede Unterstützung für dieses kostenlose Programm ab (Ratings anrufen nützt also nichts).

Am Besten machen wir dazu ein kleines Beispiel:

Die hier vorgestellten Programme SPRG, SOBJ und TXTGROB sollen in eine Bibliothek. Dazu rufen wir zunächst die drei Programme in den Stack zurück:

```
'SPRG' RCL LASTARG
```

```
'SOBJ' RCL LASTARG
```

```
'TXTGROB' RCL LASTARG
```

Dann erstellen wir auf dem 48SX ein neues Verzeichnis z.B. mit dem Namen LIB:

```
'LIB' CRDIR
```

und wechseln hinein, indem wir die Menütaste A (nach CRDIR steht da LIB drauf) drücken. Jetzt die drei Programme speichern:

```
STO STO STO
```

Einen Autoren- und Versionshinweis bauen wir dann noch mit folgendem Programm ein:

```
« "Version 1.0 vom 2.02.92
```

```
Von Harry Hacker.
```

```
Dieses Programm demoliert Ihren Rechner. Schade, daß Sie es zu spät bemerkten."
```

```
1 DISP 3 FREEZE »
```

```
'VERSION' STO
```

Natürlich kann man hier auch seriöse Kommentare einbauen.

Als Nummer geben wir der Bibliothek eine 999, also:

```
999 ' $ROMID' STO
```

und außerdem den Titel mit

```
"GROBs in TIFF-Dateien" ' $TITLE'
```

Da wir keine der vier Dateien verstecken wollen, brauchen wir weder eine Datei \$HIDDEN noch die Datei \$VISIBLE anlegen. Es reicht vollkommen aus, dieses Verzeichnis mit diesen sechs Variablen via Kermit an den PC zu schicken:

```
K (ENTER) R (ENTER)
```

und im I/O SETUP Menü des 48ers wählt man die Übertragungsart binary und führt dann

```
UPDIR 'LIB' SENDOBJ
```

aus. Jetzt beginnt man auf dem PC die Umwandlung mit

```
PUSH (ENTER)
```

wobei der Befehl PUSH es ermöglicht, den folgenden

```
USRLIB LIB PCGROB (ENTER)
```

Befehl auszuführen, ohne Kermit zu beenden und

```
EXIT (ENTER)
```

ins Kermit zurückkehrt. Die neue Datei PCGROB schickt man jetzt an den 48SX zurück:

```
SEND PCGROB (ENTER)
```

Auf dem HP-48SX speichert man diese zunächst in PORT 0 und bindet sie dann ein:

```
PCGROB RCL :0:PCGROB STO  
:0:999 ATTACH OFF ON
```

Nach dem Einschalten beginnt die Anzeige kurz zu flimmern (harmlos) außerdem bewirkt ein Systemhalt, daß die Inhalte von Stack und den Last-Funktionen gelöscht werden. Vorsicht: Die Menütaste GROB ist im LIBRARY-Menü nur dann sichtbar, wenn man sich in dem Verzeichnis befindet, in welchem man ATTACH ausgeführt hat (oder dessen Unterverzeichnissen). Die Bibliotheken werden nämlich wie ein normales Programm an ein Verzeichnis angehängt, und sind nur in diesem oder dessen Unterverzeichnissen verwendbar. Weitere Einschränkung: Nur das HOME-Verzeichnis kann mehr als eine Bibliothek anbinden! Zum Schluß noch ein paar Aufräumarbeiten: Sowohl 'LIB' als auch PCGROB kann man aus dem VARVerzeichnis löschen:

```
'LIB' PGDIR 'PCGROB' PURGE
```

Um ein Objekt in Port 0 wieder loszuwerden, wechselt man in LIBRARY PORT0, drückt ENTRY (auf der aTaste), die Menütaste des zu löschenden Objekts und dann PURGE. Und nun viel Spaß beim Testen.

## Datenkompression auf dem PC

Für die Übermittlung von HP48 Programmen kann man dank Kermit normale Disketten verwenden. Allerdings gibt es einen kleinen organisatorischen Nachteil: Die Programme fliegen wild in der Gegend herum, sie sind nicht mehr strukturiert.

Dagegen gibt es folgende Möglichkeiten der Abhilfe:

1. Statt nur einzelne Programme zum PC zu senden, faßt man diese bereits auf dem HP-48 (teilweise) in Verzeichnissen (soweit logisch passend) zusammen. Im Gegensatz zu Verzeichnissen für die Erstellung von Bibliotheken darf man hier auch Unterverzeichnisse anlegen. Außerdem kann man andere Variable, die man normalerweise nicht überträgt, ebenfalls einbauen, so daß ein sinnvolles Paket entsteht. Auf diese Art habe ich z.B. meine Trigonometrieprogramme aus Pas de deux 1 samt Variablen zusammengefaßt.
2. Man nimmt eines der üblichen Datenkompressionsprogramme für den PC und faßt die HP48 Programme und erklärende Texte in einer Datei zusammen.

Die letztere Methode hat folgende Vorteile:

1. Auf dem PC bleibt nur eine Datei zurück, die aber nach dem Auspacken (Dekomprimieren, Expandieren) alle 48 Programme, Verzeichnisse und Textdateien wiederherstellt.
2. Bei der Weitergabe gehen keine HP-48 Programme verloren, weil nur eine komprimierte Datei übertragen wird.
3. Komprimierte Programme sparen Platz.

Der gewonnene Speicher hat zwei Quellen: Zum einen verdichten die Kompressionsprogramme wirklich, weil sie sich wiederholende Zeichenfolgen erkennen und ersetzen. Zum anderen verbessert sich die Speicherverwaltung: Auf einer Festplatte ist der Speicher in Sektoren (üblicherweise 1024 Bytes lang) aufgeteilt. Hat man zwei Programme mit 500 und 1500 Bytes Länge, dann würden diese in zwei Sektoren passen, aber die Speicherverwaltung erlaubt nur eine Datei pro Sektor. Das Programm mit den 1500 Bytes bekommt also zwei Sektoren, das Programm mit den 500 Bytes kann aber nicht angehängt werden und bekommt deshalb einen eigenen, neuen Sektor. Ein Kompressionsprogramm macht dagegen aus den beiden Programmen eine Datei und die kann dann in zwei, bei guter Kom-

pression sogar in einen Sektor geschrieben werden !

An dieser Stelle möchte ich das Programm LHA.EXE vorstellen. Es ist für nichtgewerbliche Zwecke frei einsetzbar und von seinen vielen Möglichkeiten kann es auch selbstexpandierende Dateien schreiben, d.h. man faßt zunächst alle 48er Programme zusammen und macht daraus eine .EXE Datei (EXE kommt von EXECution nicht von Expand), die beim Aufruf automatisch alle Programme auspackt. Der Vorteil liegt darin, daß der Empfänger des Programms kein Expansionsprogramm braucht und die Suche nach dem richtigen Expansionsprogramm/Version spart !

Selbstexpandierende Programme brauchen etwas mehr Platz aber das fällt nur bei vielen komprimierten Dateien auf dem selben Datenträger ins Gewicht. Andererseits kann man durch die Weitergabe des Kompressionsprogramms auch dem Empfänger eine Freude bereiten, weil der dann auch seine Dateien besser verschicken kann.

Hier der wichtigste Teil der Bedienungsanleitung für LHA.EXE. Dabei

sollte LHA.EXE im gleichen Verzeichnis wie KERMIT.EXE stehen (bei mir wäre das im Verzeichnis HP).

1. In das Verzeichnis von KERMIT und LHA wechseln, z.B.  
CD \HP [ENTER]
2. Auf dem PC ein neues Verzeichnis einrichten und wechseln z.B.  
MD PROGRAMM (ENTER)  
CD PROGRAMM (ENTER)
3. Kermit starten  
K (ENTER) SER (ENTER)
4. Auf dem HP48 alle Variablen mit SEND an den PC schicken. Dabei ist es egal, ob der ASCII oder binäre Übertragungsmodus gewählt wurde. Der Erfolg des Packprogramms ist normalerweise auch vom Modus abhängig (aber wie ?).
5. Nun kann man in das Verzeichnis PROGRAMM noch Texte kopieren, wenn man z.B. eine LIESMICH.TXT oder READ.ME Datei einbauen will (wegen Autorenschrift, Erklärungen o.ä.)
6. LHA starten:  
CD.. (ENTER)

LHA A MEIERZWO PROGRAMM\ (ENTER)

Jetzt erzeugt LHA eine Datei, die alle Dateien im Verzeichnis PROGRAMM zusammenpackt und als MEIERZWO.LZH ausgibt. Um daraus eine selbstexpandierende Datei zu machen muß man noch einmal

LHA S MEIERZWO (ENTER)

ausführen. Dann erscheint neben MEIERZWO.LZH auch noch die Datei MEIERZWO.EXE auf dem PC.

Um Dateien mit der Endung .LZH auspacken, braucht man ebenfalls LHA.EXE. Dazu gibt man

LHA X MEIERZWO

ein. LHA erkennt automatisch die MEIERZWO-Datei mit der richtigen Endung .LZH.

Genaueres über LHA.EXE steht in dessen Bedienungsanleitung. Soweit mir bekannt, wurden diese Programme bereits von der MS DOS-Gruppe verteilt.

### Programmaustausch

Damit jedes CCD-Mitglied die Möglichkeit hat, mit USRLIB, LHA und meinen bisher in PRISMA veröffentlichten Programmen zu arbeiten, habe ich über den Diskettendienst der MS-DOS-Gruppe eine Datei mit dem Namen PFEIFER1.EXE weitergegeben. Sie enthält die aktuellsten Versionen meiner Programme und packt diese beim Start automatisch aus. Einige Programme habe ich in Verzeichnissen zusammengefaßt, genaueres kann man aber nach dem Auspacken mit

TYPE LIESMICH.TXT | MORE

erfahren. In einigen Programmen (auch den hier vorgestellten) kann man die dort (aus Gründen der Lesbarkeit) vorkommenden 10 CHR immer durch einen String mit dem Newline-Befehl (Zeilenschaltung, blaue Vortaste und Dezimalpunkt) ersetzen.

Hinweis: PC ist höchstwahrscheinlich ein eingetragenes Warenzeichen der International Business Machines Corp. (IBM). Hewlett-Packard ist möglicherweise der Hersteller des HP-48SX.

Ralf Pfeifer (116), Rubensstr. 5,  
5000 Köln 50

### Amerkung der Redaktion:

Bestimmte Namen haben für DOS eine festgelegte Bedeutung. Dies sind sogenannte DOS-Einheitennamen. Dabei handelt es sich um reservierte Namen, die als Dateinamen nicht verwendet werden dürfen.

| Reservierter Name | Einheit  |
|-------------------|--|
| CON               | Konsoltastatur/-bildschirm   |
| AUX oder COM1     | Erster serieller Anschluß  |
| COM2              | Zweiter serieller Anschluß   |
| COM3              | Dritter serieller Anschluß   |
| COM4              | Vierter serieller Anschluß   |
| LPT1 oder PRN     | Erster Paralleldrucker   |
| LPT2              | Zweiter Paralleldrucker  |
| LPT3              | Dritter Paralleldrucker  |
| NUL               | Nicht existierende Einheit zum Testen von Anwendungen. Ist NUL die Ausgabeinheit, werden die Schreiboperationen nur simuliert. |

Der von Ralf Pfeifer geschilderte Effekt ist also durch die Ausgabe zur Konsole entstanden. dw

# Suche nach Objekt-Typen

von Michael Ritter

Diese Routine sucht in einem vorgegebenen Objekt nach Element-Typen, z.B. nach globalen Variablen.

## Die Routine SEARCH

Diese Routine erzeugt aus einem vorgegebenen Objekt und einer vorgegebenen Objekt-Typ-Nummer eine Liste, in der alle Elemente, die der angegebenen Nummer entsprechen, enthalten sind.

Als Vorgabe sind möglich: Programm, Liste oder algebraisches Objekt.

Ist das Argument für die Routine keines der zulässigen Objekte, oder enthält das Argument keine Elemente des angegebenen Typs, wird eine leere Liste ausgegeben.

Die Zerlegung des vorgegebenen Objektes erfolgt wie bei der Routine →UPN.

Stößt die Routine in dem zerlegten Objekt auf ein Programm, eine Liste oder ein algebraisches Objekt, so werden diese Objekte ebenfalls zerlegt und die Bearbeitungsschleife um die Anzahl der hinzugekommenen Elemente erweitert (WHILE Schleife).

Mit 'IF t s SAME ...' werden die gewünschten Elemente herausgefiltert und in die Liste aufgenommen.

'IF POS 0 SAME ...' vermeidet doppelte Aufnahmen in die Ergebnisliste.

Ein Beispiel dazu:

```

                                1USR
{ HOME }
4:
3:
2:  * A X + SIN C / *
1:                               6
SEARCH

```

In dem Programm soll nach globalen Variablen (Typ: 6) gesucht werden.

Mit SEARCH ergibt sich:

```

                                1USR
{ HOME }
4:
3:
2:
1:                               { A X C }
SEARCH

```

Zur Verwendung der Objekttypen-Nummern (Benutzerhandbuch I, Seite 102), bleibt noch folgendes anzumerken.

Die Suche nach Programmen und Listen ergibt eine leere Liste, da diese Objekte aufgelöst werden und deshalb nicht mehr gefunden werden. Da algebraische Objekte ebenfalls zerlegt werden, liefert die Suche danach keine algebraischen Objekte, sondern Userdefinierte-Funktionen.

Indizierte Variablen werden wie Userdefinierte-Funktionen behandelt. Die Reihenfolge der Elemente in der Ergebnisliste sagt nichts über die Reihenfolge im Ausgangsobjekt aus.

Dazu ein Beispiel:

```

                                1USR
{ HOME }
4:
3:
2: 'F(23;5)+3'
1:                               9
SEARCH

```

ergibt:

```

                                1USR
{ HOME }
4:
3:
2:
1:                               { 'F' }
SEARCH

```

Die Verwendung von Objekttypen-Nummern zwischen 0 und 17 stellt in der Regel kein Problem dar, die Suche nach eingebauten Funktionen und Befehlen ist ebenfalls möglich.

Eine Suche nach Systemobjekten, z.B. System Binary Typ 20 ergibt auf die Userdefinierte-Funktion des Beispiels angewendet, eine Liste in der Form { <2h> }, an der man prima erkennen kann, daß die Funktion zwei Argumente benötigt.

Ebenso prima befördert ein EVAL hinter dieser Liste ein "Try to recover Memory?" zutage.

Die System-Objekt-Typen zwischen 20 und 31 sind also mit erhöhter Vorsicht anzuwenden.

# AOS → UPN

## Konvertierung von algebraischem Objekt zur UPN-Programmzeile

von Michael Ritter

```
SEARCH
* SWAP DUP TYPE ROT
( ) → t s e
*
CASE DUP ( ) ≠
t 5 SAME AND t 8
SAME OR t 9 SAME OR
THEN # 1C973h
SYSEVAL 1 SWAP
FOR i DUP
TYPE 't' STO
WHILE DUP
QUOTE TYPE 9 SAME
OVER ( ) ≠ t 5 SAME
AND OR t 8 SAME OR
REPEAT
# 1C973h SYSEVAL 1
- NEG 'i' STO+ DUP
TYPE 't' STO
END
IF t s
SAME
THEN e
OVER
IF POS
0 SAME
THEN
'e' STO+
ELSE
DROP
END
ELSE DROP
END
NEXT e
END t s SAME
THEN ( ) +
END DROP e
END
*
Checksum: # 908Dh
Bytes : 386
```

Michael Ritter  
Fasanenstr. 8  
5060 Bergisch Gladbach 1

Ich möchte in im Folgenden eine Routine vorstellen, die auf die Veröffentlichung der Syseval Adressen zurück geht.

Diese Routine wandelt algebraische Objekte in eine UPN-Programmzeile um.

Es wird die SYSEVAL-Adresse (#1C973h) verwendet.

Die Systemroutine, die sich hinter dieser Adresse verbirgt, zerlegt Objekte in einzelne Elemente. So z.B. algebraische Objekte in einzelne Funktionen und Variablen. Ein Aufruf der Adresse ohne Argument im Stack hinterläßt ca. 4000 ! Externals im Stack. Das Ausführen dieser Externals bedeutet in der Regel den Speicherverlust.

### Die Routine →UPN

Diese Routine konvertiert algebraische Objekte in eine UPN-Programmzeile.

Nach Ausführen des Sysevals liegt die UPN-Programmzeile eigentlich schon vor. In Stack 1: ist das letzte Element der UPN-Zeile enthalten, in Stack 2: das Vorletzte usw.

Diese Elemente werden in einem String gesammelt und mit den Programmbegrenzern versehen.

Die Abfrage 'IF TYPE 19 > ...' vermeidet, daß Systemobjekte in den String gelangen können. Der Befehl 'STR→' erzeugt das Programmobjekt.

Ein Beispiel zur Anwendung:

```
RAD      1USR
{ HOME } 08.02.92 14:16:50
4:
3:
2:
1:      'SIN(A+X)/C'
→UPN
```

wird mit →UPN

```
RAD      1USR
{ HOME } 08.02.92 14:17:10
4:
3:
2:
1:      * A X + SIN C / *
→UPN
```

Konvertiert man algebraische Objekte, die z.B. Integrale oder Differentiale enthalten, werden die Argumente für diese Funktionen nicht zerlegt, da standardmäßig algebraische Objekte benötigt werden.

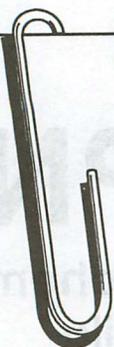
Algebraische Objekte, die indizierte Variablen enthalten, werden nicht richtig umgewandelt, da die Indizierung als Argument für eine Userdefinierte-Funktion angesehen wird.

Sollten indizierte Variablen berücksichtigt werden, müßte das Programm anhand der im Verzeichnis vorhandenen Variablen entscheiden, ob indizierte Variablen vorliegen und in diesem Fall den Befehl GET einfügen. Da dies durch alte Variablen leicht zu unbeabsichtigten Ergebnissen führt, wurde darauf verzichtet.

```
→UPN
* NEWOB DUP TYPE
IF 9 SAME
THEN "←" SWAP
# 1C973h SYSEVAL
"→" 1 ROT
START SWAP DUP
IF TYPE 19 >
THEN DROP
ELSE QUOTE
SWAP + " " SWAP +
END
NEXT + STR→
END
*
```

```
Checksum: # 12D0h
Bytes : 136,5
```

Michael Ritter



# Neues aus Corvallis USA

## Neuer 48LX, 28E, IR-Drucker und Software

Äußerlich kaum verändert geht der HP-48SX in's zweite Jahr. Dennoch hat HP den Preis (jetzt 698 US\$) und das Innere gründlich renoviert.

Ziel ist es, die Möglichkeiten der alten "Saturn"-CPU zu steigern, und zugleich möglichst viele periphere Komponenten des HP-95LX zu nutzen, um die Produktionskosten zu senken.

Zunächst fällt die normgerechte RS 232 Schnittstelle auf und das neue LCD.

Statt 131x64 Pixel hat letzteres jetzt wie der 95LX 240x128 Pixel (aber in etwas kleineren Abmessungen), einen Grafikprozessor mit eigenem RAM (GGA=Gray Graphics Adapter, graues VGA), sowie ein aktives Display mit besserem Kontrast (es ist nur schwarz/wei, funktioniert aber wie Farb-LCDs) die den Rechner schon beim Einschalten schneller erscheinen lassen.

Neue Wege beschritt man auch bei der CPU:

Während man bisher durch vergleichsweise niedrige Integrationsdichte den Chip mit geringen Ausfallraten fertigen konnte, ermöglicht ein neues Verfahren die getrennte Herstellung von CPU, RAM, ROM und Treiberbausteinen (Chipset), und deren anschließende Verbindung durch Elektronenstrahlschweißen zu einem Chip bei fast gleicher Ausschußrate.

HP verdoppelte die Taktfrequenz

auf 8 MHz, der Prozessor schafft statt 4 Bit Wortbreite jetzt 20 Bit, verfügt über einen zusätzlichen internen 32K großen Cache und bekommt serienmäßig 512 kB RAM.

Fast alle höheren Funktionen führt die neue CPU als Maschinenbefehl in einem Takt durch, so daß der "Saturn" auch Coprozessor-Qualitäten hat, und insbesondere komplexe Zahlen schneller bearbeitet.

Zu der bereits erwähnten RS 232 Schnittstelle kommt auch ein schnellerer IR-Port und mit 44,3 kBit/s (voll Duplex) arbeitet, außerdem kann man mit dem neuen DEVICE-Befehl verschiedene Geräte (den neuen Drucker oder das neue Laufwerk) selektieren.

Um die Signalübertragung allerdings zuverlässig zu gestalten, mußten Laserdioden aus den CD-Playern her, deren kohärentes Licht besonders zur Signalübertragung geeignet ist.

Durch eine Facettenoptik (hier stand die Stubenfliege Pate) streut man das normalerweise gebündelte Laserlicht, damit die Peripheriegeräte nicht auf einer optischen Bank ausgerichtet werden müssen.

Ein Barcodeplotter gehört inzwischen ebenfalls zur Grundausstattung, das Lesen von Barcodes wird indes ein teurer Spaß, der mit der verbesserten IR-Diode wie bei den Scannerkassen im Supermarkt abläuft, aber ein spezielles Treibermodul (150 US\$, ab Herbst) erfordert.

Überarbeitet hat HP auch einige Algorithmen. Der Nullstellensucher (SOLVE) kann endlich auch komplexe Nullstellen finden, ein neuer interaktiver Grafikeditor hilft den GROBs auf die Sprünge, und eine Reihe von Funktionen können mehr:

Die von Prof. Stephen Hawking bewiesene Existenz der "Funktional-deterministischen Kongruenzen" setzte HP erstmalig zum ALGEBRAXAS Verfahren um, so da die symbolische Integration, COLCT und EXPAN praktisch jede Umformung entweder durchführen oder deren Unmöglichkeit feststellen können (sofern die Batterieerwartung das noch zuläßt).

### Neuer HP28E

Seit dem Erscheinen des 48SX und des kleineren 48S (der entfällt bald wieder) hat der 28S in der Käufergunst und im Preis nachgelassen.

Inzwischen produziert HP deshalb an der Rentabilitätsgrenze, weshalb die Entscheidung fiel, eine Sparversion des 48LX als 28E (E=Economy, 349 US\$) anzubieten:

Äußerlich bis auf das größere LCD und die IR-Dioden ein 28S, aber sonst baugleich mit dem 48LX. Auf Module, RS 232 und die meisten I/O-Funktionen muß der 28E verzichten.

Auch hier spart HP ganz schön: Der 28E läuft nur mit halber Geschwindigkeit, und das RAM hat nur 128 KB, denn für ihn werden



bevorzugt Komponenten zweiter Wahl aus der 48LX Produktion eingesetzt, z.B. wenn ein Teil eines 512 KB RAMs nicht funktioniert.

### Neuer HP95LX ?

Dank ihrer guten Kontakte zu gesprächigen HP-Mitarbeitern, veröffentlichte der (US-)amerikanische HP-ComputerClub bereits Daten zu einem noch geheimen Projekt, dem neuen 95AT, und die Fakten scheinen es zu bestätigen.

Die Verkaufszahlen des 95LX überraschten selbst HP, denn besonders Versicherungen statten ihre Außendienstmitarbeiter mit dem 95LX aus, HP ist auf dem besten Wege, wie beim DeskJet und beim LaserJet zum Marktführer zu werden.



ATARI, mit dem Portfolio (ebenfalls mit 8088 Prozessor) länger und erfolgloser im Geschäft, versucht sich an diesem Kuchen doch noch ein Stück abzuschneiden, weshalb auch die Werbung auf den 95LX eindringt (in den USA ist vergleichende Werbung erlaubt) und im neuen Portfolio ein Softwarepaket anbietet, welches den 95LX übertrifft (z.B. mit WINWORD).

Als Gegenschlag - so der Anwenderclub - kommt der 95AT Ende des Jahres mit einem von NEC entwickelten Prozessor (soll V2 heißen und kombiniert 80386SX und den mathematischen Coprozessor 80387SX), 2 MB RAM und eine "CMOS Harddisk" mit 4 MB, welche mit max. 2 weiteren Modulen auf 12 MB aufgestockt oder entsprechend partitioniert werden kann.

Die bisherigen Softwarepakete ergänzt man um ELLBOWS 1.0 (eine an WINDOWS angelehnte grafische Benutzerführung), einige Funktionen aus dem Norton Commander, der schon bisher verfügbare HP 19B II erhält im 95AT zusätzliche Funktionen und Programmiermöglichkeiten, aber ob eine Textverarbeitung auf diesen winzigen Tastaturen dem Gelben vom Ei die Farbe reichen kann, darf bezweifelt werden, auch wenn Lotus Manuscript 2.1 für den 95AT unvermeidlich scheint.

Einen Faux-pas soll das neuen Modell ebenfalls ausbügeln:

Von den drei Druckertreibern des aktuellen 95LX unterstützt nur einer einen HP-Drucker, der neue soll aber mit allen HP's (auch dem DIN A0 Plotter) und den EPSON LQ-Modellen sprechen.

### ThermoJet IR-Drucker

Am IR-Drucker will HP festhalten. Der neue Drucker für alle Taschenrechner und den 95LX bekommt auch eine Sendediode, so daß er den Rechnern den Fortgang seiner Arbeit mitteilen kann.

Außerdem soll ein größerer Druckpuffer (8-KB statt bisher 200 Zeichen) den Rechner entlasten.

Aber die richtige Speed-Performance kommt aus dem neuen Druckverfahren:

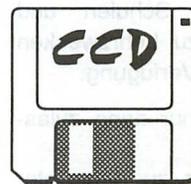
Das Thermopapier läuft zwischen zwei Walzen, eine dritte wirkt als Druckkopf, die entgegen der Papiervorschubrichtung dreht und schräg (!) zum Papier eingebaut ist

(das Prinzip entspricht der Kopftrommel beim VHS-Videorecorder) und alle 240 Pixel einer Zeile zugleich druckt.

Eine zusätzliche Bürste verhindert einen verschmutzten Druckkopf. In NLQ entspricht das Ergebnis sogar einem 24-Nadel-Drucker. Auch in der Geräuschkulisse hat der Drucker zugelegt, statt einem morschen "quietsch-rascht" ertönt neuerdings das mürbe DeskJet-Schlurfen. Für 299 US\$ ist das ein bißchen zuviel.

### Windows & 48LX

Zum Schluß noch eine Ankündigung aus dem Hause HP:



WINDPOCKET emuliert alle Modelle vom 28C bis zum 48LX unter WINDOWS.

Ob HP an einen echten Bedarf glaubt (es schon die Tasten des Rechners, wenn man sie mit der Maus anfährt) oder ob es sich nur um einen Werbegag handelt (um Computerbesitzern den echten 48SX schmackhaft zu machen) wird erst die Preisempfehlung zeigen.

Wesentlich kostengünstiger erscheint das Programm GRABSCH des österreichischen Hackers Gernot Wildkart.

Dieses Shareware Programm besitzt einen Simulator für die Saturn-CPU und "zieht" das Betriebssystem des 48LX auf den Computer, um es dort laufen zu lassen.

Wer Informationen über diese Produkte haben möchte sollte mit der HP-Vertriebszentrale (Hewlett-Packard-Straße, 6380 Bad Homburg) etwas Geduld haben - die deutsche Markteinführung ist erst für den Herbst geplant, weshalb ausführliche Unterlagen derzeit nur in Englisch und nur nach geduldigem Nachfragen verfügbar sind.

Rainer zu Fall (1166)

# Programmsammlung

## Übersicht über Pas de deux 1-7

von Ralf Pfeifer

### Inhalt:

Programme aus den Artikeln  
Pas de deux Teil 1-7 (Abgekürzt PDD n)  
Graphisto 48 (Abgekürzt GRA)  
Kabelsalat (Abgekürzt KABEL)

### Copyright:

I. Alle hier aufgeführten Programme stehen den CCD-Mitgliedern zu nicht-kommerziellen sowie Schulen und Universitäten zu Lehrzwecken kostenlos zur Verfügung.

Die Weitergabe ist nur dann zulässig, wenn

- II. der Empfänger zu den unter 1. genannten Gruppen gehört
- III. er die Bedingungen des Copyright akzeptiert
- IV. den Hinweis auf Autor und Copyright-Bedingungen erhält
- V. die Übermittlung vollkommen unentgeltlich erfolgt.

### Garantie:

Die Nutzung der Programme geschieht auf eigene Gefahr. Als Autor lehne ich jede Verantwortung für Schäden aller Art, Frust oder sonstiger psychischer oder physischer Beeinträchtigung ab.

Das gleiche gilt für Folgeschäden (z.B. durch Wutanfälle, zerrüttete Ehen bzw. Rechnergehäuse).

Die Programme wurden nach logischer Zusammengehörigkeit organisiert. Die folgende Tabelle ermöglicht die Referenz zu den Dokumentationen in den entsprechenden PRISMA-Ausgaben. Die Pfade in Klammern kann man aufrufen, um das Programm mit RCL zurückzurufen (RCL erlaubt solche Eingaben ohne das aktuelle Verzeichnis zu wechseln).

### Hinweis:

Version PDD n bedeutet, daß die Version in einer anderen Folge von

Pas de deux verändert wurde, und dort nach der Bedienungsanleitung gesucht werden sollte.

Die erste Spalte enthält den Namen des Programms, die zweite beschreibt es kurz, und die dritte gibt die Verzeichnisse an.

*Beispiel:* Wenn TRI im Verzeichnis *MATH TRIG* zu finden ist, bedeutet das, daß zunächst das Verzeichnis *MATH* gewählt werden muß, in welchem sich das Unterverzeichnis *TRIG* befindet. Nach Auswahl von *TRIG* kann man das Programm starten (ggf. *NXT*-Taste betätigen)

### Pas de deux 1 (PRISMA 3/90)

|       |   |                  |
|-------|---|------------------|
| △     | Interaktives Eingabemenü zur Dreiecksberechnung | <i>MATH TRIG</i> |
| HOEHE | Berechnet die Höhen im Dreieck                  | <i>MATH TRIG</i> |
| SHALB | Seitenhalbierende                               | <i>MATH TRIG</i> |
| WHALB | Winkelhalbierende                               | <i>MATH TRIG</i> |
| FLKR  | Fläche und In- und Umkreis des Dreiecks         | <i>MATH TRIG</i> |
| TRI   | Berechnungsprogramm für Dreiecke                | <i>MATH TRIG</i> |
| ZYKL  | Hilfsprogramm (vertauscht Variable zyklisch)    | <i>MATH TRIG</i> |
| DKAL  | Zeigt im LCD Monatskalender an                  | <i>DATUM</i>     |
| OF    | Gausche Osterformel                             | <i>DATUM</i>     |
| →MJ   | Hilfsprogramm (fragt nach Monat und Jahr)       | <i>DATUM</i>     |
| PRP   | Print Programm (Programmzeildrucker)            | <i>PRP</i>       |
| PRDIR | Print Directory druckt die Verzeichnisstruktur  | <i>PRP</i>       |
| GLN   | Verwandelt Polynome in algebraische Objekte     |                  |

|                 |                                   |
|-----------------|-----------------------------------|
| (Version PDD 6) | <i>MATH</i>                       |
| MINV            | Matrixinversion mit Nachiteration |
|                 | <i>MATH</i>                       |

### Pas de deux 2 (PRISMA 4/90)

|       |  |                       |
|-------|--|-----------------------|
| GGT   | Größter gemeinsamer Teiler (schnellste & kürzeste Version) | <i>MATH</i>           |
| XEXP  | Integration (rekursiv)                                     | <i>MATH INTEG</i>     |
| XSIN  | Integration (rekursiv)                                     | <i>MATH INTEG</i>     |
| XCOS  | Integration (rekursiv)                                     | <i>MATH INTEG</i>     |
| FR13  | Wann ist Freitag der 13.                                   | <i>DATE</i>           |
| KAL   | Kalenderdrucker  | <i>DATE</i>           |
| CJ    | Calendar Date to Julian Day Number                         | <i>DATE</i>           |
| JC    | Julian Day Number to Calendar Date                         | <i>DATE</i>           |
| DOW   | Day of the week  | <i>DATE</i>           |
| →MJ   | Hilfsprogramm (s. PDD 1)                                   | <i>DATE</i>           |
| MCOD  | Morse Coder  | <i>MISC</i>           |
| CODE  | Verschlüsselt einen String                                 | <i>MISC GEHEIM</i>    |
| DCD   | Entschlüsselt einen String                                 | <i>MISC GEHEIM</i>    |
| KRYPT | Kryptologisches Hilfsprogramm                              | <i>MISC GEHEIM</i>    |
| MKRY  | Mini Kryptologie   | <i>MISC GEHEIM</i>    |
| SORT  | Universelles Sortierprogramm (Version PDD 7)               | <i>MISC SORTIEREN</i> |
| ΣSORT | Sortiert die Statistikmatrix ΣDAT (Version PDD 7)          | <i>STAT</i>           |

### Pas de deux 3 (PRISMA 5/6-90)

|        |                                 |                |
|--------|---------------------------------|----------------|
| HORNER | Hornerschema für Polynome       | <i>MATH GL</i> |
| ABL1   | 1. Ableitung                    | <i>MATH GL</i> |
| →EQ    | Vektor in algebraisches Polynom | <i>MATH GL</i> |
| POLY/  | Polynomdivision                 | <i>MATH GL</i> |

|         |   |         |
|---------|---|---------|
| PADD    | Polynomaddition   | MATH GL |
| PMUL    | Polynommultiplikation                                     | MATH GL |
| COL     | COL für Polynome  | MATH GL |
| INDP?   | Bestimmt unabhängige Variable in PPAR                     |         |
| NULLWEG | Entfernt Nullstellen                                      | GRAFIK  |
| KR      | Krümmung einer Funktion                                   | MATH    |
| BOGEN   | Bogenlänge einer Funktion                                 | MATH    |
| POLYFIT | Polynomische Regression                                   | STAT    |
| NLIN    | Lineare Regression mit mehreren Variablen (Version PPD 6) | STAT    |
| GM      | Geometrisches Mittel                                      | STAT    |
| HM      | Harmonisches Mittel                                       | STAT    |
| QM      | Quadratisches Mittel                                      | STAT    |
| TIMER   | Mit Programmlaufzeiten in Ticks                           |         |
| TMHR    | Mit Programmlaufzeiten im HMS-Format                      |         |

#### Pas de deux 4+5 (PRISMA 1/91)

|       |   |             |
|-------|---|-------------|
| SDET  | Symbolische Determinante                    | MATH SYMB   |
| CEQN  | Charakteristische Gleichung                 | MATH SYMB   |
| SMUL  | Multiplikation zweier symbolischer Matrizen | MATH SYMB   |
| DIM   | Größe einer symbol. Matrix                  | MATH SYMB   |
| APLY1 | Wendet eine Funktion auf eine Matrix an     | MATH SYMB   |
| APLY2 | Verknüpft zwei symbol. Matrizen             | MATH SYMB   |
| NLGS  | Nichtlineare Gleichungssysteme              | MATH MATRIX |
| VDISP | Anzeigeprogramm für NLGS                    | MATH MATRIX |
| PRIM  | Primfaktorenzerlegung                       | MATH        |

#### Pas de deux 6 (PRISMA 2/3-91)

|        |                            |      |
|--------|----------------------------|------|
| SINTER | Tabelleninterpolation      | STAT |
| COL?   | Findet X- und Y-Spalte der |      |

|        |                                  |      |
|--------|----------------------------------|------|
| MINV   | Statistikmatrix                  | STAT |
| NEWTON | Matrixinversion                  | MATH |
| JRPT   | Newtoninterpolation für Polynome | MATH |
| TRAPEZ | Jährliche Alarmwiederholung      |      |
|        | Fläche eines Polygonzuges        | MATH |

#### Pas de deux 7 (PRISMA 4-6/91)

|         |   |                |
|---------|---|----------------|
| INSERT  | Fügt in Vektoren, Listen, Stack ein         | MISC           |
| SORT    | Der Universal-sortierer                     | MISC SORTIEREN |
| PSORT   | Stacksortieren                              | MISC SORT.     |
| BISORT  | Stacksortieren                              | MISC SORT.     |
| BITSORT | Stacksortieren                              | MISC SORT.     |
| DUOSORT | Für komplexes Sortieren                     | MISC SORTIEREN |
| DAT     | Hilfsprogramm um Daten zu sortieren         | MISC SORTIEREN |
| DUDEN   | Hilfsprogramm zum alphabetischen Sortieren  | MISC SORTIEREN |
| →ROM    | Arabische in römische Zahlen                | MISC           |
| →ARAB   | Römische in arabische Zahlen                | MISC           |
| NGL     | Gleichungen 1.-4. Grades mit Formeln gelöst | MATH GL        |
| SSORT   | Statistikmatrix sortieren                   | STAT           |

#### Graphisto - 48 (PRISMA 4-6/91)

|         |  |        |
|---------|--|--------|
| BALLAST | Verkürzt die Wartezeit bei der Garbage Collection  | GRAFIK |
| GAND    | Überlagert GROBs mit AND                           | GRAFIK |
| GNOT    | Invertiert Bereiche in GROBs                       | GRAFIK |
| GPACK   | Entfernt Leerzeilen und -spalten am Rand von GROBs | GRAFIK |
| GPLOT   | Plottet RAM-sparend                                | GRAFIK |
| SQPL    | Plottet noch RAM-sparender                         | GRAFIK |
| IDISP   | Erweitertes DISP                                   | GRAFIK |

**Hinweis:** Die gesamte Programmserie ist für Mitglieder der Diskettengruppen (MS-DOS ATARI) auf den Infodisketten zu finden.

Ralf Pfeifer  
Rubensstr. 5  
D5000 Köln 50

## Workspace - Ergänzung

Zum Workspace-Programm von Georg Hoppen, Prisma 1/91, S.19 habe ich einige Ergänzungen.

Man kann aus den Workspacebereichen heraus andere Anwendungen laufen lassen, die in anderen Unterverzeichnisästen von HOME 'leben', nach deren Ablauf ist wieder der Ausgangsbereich im Workspace aktiv.

Im Programm STRT wird die Existenz einer Variablen **RUN** im Verzeichnis WSU abgefragt.

Wenn **RUN** mit z.B. dem Inhalt "GS GX" da ist, dann läuft Programm GX im Verzeichnis GS ab.

Damit dies auch richtig funktioniert, muß in der Variablen **GLOB** im Verzeichnis WSU folgendes abgespeichert sein:

```
{ { "GX" « PATH 1 →LIST STRT » } }.
```

Darin ist GX der Programmname (deshalb in Anführungszeichen), kann aber auch ein anderer aussagekräftiger Text sein (dann ohne Anführungszeichen).

Die Pfadliste wird noch einmal in einer Liste gespeichert, da sonst der Befehl EVAL im Programm STRT die Liste evaluiert, d.h. ohne etwas anderes zu machen in das Ausgangsverzeichnis zurückkehrt. So ist nach EVAL die Pfadliste vorhanden und kann zur späteren Rückkehr benutzt werden. Diese Anwendung ist bei mir eine schnelle Telefonliste, siehe separater Artikel.

Werden in einem Workspace noch weitere Unterverzeichnisse gebraucht, um z.B. Programme von Daten getrennt zu halten (für die generelle Übersicht oft unerlässlich!), dann muß dieser Weg manuell in der Variablen **SPEC** des jeweiligen Verzeichnisses niedergelegt werden.

Beispiel für Programm XX, das in der XSDT-Ebene darunter laufen soll :

```
{ { "XX" « XSDT XX » } }
```

Günter Schapka  
Rebusgasse 11; D6100 Darmstadt

# Schnelle Telefonliste

## von Günter Schapka

Voraussetzung: Toolkit von Donelly.

Im Prisma 5/6-90 war ein Adreßverwaltungsprogramm von M. Breidenbach und H. Lehmann vorgestellt worden.

Durch die bequemen Funktionen Editieren und Sortieren habe ich dieses seitdem für die Pflege einer Telefonliste in Benutzung.

Mein Suchprogramm gab es bereits vorher, es besteht mittlerweile aus einem allgemeinen Suchunterprogramm GX2 sowie dem speziellen Telefonsuch-Unterprogramm GX1.

Im Hauptprogramm GX wird entschieden, welches Unterprogramm zu benutzen ist:

Bei Eingabe nur eines Namens (d.h. nur eine Stackebene belegt) >>> Telefonliste, bei zusätzlicher Eingabe eines Filenamens (d.h. zwei Stackebenen belegt) das entsprechende File.

GX akzeptiert beim Aufruf eine Li-

ste in Ebene 1 des Stacks, das ist z.B. die Pfadliste, wenn aus einem Workspace aufgerufen wurde.

Im Laufe der Zeit wuchs die Telefonliste.

Infolge der Größe brauchte das ursprüngliche, jetzt allgemeine Suchprogramm entsprechend lange, um einen Eintrag zu finden, da es mit dem POS-Befehl die komplette Datei abklappert.

Diese Methode hat den Vorteil, auch bei Namensmerkmalen an anderen als der ersten Stelle alle Einträge abzuliefern, wenn z.B. hinter dem Name irgendwo CCD steht. Aus diesem Grund wurde das Telefonsuchteil so abgeändert, daß es mit dem ersten Buchstaben des Suchnamens in einem Indexstring die entsprechende Position in der TL-Datei direkt anspringt, dann mittels POS das Auftauchen des Suchstrings ermittelt und anzeigt.

So ist die Suchzeit erfreulich kurz,

wenn der Erstname eingegeben wurde.

Bei anderen Einträgen wird ab der Position des ersten Buchstabens die komplette Datei abgesucht, wie gehabt, ist es dann zwar langsam, findet aber sicher alle Einträge.

Wird nach Anzeige eines Eintrags mit der Rückpfeiltaste geantwortet, dann erfolgt keine weitere Suche, sondern ein schneller Abbruch; bei jeder anderen Antworttaste wird der Rest der Datei auch noch durchsucht.

Das Erstellen der Indexstrings INDEX erfolgt mittels Programm NIND (Neue Indizierung), die Überprüfung eines vorhandenen INDEX Files kann mittels TIND (Test INDEX) vorgenommen werden:

Ziffermäßige Ausgabe der Alphabetpositionen in einer Liste.

Weitere Details zu den Programmen findet Ihr jeweils beim Kommentar.

### TIND

« { } → in li

```
« 1 26
  FOR i in i
SUBNUM li SWAP +
```

```
'li' STO
  NEXT li
» »
```

### NIND

« FILE OBJ → INDEX 0 →

s in li

```
« 1 s
  FOR i s 1 SROLL
```

### CAR

Leerliste in lokaler Variablen li zum Abspeichern des Ergebnisses. 26 Buchstaben hat das Alphabet  
SUBNUM liefert für aktuelle Position im Alphabet den Zifferwert des Index  
Anzeige der Liste im Stack

FILE (hier Telefonliste) auf dem Stack  
s: Size von FILE  
in: INDEX-String  
li: Index-Liste zur Anzeige mit TIND  
Alle Stackobjekte \* nach oben rotiert d.h. erstes Listenelement in Ebene 1.  
CAR gibt ersten Buch-

NEXT s →LIST

```
'li' STO in 1 26
  FOR i li i 64 +
CHR POS i SWAP
PUTCHR
  NEXT
» → in
« 1 DUP 26
  FOR i in i
SUBNUM DUP2
  IF >
  THEN DROP DUP
in i ROT PUTCHR
'in' STO
  ELSE SWAP
DROP
  END
  NEXT DROP in
» DUP TIND »
```

### Beispiel-INDEX-LISTE

```
{ 1 4 5 6 8 9 11 13
18 18 20 24 28 28
30 31 31 33 34 40
43 46 47 47 47 47 }
```

staben zurück  
jetzt nur Anfangsbuchstaben in Liste

Feststellen der Position jedes Buchstabens und Ablegen im Indexstring in.

Nullen für nicht vorhandene Buchstaben werden ersetzt durch die Position des Vorgängers

Anzeige des neuen Index-Strings als Liste, der String steht noch im Stack und muß manuell auf INDEX gespeichert werden.

Doppeleinträge zeigen, daß die Folge-Buchstaben in der Liste nicht vorkommen, d.h. Suche

würde sonst zu weit hinführen und entsprechend lange dauern.  
Hier z.B. keine Einträge für J, N, Q und W-Z

```

GX
«
  IFERR DUP
  THEN { }
  ELSE DROP
  END → p

  « "TL.LISTE" {
  :File:
  :Text:" { 2
  0 } \Ga } INPUT OBJ→
  DTAG "" +

  IFERR SWAP
  THEN p GX1
  ELSE p ROT GX2

  END CLEAR 1
MENU
» »

GX1
« → p
«
  « PATH HOME ZZZ
FILE INDEX ROT EVAL
  » 0 → n t i

  « t EVAL 'i'
STO 't' STO n 1
SUBNUM 64 - i SWAP
    
```

keine Liste im Stack >> Leerliste  
sonst Duplikat löschen p. lokale Variable für Pfad

Eingabe  
Eingabe zu Textstring machen  
Nur 1 Zeile Eingabe + Pfadliste im Stack: Aufruf Telefonsuche  
2 Zeilen Eingabe + Pfadliste: allgemeines Suchprogramm

Pfad in lokale Variable p speichern  
Programm zum Holen von Liste + Index-String vorläufig in lok. Variable t speichern, gesuchter Name in n.  
Index-String in lok.Var. i  
Liste in lokale Variable t  
Position des 1. Buchstabes des gesuchten Namens aus dem

```

SUBNUM t SWAP 1
SWAP EXTRACT DROP 1
t SIZE

FOR j
  IF DUP n
    POS
    THEN DUP
  CLLCD 3 DISP -1
  WAIT IP DUP
    IF 25 ==
    SWAP 94 == OR
    THEN 1
MENU DROP p EVAL
KILL
  ELSE t
  SWAP NXTOB
  END
  ELSE
  END t SWAP
NXTOB
  NEXT 1 MENU

DROP p EVAL
» » »

GX2
« 0 → n p t x
« n EVAL 1
DO GETI DUP t
POS
  IF 0 ≠
  THEN CLLCD 3
  DISP -1 WAIT DROP
  ELSE DROP
  END
  UNTIL DUP 1 ==
  END p EVAL
» » }
    
```

Indexstring holen  
ersten Eintrag mit diesem Anfangs-Buchstaben aus Liste holen

auf Suchstring überprüfen  
gefunden >>> Anzeige  
Tastencode als Integer

bei Rückpfeil oder Space sofort beenden und Pfad zurück in WRK-Bereich

andere Tasten: weiter suchen

wenn komplette Liste durchsucht  
Pfad zurück in WRK-Bereich

Liste auf Stack

Suchstring vorhanden >> Anzeige

komplette Liste durchgesucht: Pfad zurück

Günter Schapka

# Alarmer zurückholen

Nach einem Rechnerabsturz fragt der 48er, ob er versuchen soll, so viel Memory wie möglich zurückzuholen. Hat man ein Sortiment Alarmer gehabt, so sollte man diese Frage auf alle Fälle mit JA beantworten. Wenn nach dem Zurückholen die Verzeichnisnamen mit D.01, D.02 usw. durchnummeriert sind, dann ist der Inhalt von HOME weg. Die Alarmer stecken dann im letzten

dieser numerierten Verzeichnisse, allerdings nicht mehr im USER- sondern im internen Systemformat. Für uns normalerweise nicht zu gebrauchen; lediglich der Text ist noch lesbar, Speicherversuche mittels STOALARM führen zu keinem Erfolg. Erfolg aber liefert die SYSEVAL-Liste: # 0E1D8h SYSEVAL wandelt das interne Alarmformat in's User-

format um. Das kleine Programm arbeitet die komplette Liste ab.

```

« IFERR OBJ→
  THEN DROP
  ELSE 1 SWAP
  FOR i # E1D8h
SYSEVAL STOALARM
DROP
  NEXT
  END »
    
```

Günter Schapka  
Rebusgasse 11  
6100 Darmstadt

# RAM/ROM Editor beim HP42S - 32kByte RAM ?? von Ralf Pfeifer

In PRISMA 1/89 habe ich den HP42S vorgestellt, ein Gerät, welches kompatibel zum HP41C/CV ist.

In dieser Vorstellung erwähnte ich zwei Punkte, die inzwischen wohl der Korrektur bedürfen:

Zum einen hatte ich festgelegt, daß der HP42S die synthetische Programmierung nicht unterstützt.

Das ist grob falsch, denn der Rechner besitzt einen Editor, mit dem man an jeder beliebigen Stelle im Adressbereich (d.h. RAM wie auch ROM) herumwerkeln kann, das RAM läßt sich damit auch verändern.

Der Einstieg in diesen Editor erfolgt beim HP42S genauso wie beim HP48SX:

- ☞ Rechner einschalten,
- ☞ ON, drücken, LOG drücken, ON loslassen, LOG wieder loslassen.
- ☞ Die Pfeiltaste schaltet den Editor ein.

Nun erscheint links oben die Adresse #023F1 (das # bedeutet Hexadezimalzahl) und hinter dem Doppelpunkt 16 HEX-Zeichen aus dem ROM. In diesem Modus sind folgende Tasten aktiv:

- + Adresse um #1 erhöhen
- Adresse um #1 erniedrigen
- \* Adresse um #100 erhöhen
- / Adresse um #100 erniedrigen
- ▲ Adresse um #1000 erhöhen
- ▼ Adresse um #1000 erniedrigen
- COS Obere Displayzeile ausdrucken und um #10 erhöhen

Die obere Tastenreihe entspricht den HEX-Zahlen #A-#F (so wie im BASE-Menü), zusammen mit den Zifferntasten kann man damit jedes

Zeichen in's RAM schreiben.

Im ROM bewirken die Tasten lediglich die Erhöhung der Adresse um #1.

Über die Wirkung der Pfeiltaste und des Dezimalpunktes habe ich noch keine Erkenntnisse, nach dem Einstieg in den Editor bewirkt der Punkt irgendeine merkwürdige Anzeige.

Der Ausstieg aus dem Editor geht wieder wie beim HP48SX, indem man einen Systemhalt durchführt:

- ☞ ON drücken, Wurzeltaste drücken,
- ☞ ON loslassen, Wurzeltaste wieder loslassen.

Dies führt zu "Machine Reset", aber offensichtlich löscht der HP42S hierbei nichts, Stack und ALPHA-Register bleiben erhalten.

Nun zum RAM-Bereich:

Dieser beginnt bei #50000. Ab hier wirken auch die Zifferntasten 0-9 und A-F so, daß das Zeichen rechts vom Doppelpunkt ersetzt wird und zugleich nach rechts aus der Anzeige verschwindet. Will man die Veränderung betrachten, so muß man nach der Änderung noch einmal mit zurückgehen.

Der Adresszähler numeriert die Halbbytes (= 4Bit) durch, man kann leicht nachrechnen, daß für den von HP genannten RAM-Bereich von 8192 Bytes zu 8 Bit genau  $16384 = \#4000$  Adressen existieren müssen.

Wenn man also bei Adresse #50000 beginnt, dann sollte #53FFF die letzte Adresse sein, die man mit dem Editor verändern kann.

Tatsächlich kann man aber alle

Halbbytes bis Adresse #5FFFF verändern, der HP42S muß also über 32kByte internes RAM verfügen.

Merkwürdig finde ich außerdem, daß mein ALPHA-Register zwischen #583F8 und #5844B liegt, weitere Adressen kenne ich bislang (noch) nicht.

Wichtig zu wissen ist, daß der HP42 auch die Halbbytes vertauscht.

Steht im ALPHA-Register "ABC", so schreibt der HP41 #414243 in's RAM, der HP42 verdreht aber zu #142434, so wie es der HP48SX und der HP28S auch tun.

Der zweite Punkt, der sich vielleicht korrigieren läßt, ist die Rechengeschwindigkeit.

In 1/89 habe ich festgestellt, daß der HP42S nur 1,7 Mal schneller arbeitet als der HP41C.

Führt man aber Funktionen über die Tastatur durch, so ist das Gerät unheimlich schnell.

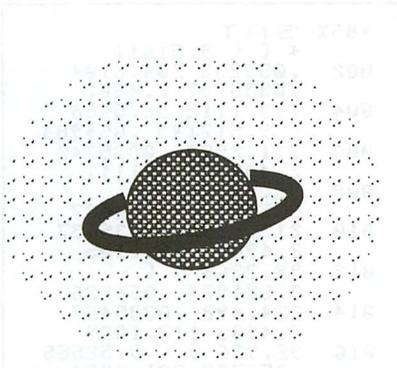
Eine 7\*7 Matrix zu invertieren bedeutet eine Menge Rechenarbeit:

Bei dieser Größe brauchen die schnellsten Algorithmen etwa  $7^3=343$  Punktoperationen (\* bzw. /), dennoch kommen der "langsame" HP42S und der "schnelle" HP48SX in der gleichen Zeit zur Lösung.

Hier hat Hewlett Packard den HP42S anscheinend stark gebremst, und vom HP28 wissen wir aus PRISMA 90.03.37, wie man die Taktfrequenz per Software verändert.

Wenn also jemand die richtige Stelle im RAM findet, dann könnte man die alten 41'er Programme so richtig in Schwung bringen.

Ralf Pfeifer  
Rubenstraße 50  
D5000 Köln 50



### Äußere Planeten, Kleinplaneten, wiederkehrende Kometen Erweiterung von EPH (PRISMA 5/6/90)

Diese Erweiterung beseitigt einmal einen Fehler in PRZ:

statt .33 heißt es dort .033. Dann ist in der Transformation E A, elliptisch nach äquatorial, bei der Transformation auf ein neues Äquinoktium die Neigung der Erdachse fehlerhaft.

Es ist notwendig, einige Programmteile aufzuspalten, weil die vorliegende Erweiterung die Transformation von Bahnelementen auf die aktuelle Epoche erfolgt.

Weiter sind einige Namen von Unterprogrammen geändert, um Bytes zu sparen.

Es gibt folgende *Umbenennungen*:  
IPS in IS, HS in H, SH in G, MM in D und TOG in TG.

*Aufspaltungen* sind:

IS in IS KK mit EN, LS in DB  
B→E, PRZ in PRN PRL.

FKP und SKP fallen weg. KP ist neu und allein mit Stackoperationen gemacht.

Änderungen außer Namensänderungen enthalten:

IS, B→E, PRN, RM.

Neu ist EN, die Berechnung des Anfangs des Bernoulli-Jahres aus der Jahreszahl (iiii).

Damit ist EPH so geändert, daß Körper aufgenommen werden können, deren Bahnen nur mit oskulierenden Bahnelementen befriedigend beschrieben werden können. Das Bindeglied zum bisherigen Programm bildet das Unterprogramm AS.

# Orte aller Körper mit stückweise elliptischen Bahnen im Sonnensystem von Dr. G. Heilmann

Von AS wird folgende Eingabe verlangt:

Ebene 7: eine Liste von Listen mit oskulierenden Bahnelementen,

Ebene 6: die Anzahl  $k>0$  von Unterlisten dieser Liste,

Ebene 5: die Anzahl von Tagen, die die oskulierenden Bahnelemente einer Unterliste gelten,

Ebene 4: das julianische Datum der Epoche, für die die Bahnelemente der ersten Unterliste berechnet sind,

Ebene 3: den Namen des Körpers,

Ebene 2: Äquatorialdurchmesser in km,

Ebene 1: das Äquinoktium, auf das sich die Bahnelemente beziehen (jjjj).

Die Liste für Ebene 6: enthält die Unterlisten, der Zeit ihrer Gültigkeit aufsteigend geordnet.:

a: Länge der Hauptachse in astronomischen Einheiten,

e: Exzentrizität  $e, 0<e<1$ ,

M: Mittlere Anomalie,

n: mittlere tägliche Bewegung,

i: Bahnneigung gegen Ekliptik,

$\Omega$ : Länge des aufsteigenden Knotens,

$\tilde{\omega}$ : Länge des Perihels.

Statt M wird auch L, die mittlere Länge und  $\tilde{\omega}$  auch  $\omega$ , der Winkel zwischen Perihel und aufsteigendem Knoten, angegeben.

Es gilt:  $L=M+\tilde{\omega}$  und  $\tilde{\omega}=\omega+\Omega$ .

Diese Eingabe für AS erzeugen die Programme JUP und SAT des Listings für die Planeten Jupiter und Saturn. die Daten gelten vom 28.7.1990 bis 18.1.1995. Diese Bahnelemente sind für die meisten Be-

dürfnisse über einen noch größeren Zeitraum brauchbar.

Dann aber erscheint während der Berechnung die Meldung "untersritten" beziehungsweise "übersritten".

Die Bedienung des Programms bleibt unverändert.

Die Ausgabe ist jetzt mit Benennung versehen und im äquatorialen Modus erweitert (Muster).

t ist der Stundenwinkel zum Beobachtungszeitpunkt am Beobachtungsort. Daraus ist eine leichte Näherung für die Kulminationszeit des Körpers angebar.

d ist der scheinbare Durchmesser des Körpers.

Auch nicht-periodische Kometen sind erfaßbar.

Dann  $isr=t e>1$ .

In B→E ist dann eine Bedingung einzufügen, die zwischen KP und den Programmen für hyperbolische Bahnen entscheidet. Diese Programme sind im ersten Unterverzeichnis von EPH aufzunehmen.

Bahnelemente befinden sich z.B. in Ahnerts Kalender für Sternfreunde 1991, J.A.Barth, Leipzig 1990 und in O. Montenbruck, Grundlagen der Ephemeridenrechnung, Dr.Vehrenbert GmbH, München 1984.

*Wichtiger Hinweis:* (siehe Text) muß nach UT laufen. Beim ersten Gebrauch muß IN bestätigt werden, um die Ortskoordinaten einzugeben.

Dr. G. Heilmann  
Obernhof Str. 15  
D5408 Seelbach

*Programme siehe nächste Seiten →*

# 48SX

```

48SX STZ
< SZ S 15 / →HMS 4
002 FIX →STR
"Sternzeit:" SWAP +
004 CLLCD 1 DISP 1
FREEZE
006 > Bytes : 75.5
Checksum : # B68Eh

48SX SON
< "Sonne" 23200 20
002 SF 6 IS
> Bytes : 51.5
004 Checksum : # 1B7Ah

48SX MON
< SZ T RM 90 SWAP -
→V3 DUP ABS SWAP
DUP V→ T E→A →V3
004 6378.14 S 90 B -
→V3 - V→
006 IF 4 FS?
THEN DA ROT DROP
ELSE ROT DROP DQ
ROT 1 FIX 3476 OVER
/ 3438 * 'd' →TAG
→STR " _arcmin
" + 0
012 FIX SWAP 'r' →TAG
→STR " km" + + +
" END "
016 " + SWAP
NEG DUP LS
018 149597870 * SWAP -
PH "Mond" "
020 " +
SWAP + KC
022 > Bytes : 332
Checksum : # 83AEh

15 EPH
15 IPL
00 E
08 STZ
08 SON
08 MON
08 MER
08 VEN
08 MAR
08 JUP
08 SAT
08 IN
08 DT
08 PZ
08 AZ
00 S
00 T
00 B
00 L
00 Q
02 N
00 A
05 CST
08 SZ
08 IS
08 AS
08 KK
08 LS
08 DB
08 B→E
08 E→A
08 PRN
08 PRL
08 A→H
08 P→R
08 C→S
08 PH
08 KP
08 RM
08 EN
08 H
08 D
08 G
08 KC
08 DA
08 DQ
08 TG
    
```

```

48SX JUP
< ( ( 5.203264
.048228 100.8077
.0830802 1.3065
004 100.0444 14.9836 )
( 5.203234 .048241
006 117.4245 .0830809
1.3065 100.0443
008 14.9816 ) {
5.203176 .048256
010 134.0432 .0830823
1.3065 100.0443
012 14.978 } { 5.203094
.048275 150.662
014 .0830843 1.3065
100.0443 14.9749 }
016 ( 5.202993 .048296
167.2793 .0830867
018 1.3065 100.0445
14.9739 ) {
020 5.202879 .04832
183.8937 .0830894
022 1.3065 100.0447
14.9766 } {
024 5.202757 .048343
200.5048 .0830923
026 1.3065 100.045
14.9835 } {
028 5.202637 .048366
217.113 .0830952
030 1.3065 100.0454
14.9944 } {
032 5.202527 .048386
233.7197 .0830979
034 1.3064 100.0456
15.0076 } {
036 5.202437 .048402
250.3274 .0831
038 1.3064 100.0458
15.0207 } } 10 200
040 2448200.5 "Jupiter"
142796 1950 AS
042 > Bytes : 870
Checksum : # 2DB3h

48SX IN
< 2 TG
002 > Bytes : 18
Checksum : # 229Ah

48SX DT
< 1 TG
002 > Bytes : 18
Checksum : # 6009h

48SX PZ
< 3 TG
002 > Bytes : 18
Checksum : # A94Bh

48SX AZ
< 4 TG
002 > Bytes : 18
Checksum : # 4502h

48SX EN
< 1900 -
002 365.24219878 *
.31352 + 36525 /
004 > Bytes : 62
Checksum : # D2F7h

48SX MER
< "Merkur" 4878 7
002 IS
> Bytes : 39.5
004 Checksum : # 60B2h

48SX VEN
< "Venus" 12104 8
002 IS
> Bytes : 38.5
004 Checksum : # 1A10h

48SX MAR
< "Mars" 6794 9 IS
002 > Bytes : 37.5
Checksum : # BF06h
    
```

```

48SX SAT
< ( ( ( 9.51611
.055233 204.8104
.0335797 2.4907
004 113.1112 92.5688 )
( 9.517149 .054984
006 211.2553 .0335742
2.4907 113.1111
008 92.8022 ) {
9.519141 .054637
010 217.8103 .0335637
2.4905 113.1105
012 92.9268 } {
9.521974 .054226
014 224.492 .0335487
2.4903 113.1089
016 92.926 } { 9.52555
.053788 231.3094
018 .0335298 2.49
113.106 92.7904 } {
020 9.529784 .053356
238.265 .0335075
022 2.4897 113.1018
92.5172 } { 9.5346
024 .052964 245.3541
.0334821 2.4894
026 113.0964 92.1109 }
{ 9.539923 .052645
028 252.5641 .0334541
2.4891 113.0901
030 91.5841 } {
9.545664 .052428
032 259.8728 .0334239
2.4888 113.0835
034 90.959 } { 9.551712
.05234 267.2465
036 .0333921 2.4886
113.0773 90.2697 }
038 } 10 200 2448200.5
"Saturn" 120000
040 1950 AS
> Bytes : 869
042 Checksum : # 6483h

48SX SZ
<
002 IF 1 FS?
THEN
004 "Datum, Zeit:" (
":D:
006 ":Z:0
" { 1 0 } .
008 V ) INPUT OBJ→
ELSE DATE TIME
010 END HMS→
IF 2 FS?C
012 THEN
"Breite, Länge, Äqu
014 in."
{
016 ":B:50.1839
:L:-7.5119
:Q:1950"
018 ( 1 0 ) V ) INPUT
020 OBJ→ EN 'Q' STO
HMS→ 15 'L' STO
022 HMS→ 'B' STO
END DUP 1.0027379
024 * SWAP 1.0119 4
ROLL DDAYS .5 + DUP
026 ROT 24 / + 36525
SWAP OVER / 'T' STO
028 / DUP .0929 *
8640184.542 + *
030 23925.836 + G + L -
15 * D 'S' STO
032 > Bytes : 449
Checksum : # AB08h

48SX KC
< CLLCD 1 DISP 3
002 FREEZE
> Bytes : 22.5
004 Checksum : # 25FDh
    
```

# 48SX

```

48SX IS
< SF 'A' STO 'N'
002 STO SZ LS
CASE 7 FS?C
004 THEN .387099
.205614 .00002
006 102.2794
149472.5153 28.7537
008 .3705 7.0029 .0019
47.146 1.185
010 END 8 FS?C
THEN .723332
012 .006821 -.000048
212.6032 58517.8039
014 54.3838 .508 3.3936
.001 75.78 .9
016 END 9 FS?C
THEN 1.523692
018 .093313 .000092
319.5294 19139.8585
020 285.4322 1.0698
1.8503 -.0007
022 48.786 .771
END
024 END
IF 6 FC?
026 THEN DB B>E DUP
ROT + DUP
028 END V> KK
> Bytes : 496.5
030 Checksum : # F3B7h.

48SX AS
< EN 'E' STO 'A'
002 STO 'N' STO SZ
2415020 - T 36525 *
004 SWAP - DUP2 SWAP /
1.5 + IP DUP 1
006 IF <
THEN DROP 1
008 "unterschriften!"
CLLCD 1 DISP 1
010 FREEZE
END 4 ROLL SWAP
012 DUP2 <
IF <
014 THEN DROP
"überschritten!"
016 CLLCD 1 DISP 1
FREEZE
018 ELSE SWAP DROP
END SWAP OVER 1 -
020 4 ROLL * - 3 ROLLD
GET OBJ> DROP OVER
022 - SWAP 90 OVER OVER
- 5 PICK E T PRN
024 OVER 10 ROLLD 3
PICK 11 ROLL PRL 3
026 ROLLD + 1 7 ROLL
P>R 5 ROLL 8 ROLL -
028 P>R NEG 1 8 ROLL
P>R 5 ROLL * SWAP 4
030 ROLL * - SWAP -15
CF -16 CF >V2 -16
032 SF V> SWAP DROP 4
ROLL + ROT SWAP 7
034 ROLL 7 ROLL 6 ROLL
8 ROLL * 7 ROLL +
036 B>E LS + 6 SF V> KK
> Bytes : 561.5
038 Checksum : # 1F7Ah

48SX H
< * + SWAP SIN
002 > Bytes : 20
Checksum : # 209Bh

48SX LS
< 1 .016751 T
002 .000042 * -
358.4758 35999.0498
004 T * + KP 281.2208
1.7192 T * + + 90
006 >V3
> Bytes : 125
008 Checksum : # 8220h

```

```

48SX KK
<
002 IF 3 4 FC? SWAP
FS? AND
004 THEN T Q PRN PRL
Q
006 ELSE T
END E>A
008 IF 4 FS?
THEN DA
010 ELSE DQ SWAP A
OVER / .0013788 * 1
012 FIX 'd' >TAG >STR
IF 20 FS?C
014 THEN "_arcmin"
" ELSE "_arcsec"
016
" END + SWAP 4
018 FIX 'r' >TAG >STR +
"-au" + +
020 " END N "
" + SWAP
022 +
024 IF 4 6 FC? SWAP
FC? AND
026 THEN "
" + SWAP
028 ROT PH
END
030 IF 4 6 FC?C SWAP
FS? AND
032 THEN SWAP ROT
DROP2
034 END KC
> Bytes : 347
036 Checksum : # FC3h

48SX P>R
< -15 CF -16 SF >V2
002 -16 CF V>
> Bytes : 54
004 Checksum : # 4AE2h

48SX B>E
< KP ROT + P>R ROT
002 P>R C>S SWAP 4 ROLL
+ SWAP >V3
004 > Bytes : 57.5
Checksum : # 5066h

48SX E>A
< -.013004 *
002 23.452295 + 4 ROLL
ROT P>R 4 ROLL P>R
004 4 PICK P>R 4 ROLL 5
ROLL P>R NEG 4 ROLL
006 + SWAP ROT + C>S
> Bytes : 113.5
008 Checksum : # CE59h

48SX PRN
< OVER - SWAP 1 -
002 NEG > t u
< t DUP .04 * u
004 .5 * 869.81 + - * u
DUP .61 * 3289.48 +
006 * 629554.98 + + G t
DUP -.033 * u -.067
008 * 47.003 + + * G t
DUP 1.111 * u 2.222
010 * 5029.097 + + * G
>
012 > Bytes : 272.5
Checksum : # 6803h

48SX PRL
< 6 ROLL 5 ROLL P>R
002 6 ROLL 6 PICK - P>R
5 PICK P>R 4 ROLL 6
004 ROLL P>R SWAP ROT -
SWAP ROT + SWAP C>S
006 SWAP 5 ROLL + 4
ROLL + SWAP
008 > Bytes : 117.5
Checksum : # F086h

```

```

48SX C>S
< -15 CF -16 CF >V3
002 -15 SF -16 SF V>
> Bytes : 67
004 Checksum : # D547h

48SX PH
< DOT LASTARG ABS
002 SWAP ABS * / 1 + 2
/ 3 FIX 'k' >TAG
004 >STR +
> Bytes : 59.5
006 Checksum : # 6012h

48SX KP
< -3 SF RAD D>R π
002 DUP + MOD DUP
IF π >
004 THEN 21 SF π DUP
+ SWAP -
006 END DUP2 + DUP
IF π >
008 THEN DROP π
END DUP DUP SIN 5
010 PICK * - 3 PICK - 3
PICK DUP DUP SIN 7
012 PICK * - 5 PICK -
DO OVER 5 PICK -
014 4 PICK ROT / 1 - /
+ DUP DUP SIN 7
016 PICK * - 5 PICK -
DUP 7 RND 0
018 UNTIL ==
END DROP 4 ROLLD
020 3 DROPN
IF 21 FS?C
022 THEN π DUP + SWAP
-
024 END R>D DEG 1 3
ROLL DUP2 5 PICK
026 COS * - 5 ROLL * 4
ROLLD + LASTARG - /
√ SWAP 2 / TAN *
ATAN DUP + D
030 > Bytes : 363
Checksum : # 1766h

48SX DA
< 1 FIX 5 ROT -
002 SWAP B A>H 90 SWAP
- ROT DROP SWAP D
004 'a' >TAG >STR "
"
006 + SWAP 'h' >TAG
>STR + "
008 "
"
> Bytes : 120.5
010 Checksum : # CADBh

48SX TG
< DUP
002 IF FS?
THEN CF
004 ELSE SF
END
006 > Bytes : 30
Checksum : # FABFh

48SX A>H
< 4 ROLL ROT P>R 4
002 ROLL P>R NEG SWAP 4
PICK P>R NEG 4 ROLL
004 5 ROLL P>R SWAP ROT
+ SWAP ROT + ROT
006 SWAP C>S
> Bytes : 97.5
008 Checksum : # 7132h

48SX D
< 360 MOD
002 > Bytes : 23
Checksum : # 64BEh

48SX G
< 3600 /
002 > Bytes : 23
Checksum : # D040h

```

```

48SX R M
< DUP DUP -.001133
002 * 481267.883142 + *
270.434164 +. D OVER
004 DUP .009192 *
477198.849108 + *
006 296.104608 + D 3
PICK DUP .002078 *
008 1934.142008 - *
259.183275 + D 4
010 PICK DUP .000303 *
36000.768925 + *
012 279.696678 + D 5
ROLL DUP -.00015 *
014 35999.04975 + *
358.475833 + D → a
016 b c d e
< 6378.14 3423 b
018 DUP DUP + COS 10 *
SWAP COS 187 * +
020 b a d - DUP + DUP
ROT DUP2 - SWAP ROT
022 + COS 3 * ROT COS
28 * + SWAP COS 34
024 * + + G SIN / b DUP
DUP + SWAP +
026 LASTARG e - LASTARG
+ LASTARG a d - DUP
028 DUP + 4 ROLL +
LASTARG - LASTARG
030 ROT SWAP - LASTARG
SWAP 7 ROLL -
032 LASTARG 6 ROLL SWAP
- LASTARG SWAP SIN
034 2370 * SWAP SIN
-668 H 165 H 4586 H
036 206 H 22640 H 212 H
192 H -125 H -110 H
038 148 H 769 H 36 * +
a c - DUP + SIN 412
040 * SWAP OVER - G a +
DUP c - ROT e SIN
042 541 * + G + SIN
18520 * d DUP + a c
044 + - b + LASTARG -
LASTARG DROP e +
046 LASTARG - LASTARG
DROP a c - b - DUP
048 b - 8 ROLL SWAP SIN
-25 H 21 H -526 H
050 11 H -23 H -31 H 44
* + G
052 >
> Bytes : 1093
054 Checksum : # 54C2h
    
```

```

äquatorial: Flag 4 gelöscht
Mond
:α: 14.3650_h
:δ: -20.4235_
:t: 15.0608_h
:d: 30.4_arcmin
:r: 393049_.km
:k: 0.410
STR | SGN | MON | MER | EN | M+
-----|-----|-----|-----|-----|-----
Sonne
:α: 9.4112_h
:δ: 13.5155_
:t: 20.0220_h
:d: 31.6_arcmin
:r: 1.0127_au
STR | SGN | MON | MER | EN | M+
-----|-----|-----|-----|-----|-----
Venus
:α: 10.0924_h
:δ: 3.2426_
:t: 19.3445_h
:d: 56.7_arcsec
:r: 0.2943_au
:k: 0.023
STR | SGN | MON | MER | EN | M+
-----|-----|-----|-----|-----|-----
Jupiter
:α: 9.4656_h
:δ: 14.1114_
:t: 19.5756_h
:d: 30.9_arcsec
:r: 6.3653_au
JUP | SGN | MON | MER | EN | M+
-----|-----|-----|-----|-----|-----
48SX D B
< T * + SWAP T *
002 ROT + ROT T * 4 STD
ROLL + 8 ROLL 7
004 ROLL T * 8 ROLL + 6
ROLL T * 7 ROLL +
006 > Bytes : 97.5
Checksum : # F831h
    
```

```

azimutal: Flag 4 gesetzt
Mond
:a: 69.2_
:h: -42.6_
:k: 0.411
STR | SGN | MON | MER | EN | M+
-----|-----|-----|-----|-----|-----
Sonne
:a: 105.9_
:h: 30.5_
STR | SGN | MON | MER | EN | M+
-----|-----|-----|-----|-----|-----
Venus
:a: 107.2_
:h: 18.1_
E | SGN | MON | MER | EN | M+
-----|-----|-----|-----|-----|-----
48SX D G
< 4 FIX 90 SWAP -
002 SWAP D DUP 5 SWAP -
SWAP 15 / →HMS 'α'
004 →TAG →STR "_h
" +
006 ROT →HMS 'δ' →TAG
→STR + "
008 " + SWAP -
D 15 / →HMS 't'
010 →TAG →STR + "_h
" +
012 > Bytes : 175
Checksum : # 9813h
    
```

# Sommer-/Winterzeit automatisch stellen

## von Günter Schapka

Jeder, der am 29.09.91 für nachts 2 Uhr einen Alarm zum Umstellen von Sommer- auf Winterzeit gesetzt hatte (zum Beispiel wie von Ralf Pfeifer schon vorgeschlagen), wird sich morgens sehr gewundert haben!

In der Tat war die Zeit verstellt, jedoch um kurz nach 9:00 Uhr bereits um 8 Stunden und nicht wie erwartet nur um eine Stunde.

Grund: Der Alarm war infolge der Zeitrückstellung immer aktiv geblieben und hat jede Stunde erneut um eine Stunde die Uhr zurückgestellt.

Diese Sache läßt sich folgendermaßen recht einfach beheben: Im Alarm für die Zeitumstellung wird ein Hilfsalarm gesetzt, der nach der Umstellung aktiv wird; er sucht zuerst

den Rückstellalarm und löscht ihn, so wie anschließend auch sich selbst und löscht sich.

Seine Fälligkeitszeit muß zwischen den beiden Umstellzeiten liegen, hier 10 Sekunden nach der Umstellung, nämlich um 1:0010 Uhr.

Die Stackanzeige "WIZ" beim nächsten Einschalten erinnert noch einmal daran, daß da etwas abgelaufen ist.

Das ist alles, ein Beispiel für 1992 siehe unten, der Übersicht halber wie ein Programm auseinandergezogen. Aktivieren mittels STOALARM.

Im übrigen braucht die Jahreszahl nicht angegeben zu werden, dann akzeptiert der HP 48 den Alarm im gerade laufenden Jahr.

sucht sich selbst, sucht Schaltalarm-Umschaltalarm

```

↓ Hilfslöscharm
{ 28.091992 2
" -29491200 CLKADJ
CLLCD { 28.091992
1.001
" { 28.091992 2 }
FINDALARM DELALARM
DROP { 28.091992
1.001 } FINDALARM
DELALARM DROP "WIZ"
" Meldung
0 } STOALARM SWAP
DROP OFF
"
0 }
    
```

GÜNTER SCHAPKA  
Rebusgasse 11  
D6100 Darmstadt



Datum jünger als das 2. Datum ist.

**DOW\$(T,M,J)**

gibt den Wochentag des Datums TT.MM.JJJJ im Format *MoDiMiDo FrSaSo* an.

**DOW(T,M,J)**

gibt den Wochentag des Datums TT.MM.JJJJ im Format 1-7 an.

**DATJUL(T,M,J,Z)**

gibt das julianische Datum des Datums TT.MM.JJJJ zum Zeitpunkt Z (dezimale Uhrzeit) bezogen auf den 0.5. Januar 1900 an.

*Beispiel:*

Julianisches Datum für den 29.1.1981, 1:42 = DATJUL(29,1,1981,HR(1.42)) = 29613.5708333.

Wird dazu der Wert 2415020 addiert, so erhält man das volle julianische Datum.

**JULDAT N,T,M,J,Z**

ist die Umkehrung von DATJUL. JULDAT von 29613.5708333,T,M,J,Z ergibt wieder den 29.1.1981, 1:42. Die Zeit muß allerdings gerundet werden, z.B. mit HM\$).

**Uhrzeit-Befehle**

**UHR**

Es wird eine laufende Uhr im Format HH.MM:SS Wo TT.MM.JJ angezeigt.

**UHR\$**

Es wird die momentane Uhrzeit und

das Datum im Format HH:MM:SS Wo TT.MM.JJ als String ausgegeben.

**UHRZ\$(Z)**

Die in Z gespeicherte Zahl wird im Format HH:MM wiedergegeben.

**Zeit und Winkelfunktionen**

**HR(Z)**

wandelt eine Zeit im Format HH.MMSSs.. in HH.hhh... um.

**HMS(Z)**

wandelt HH.hhh... in HH.MMSSs... um.

**HMS+(Z,Z1,Z2,...)**

addiert maximal 15 Zeiten im HH.MMmmm...-Format.

**HRHM(Z)**

wandelt eine Zeit im Format HH.MMmm... in HH.hhh... um.

**HMHR(Z)**

wandelt eine Zeit im Format HH.hhh... in HH.MMmmm... um.

**HM+(Z,Z1,Z2,...)**

addiert maximal 15 Zeiten im Format HH.MMmmm... um.

**HM\$(Z)**

Die dezimale Zeit in Z wird im Format HH:MM wiedergegeben.

**GH\$(Z,N,C)**

gibt eine Zeit (Winkel) Z im Sexagesimalformat aus.

N ist die Zeichennummer, die an

Stelle des Kommas zu stehen kommt.

C ist eine Code-Zahl, die das Rundungsformat bestimmt.

Dabei bedeuten:

- 2 = zweistellig = HH.MM
- 3 = dreistellig = HH.MMm
- 4 = vierstellig = HH.MMSS
- 5 = fünfstellig = HH.MMSSs

*Beispiel:*

GH\$(52.38833333,1,4) ergibt 52°26'18"

**Astronomische Werte**

**XSEKL(Z)**

gibt die Schiefe der Ekliptik zum Zeitpunkt Z (in DATJUL/36525) an.

XSEKL(DATJUL(1,1,1990,0)/36525) = 23.4405817883  
(Wert gemäß Astronomic Almanach, C1 = 23.44059164)

**XSTERNZ(N,Z)**

gibt die Sternzeit in Grad am julianischen Datum N zur Zeit Z an.

XSTERNZ(DATJUL(8,7,1990,0)/36525,HR(9.4430)) = 72.2101140804

Dividiert man diesen Wert durch 15, so erhält man die Zeit in dezimalen Stunden.

Im HMS-Format ergibt sich 4:48:50.4274 (Wert gemäß Astronomic Almanach, B7 = 4:48:50.4956).

```
calend2 L ID#5E
3078.00 Bytes

0123 4567 89AB CDEF ck

000: 3414 C454 E444 2302 FD
001: 802E 0050 9103 3019 4A
002: CE71 0E58 CFD0 0000 1A
003: F6E0 0000 0602 0000 35
004: 0077 00FF 0034 800F 14
005: E104 3200 DD20 EC80 E7
006: OFE3 OAE4 00DB 400F 58
007: 600F 6508 C600 FF50 DE
008: 22B0 OF86 ODAD 00F1 75
009: 70BF 010F A70C F010 BB
00A: F580 0801 OF09 005A 93
00B: 00F9 903F 010F 4A08 8C
00C: 4A00 FBA0 3D50 ODAB 48
```

```
00D: 0577 00F7 C00B F00F 8E
00E: 2D0B 1E00 FFD0 FCF0 65
00F: 0D8E 0211 10D7 F07F 4F
010: 210D 801A 6610 F511 8B
011: 5551 OFB4 4144 5A45 09
012: 5C48 CB44 1445 55D4 6A
013: 429C B441 4455 5D4B F9
014: 2ACD 4414 4555 D413 C4
015: 42BC 9441 4455 5D4C F7
016: C744 F475 42DC 544F A9
017: 475E C574 8442 FC58 6A
018: 4D44 20D5 84D4 B21D 57
019: 784D 4842 52D7 84D4 E0
01A: 35B2 3D58 4D43 54D7 06
01B: 8425 84D4 5D38 4256 07
01C: DBA4 55C4 4414 457D 81
01D: 9451 4745 4448 D755 FF
01E: 8425 429D 9558 425A 6E
01F: 542A D555 8425 BDB8 F6
```

```
020: 594E 4441 445C DD85 92
021: 94E4 A554 9445 DD98 9A
022: 5355 4B4C 4EDD 8535 CF
023: 4554 25E4 A5FD 1FF9 50
024: 6940 0011 B135 1121 05
025: CD13 78B6 C113 510B 1E
026: 3D24 23A3 C414 3402 F9
027: 15DD 008D 0745 09FF C5
028: FF72 1008 F681 F016 17
029: 18F6 81F0 1618 F681 5D
02A: F016 18F6 81F0 1361 95
02B: 0B1B E95F 2146 1361 92
02C: 8F15 278F 2EA2 1100 B9
02D: 18F1 5278 F2EA 2110 31
02E: 118F 1527 8F2E A211 AA
02F: 19AF 5118 AF70 48F4 74
030: 0331 1081 8F15 278F 97
031: 2EA2 1AF6 8FCD CE01 2E
032: 18A7 28F5 3331 1BE9 16
```

033: 5F21 4613 618F 100A CF 06C: 57E7 BED7 E1F7 96E7 0B 0A5: 7230 64FF 7610 8F04 76  
034: FBAF A8F2 23B1 1507 39 06D: FDD7 76FA FA78 BD8D 56 0A6: 9C08 DCA4 C078 106D 3B  
035: 18FA F48F 223B 1150 0A 06E: 84A8 08F1 9DB0 8F2E 9E 0A7: EF86 061A F22E 3012 E6  
036: 718F 1108 F223 B115 AF 06F: A211 0117 F8F1 9DB0 90 0A8: 0302 0186 OEEA F22E 29  
037: 0711 B134 7631 1CF7 A3 070: 8F2E A211 0217 F8F1 D9 0A9: 3062 0E60 11B1 09F2 12  
038: F017 B211 CF1C F710 2F 071: 9DB0 8D2E A217 6CF7 5B 0AA: AFC1 507A FC16 F150 1A  
039: 17D1 11CF 1CF1 CF70 68 072: 501A F230 5A7A 3078 5B 0AB: 7011 B129 F265 EF1B 10  
03A: F08D 84A8 020A F237 AC 073: FB7C E06C EE88 8337 BC 0AC: 109F 2AFD AF7A FDAF C1  
03B: 4666 6EEE 8166 3501 EA 074: 8DFA FA8F 223B 1AF6 B6 0AD: 6152 7AF8 16F1 5270 0B  
03C: 1B82 281E 832F 2203 98 075: 048D 832F 0888 3379 3A 0AE: 11B1 29F2 6CDF 8D66 B9  
03D: 12F9 62E2 12BA C712 EF 076: BFDA CC7A 20D4 F6FF 02 0AF: 3C08 DA91 E081 1840 B4  
03E: B94F 6064 A011 A135 88 077: 4496 FFD4 96FF 44F6 FA 0B0: 6900 8118 5084 1842 9B  
03F: 8548 DB2E 208F 1D63 65 078: FF64 27FF 3516 FF35 D1 0B1: 8F19 DB08 FFB6 C013 D1  
040: 056D 8FB E C206 6001 23 079: F6FF 07C4 C2C2 C284 7E 0B2: 610B 717F 721F 738F 9D  
041: 0B13 7135 10A8 F9DF 65 07A: 017F 8D36 6818 8884 55 0B3: 7A8F 7BBF 790F 8722 7F  
042: 3056 0603 0870 9F20 4A 07B: 48F1 9DB0 17F1 0073 E3 0B4: 3745 F75F E7D1 F709 17  
043: 11BA 065B 0831 6066 59 07C: 2FAF 6AF7 11AA F511 AC 0B5: F779 F7A5 F716 F7DD 5F  
044: EFA0 64B0 863B 1680 FE 07D: 18F4 0331 8F4B CE08 D7 0B6: E795 F7A8 F78D E79F CF  
045: 0873 2111 A135 8548 3A 07E: F223 B112 07D2 E8F8 A8 0B7: E75F E786 F7F6 F871 52  
046: D66E 2011 BAC7 AC2B B5 07F: A4C0 1188 F049 C08F 82 0B8: 0011 B136 D2E6 81E8 54  
047: F6AC B94A 50A4 E10B AC 080: 363C 0702 E8F0 49C0 EC 0B9: F499 C08D 832F 0888 00  
048: 2090 A606 73F8 D271 8E 081: 8FA9 1E02 08F4 99C0 20 0BA: 338F 19DB 08F2 EA21 D7  
049: 3015 371F 995F 2147 48 082: 6F2F 7DBE AF6A F711 F3 0BB: 2030 298E 50A8 E305 7E  
04A: 1371 5170 4208 D8F5 96 083: AAF5 1118 D403 3188 88 0BC: 98A5 0A8E 8408 5184 FD  
04B: F020 14A3 11F9 62B0 31 084: 8888 667A DF10 017F 3D 0BD: 2304 98E5 0852 1001 6C  
04C: 078D 84A8 0161 8F53 5C 085: 70DF 1108 409F A808 C2 0BE: 7F8F 19DB 08F3 22B1 34  
04D: AF08 F681 F08F 0B7F 9A 086: 50AF EB72 8F4B CE08 EB 0BF: 1361 B149 F214 816F A6  
04E: 01FE 95F2 1471 3701 30 087: F223 B186 050A 4CAF 0F 0C0: 1441 7F07 1097 10F1 9F  
04F: 20AF 2331 EEE8 1660 D7 088: 6048 D832 F080 1840 42 0C1: 1906 228F FC9C 0110 91  
050: 1F7F 708F 2B52 1AFA 8F 089: 94A4 18F1 9DB0 17F9 D3 0C2: A32A FA1B 5C8F 2154 4A  
051: 7A80 81CB F4BF 4AF2 4C 08A: 5850 8500 910B 725C 0F 0C3: 48F2 EA21 AF8A F210 A5  
052: 3408 1518 FB7C E0AF 30 08B: 11B0 AAF2 AEB8 F34D 73 0C4: A118 816D 210C 812A 52  
053: 68D5 3331 8F22 3B10 A8 08C: E0AE 98F3 4DE0 AE68 DB 0C5: FE04 8F1A 0B16 7106 DB  
054: 4645 F13D FF6A FFF2 45 08D: 60A0 BF2B F2D6 AF52 AF 0C6: 3106 F006 4206 3306 6C  
055: 0777 F79A FAFB AFA7 D8 08E: 0317 A860 5030 91B5 83 0C7: 540A FCAF 2AE6 109B 77  
056: 5DF7 B4F7 79FA F476 C5 08F: C8F2 8F33 8903 1E21 ED 0C8: F4BF 4100 6F30 AFCA B7  
057: CF7C 3F78 8F7A BF8D 37 090: B9C8 F214 C165 14C7 A3 0C9: F2A8 610A BF46 ADFA DA  
058: 84A8 0136 8FD3 DE01 8C 091: B8C3 45C8 F284 08D3 0C 0CA: FCAF 2AE6 10AB F4BF DA  
059: 378F D3DE 010C 0111 C6 092: 6681 8888 3484 0841 05 0CB: 464C FAF C AF2A 8610 10  
05A: C8FC 2DE0 1358 FC2D 8A 093: 80DF 8949 0850 6F10 F6 0CC: BBF4 6DDF 1120 5310 7D  
05B: E013 4017 ACF1 BE95 33 094: 208F 19DB 017F 958A EB 0CD: 6966 11AF 0102 111B 60  
05C: F214 6136 18F1 5276 93 095: E851 94C5 0850 2073 E4 0CE: 7410 1111 9661 1AF0 CC  
05D: ECF7 0EF7 E408 F439 C1 096: 8D10 317F 0477 1C11 A8 0CF: 1011 10B7 4100 1B14 C9  
05E: C08D 363C 076E F8F4 77 097: 1208 6031 8712 1BF0 E4 0D0: 9F20 414E 8F12 4B11 2F  
05F: F6C0 2205 8FFC 9C0A A2 098: ADOB F410 1692 0AF8 06 0D1: 14A7 A104 118A F5BF 5C  
060: FA8F 2EA2 1AF6 8FCD 1D 099: BF19 5DD1 3406 9109 A8 0D2: 5BF5 1B5C 8F21 4E90 52  
061: CE08 D533 312D AF23 16 09A: E690 3400 020A E2CA 1D 0D3: AC03 1D21 4C16 1979 0F  
062: 1422 0E60 1AF2 2836 13 09B: 1011 1BAF 711A AF51 AC 0D4: D031 FB8F 3389 0118 FD  
063: 5069 3962 0305 0171 1B 09C: 118F 4033 18F5 3331 49 0D5: AF53 11A8 F338 9011 93  
064: 9F8F 9D3D 08F4 F6C0 80 09D: 101A F910 AAFB 10B1 AB 0D6: C8FB 34B1 14C1 6111 63  
065: 8F79 6C08 F7E3 D08F B0 09E: B5C8 F220 113A F831 48 0D7: 9D53 11A8 F338 9011 87  
066: A91E 071B F8F6 34C0 87 09F: 1A8F 3389 031E 214C 6D 0D8: 4810 3039 06D1 31E2 B7  
067: 2205 8FFC 9C00 120A C0 0A0: 1611 12AF 8311 A8F3 10 0D9: 14C1 6111 AAF5 310A AF  
068: F235 26EE EE81 6618 44 0A1: 3890 31E2 14C1 6131 0D 0DA: 8F33 8903 1721 4C16 5D  
069: D5EB FF7E FFF8 F681 DE 0A2: 1A11 1AF8 BF09 5850 D5 0DB: 1114 8103 0398 6606 BF  
06A: F07C 0E7F 3FAF BAF A 42 0A3: 3038 F338 9064 DE8D 26 0DC: 7401 1AAF 5311 A8F3 4E  
06B: 748E 7AFD 7D2F AF47 A0 0A4: 4F6C 0792 08D6 34C0 77 0DD: 3890 1148 1030 4902 11

|                             |                             |                             |
|-----------------------------|-----------------------------|-----------------------------|
| ODE: D131 E214 C161 11BA 7B | 117: 8408 418C 6A9F 8888 B5 | 150: AC01 BE94 F214 ED72 B9 |
| ODF: F531 0A8F 3389 0312 51 | 118: 8888 8888 8882 F852 A1 | 151: 0310 2963 B131 B296 D1 |
| OE0: 214C 1613 1FF1 4C1B 74 | 119: 669F 8118 5085 2841 11 | 152: 3213 1D29 6360 68AC EC |
| OE1: 159F 2146 1343 45C8 E8 | 11A: 8CE6 9F81 1841 6DEF 88 | 153: A4C0 4101 1B29 4F28 39 |
| OE2: F284 0201 7F04 8D36 2E | 11B: 7000 14A2 0310 39E2 54 | 154: E690 0100 0420 3102 55 |
| OE3: 6818 1184 0852 8510 78 | 11C: 5130 99E6 DOBE OBF0 72 | 155: 8E2E 0006 1111 5171 2E |
| OE4: 7109 8E6C CF11 9062 A2 | 11D: 1610 18D0 29E0 D220 E2 | 156: 7F11 0151 711A 1368 1C |
| OE5: 211B 1368 FFC9 C010 35 | 11E: 31D0 8A20 OE68 A240 06 | 157: E75F E070 68FA B811 61 |
| OE6: 9B36 B36A FA8F 2EA2 1B | 11F: 0107 048D 84A8 0970 00 | 158: 17F8 5185 0852 1537 FF |
| OE7: 120A F231 0696 6903 43 | 120: FFEE 2FF8 E873 F1B5 C5 | 159: 8E18 5F11 B136 D2E6 51 |
| OE8: 104A 7ABF OBB4 1361 D3 | 121: C8F2 3F44 1647 57D6 B0 | 15A: 81E1 F995 F214 7137 F3 |
| OE9: OB04 1B5C 8F21 1994 78 | 122: F302 0315 4716 F3F0 34 | 15B: 8F49 9C0A FA8E 6DEE C0 |
| OEa: AC03 1D21 4C16 1814 57 | 123: 3E20 303E 2030 3021 07 | 15C: 8ECE EE07 068F AB81 5B |
| OEb: 9589 0810 6E10 3103 7E | 124: 5471 6F31 0214 C161 23 | 15D: 18EB BEE6 EEDA F08E D0 |
| OEc: 14C1 6181 095C B014 D8 | 125: 31FF 14C8 FBDO 318F 64 | 15E: CCBF 1618 E3CB FBF4 DF |
| OEd: C161 B24A F820 31FB 80 | 126: 814B 11B5 D8F2 8E8B C3 | 15F: BF48 F223 B1A3 CA3C 57 |
| OEe: 8F33 8901 8531 A314 2C | 127: 308E 2B30 8EA9 301F 51 | 160: 0120 1540 181B F615 D4 |
| OEf: C11B 1366 32F8 118F C5 | 128: 5C8F 2208 FEOC 1020 18 | 161: 4018 3BF6 2015 4018 08 |
| OF0: 19DB OB34 B348 FFB6 C4 | 129: 1B35 5F23 4784 EF14 0B | 162: 1BF6 1540 018E 8EFF D4 |
| OF1: C08E 62BF 228F FC9C BC | 12A: 48FF D151 8F99 C417 64 | 163: BF61 6701 AF1A E58F 9F |
| OF2: OAFA 8F2E A211 00BF BC | 12B: 92F5 4F8F A3A8 120A 6E | 164: E881 193E 908D D449 5A |
| OF3: 4BF4 1361 OB1B 5C8F FD | 12C: F21B E94F 214A 3102 B9 | 165: 0068 FD79 115F E070 DD |
| OF4: 2AF8 0420 311A 8F33 21 | 12D: 9664 3319 1D51 9A91 35 | 166: 1882 28F4 5DB0 AFE1 BC |
| OF5: 8903 1A31 4C16 1110 8D | 12E: 4A30 698E 8005 E504 FE | 167: OBAF 2203 E100 0026 A2 |
| OF6: D831 1A8F 3389 0638 02 | 12F: AFOA E4BF OBF0 1BA9 EE | 168: 8601 4051 8F23 4C0A 83 |
| OF7: F8F2 B521 AFAA F236 41 | 130: 4F26 0101 BA94 F28E BD | 169: F220 3E10 0000 0489 74 |
| OF8: 0000 3A28 FB7C E020 1D | 131: C9EF 8E69 EF1B 494F 1E | 16A: 0969 9755 1100 AF41 DA |
| OF9: AF23 05A7 A307 8FB7 C1 | 132: 28E9 8EF1 BE84 F28E 3B | 16B: 0120 AF23 E699 0000 4F |
| OFa: CE0D 47C1 0D4F 6449 26 | 133: C7EF BF4B F410 0203 07 | 16C: 0005 1783 1138 F234 4E |
| OFb: 6D49 644F 6642 7351 76 | 134: 1018 EFE2 0061 1015 EE | 16D: COAF 2203 E400 0039 EE |
| OFc: 635F 607C 4C2C 2136 6B | 135: 1720 8E44 2F8E C61F 73 | 16E: 8670 0063 7211 11B8 FE |
| OFd: 1462 4310 210A 8FBD A2 | 136: 0706 8FAB 811A F014 EF | 16F: E453 F119 AF71 188F 9E |
| OFe: 0311 09BF 6BD2 BF2B 2B | 137: B8E1 C1F8 E531 F070 C7 | 170: 363C 0100 AFC1 01AF A9 |
| OFF: F2AE B20A F531 7A1B AC | 138: 68FA B811 1577 BF6B 29 | 171: C744 08E8 43F8 F4F6 54 |
| 100: 5C8F 28F3 3890 3502 62 | 139: F615 57AF OAEA 8E69 F8 | 172: C073 308F A34C 08F7 9C |
| 101: 0202 1547 1651 1A15 E5 | 13A: 1F8E A01F 0706 8FAB 6E | 173: 96C0 111A F811 08FA 5F |
| 102: 4716 5119 8F23 4B1A C8 | 13B: 811A F015 B5BF 4BF4 CA | 174: 91E0 D2E6 81E8 F499 03 |
| 103: E58F 834B 1DA8 F42A 65 | 13C: 8E27 1F07 8F14 A110 EE | 175: C08D 832F 0AF2 2D31 16 |
| 104: 71BF OBB4 8F12 DE0A A2 | 13D: 48D8 4A80 20AF 2322 F9 | 176: 6320 3028 D049 C081 BA |
| 105: E4AF C317 A8F3 3890 F5 | 13E: EE81 68C8 20FB 8EEF F3 | 177: 18FD 8DB0 100A F220 6E |
| 106: 31E2 1851 4C18 514C 34 | 13F: 8EFF F136 1340 68F6 C0 | 178: 3E39 9000 0000 0030 34 |
| 107: 31A3 18F1 8514 C185 1D | 140: 81F0 1618 F681 F007 30 | 179: 58F2 34C0 AF22 03F4 E2 |
| 108: 14C1 8301 008F 7A51 0C | 141: 10A1 B5C8 F23F F4A5 9D | 17A: 9900 0000 0004 6197 1E |
| 109: 07CD E8F1 7510 2034 55 | 142: C202 A5A5 F6F3 1547 C8 | 17B: B401 188E D82F 20AF 4E |
| 10A: 5C8F 2840 048D 3668 36 | 143: 16F3 F020 303A 3030 B3 | 17C: 23F8 9900 0000 5210 B9 |
| 10B: 15C1 FF00 0007 2BE2 7F | 144: 3020 2154 716F 3903 44 | 17D: 3197 7201 188E 962F E7 |
| 10C: E315 1AF7 D320 345C B1 | 145: 03A3 0303 15C9 1693 7F | 17E: 20AF 23E1 0000 0049 90 |
| 10D: 8F28 FCB9 108F 2B52 6A | 146: 1FF1 4C1B 3E8F 2948 B1 | 17F: 2254 3274 0069 4F8F FF |
| 10E: 1AFA 81CA E0A7 4AF2 BC | 147: 9031 D214 CB34 B348 AD | 180: 049C 08D3 63C0 08      |
| 10F: 3200 2A7A 3058 FDO9 28 | 148: F2EA 211B DE8F 2AF6 5C |                             |
| 110: 218F 2C60 041B 8D84 DD | 149: 048E A610 8FBD 031B A7 |                             |
| 111: A808 8888 8888 8888 B1 | 14A: F6BF 61BF D8F2 8E05 6E |                             |
| 112: 882F 8428 5185 010A 6D | 14B: 101F 5C8F 2208 FEOC 49 |                             |
| 113: AF21 0810 91CF 17F7 5B | 14C: 1020 1B25 5F23 5022 F0 |                             |
| 114: D501 1B13 6119 AF71 73 | 14D: 9FF1 5C58 FFD1 518F 9A |                             |
| 115: 188F 663C 0AF9 1091 9C | 14E: 99C4 173F C54F 8FA3 B4 |                             |
| 116: 0011 AA4E 10A9 4EEC FB | 14F: A811 B0A4 F28E BD00 C9 |                             |

Wulf-Thorsten Gerdts  
Bunzlauer Weg 18  
D3002 Wedemark

Dr. Hanspeter Waldmann  
Schwendierweg 19  
D7959 Kirchberg

# Centronics-Schnittstelle

## Relais-Interface im Eigenbau von Peter Jochen

Rechner: alle PC mit Centronics-Ausgang, HP-71 mit IL-Konverter.

Wie bereits in PRISMA 5/6-90 beschrieben, ist der parallele Drucker-Port auch zum Ausgeben von Steuersignalen geeignet, die man nach Verstärkung zum Schalten eines oder mehrerer Relais verwenden kann.

Mit einem 8-Bit breiten Datenbus kann man prinzipiell 256 Kanäle ansteuern. Jedoch wollen wir uns hier auf maximal 8 Kanäle beschränken, weil dann eine zusätzliche Dekodier-Logik entfällt, und jede Datenleitung des Drucker-Ports einem Relais zugeordnet werden kann. Es werden dann nur wenige Bauteile (Aufwand etwa 10 DM) benötigt, wobei das teuerste Teil der 36-polige Centronics-Stecker ist.

Wenn man lediglich Zeitschalter nach Programm ein- und ausschalten will, ist hierzu der technische Aufwand denkbar einfach:

man benötigt pro Relais-Kanal ein D-Flipflop ("Latch") sowie nachfolgend einen Schalttransistor, der geeignet ist, ein Relais anzusteuern.

Das Prinzip zum Ansteuern einzelner Kanäle bzw. Daten-Bits ist folgendermaßen erklärt:

das 8-Bit-Datenwort z.B. 0000 0011<sub>bin</sub> = 3<sub>dez</sub> steht für die Dauer von ca. 10 µsec (siehe Text) am Parallel-Port an und setzt die Centronics-Pins Nr. 2 und 3 (d.h. Data-Bit 1 und 2) für diese Zeitdauer kurz auf "High".

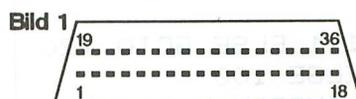
Wird an diese Pins je ein D-Flipflop angeschlossen, so läßt sich dieser Zustand an deren Ausgang speichern, bis das nächste Datenwort

gesendet wird.

Vom Centronics-Port (Bild 1) müssen nicht alle Pins beschaltet werden. Unbedingt erforderlich sind folgende Pins:

- 1 Data Strobe (invertiert)
- 2-9 Data 1,...,Data 8
- 19-29 Ground
- 11 Busy

| Pin | Bezeichnung                     | (E/A) |
|-----|---------------------------------|-------|
| 1   | DATA STROBE                     | (E)   |
| 2   | DATA 1                          | (E)   |
| 3   | DATA 2                          | (E)   |
| 4   | DATA 3                          | (E)   |
| 5   | DATA 4                          | (E)   |
| 6   | DATA 5                          | (E)   |
| 7   | DATA 6                          | (E)   |
| 8   | DATA 7                          | (E)   |
| 9   | DATA 8                          | (E)   |
| 10  | ACKNLG                          | (A)   |
| 11  | BUSY                            | (A)   |
| 12  | PE (Papierende)                 | (A)   |
| 13  | SLCT (Drucker angewählt)        | (A)   |
| 16  | 0V                              |       |
| 17  | Chassis Masse                   |       |
| 18  | +5V (vom Drucker)               |       |
| 19  | Masse, verdrillt mit Leitung 1  |       |
| 20  | Masse, verdrillt mit Leitung 2  |       |
| 21  | Masse, verdrillt mit Leitung 3  |       |
| 22  | Masse, verdrillt mit Leitung 4  |       |
| 23  | Masse, verdrillt mit Leitung 5  |       |
| 25  | Masse, verdrillt mit Leitung 6  |       |
| 25  | Masse, verdrillt mit Leitung 7  |       |
| 26  | Masse, verdrillt mit Leitung 8  |       |
| 27  | Masse, verdrillt mit Leitung 9  |       |
| 28  | Masse, verdrillt mit Leitung 10 |       |
| 29  | Masse, verdrillt mit Leitung 11 |       |



Einfacher Aufbau.

In Bild 2 ist ein Maximalaufbau für

einen Relais-Kanal zu sehen.

Die Einfachheit dieser Schaltung beruht auf einem Trick, den ich dem Buch von Friedeman [1] entnommen habe.

Danach wird der Busy-Eingang (11) des Centronics-Steckers auf "Low" (Ground) gelegt und signalisiert dem Computer, daß der Drucker bzw. das Interface ständig zum Empfang neuer Daten bereit ist.

Der alte Schaltzustand am Ausgang der D-Flipflops bleibt solange erhalten, bis ein neues Datenwort an den Port gesendet wird. Nach Bild 2 werden alle 8 Datenleitungen des Centronics-Ports an die Daten-Eingänge von IC1 gelegt.

IC1 ist ein Baustein, der 8 D-Flipflops enthält.

Diesen gibt es in der Version 74HC 373 (geeignet bis 5 mA Ausgangsstrom) und in der Version 74LS373, der nur mit ca. 2 mA belastet werden kann.

Bei Verwendung der letzteren Version muß der nachfolgende Transistor entsprechend höhere Verstärkung besitzen oder es sollte besser ein Darlington verwendet werden. Das Taktsignal des Drucker-Ports (Pin 1, Strobe) muß noch invertiert werden, bevor es den LE-Eingang (Latch enable) des IC1 erreicht.

Als Inverter eignet sich z.B. eine Einheit des CMOS-Bausteins 4069, in dem 6 gleichartige Inverter enthalten sind.

Da dieses Schalt-Interface nur wenige Bauteile enthält und für erste Versuche nur 1 oder 2 Kanäle benötigt werden, erübrigt sich ein Printplatten-Aufbau.

Ich habe einen Versuchsaufbau mit

einem Schaltrelais auf einer Lochrasterplatte der Größe 10 x 10 cm<sup>2</sup> bequem verdrahten können.

Zum Austesten von Steuerprogrammen oder auch zum Demonstrieren, welche Bits am Drucker-Port gesetzt sind, eignet sich ein Aufbau nach Bild 3.

Im Falle des HC-Bausteins können die Ausgänge den Strom für die Leuchtdioden (ca. 5 mA) direkt aufbringen (Bild 3a).

Steht nur die LS-Version zur Verfügung, muß der Ausgangsstrom über 4069-Inverter verstärkt werden (Bild 3b).

Wegen des Gegentaktausganges der Inverter sind die Leuchtdioden jetzt an den positiven Pol der Batterie zu legen.

Das Steuerwort kann hierdurch "bildlich" dargestellt werden, wenn die Leuchtdioden in einer Reihe angeordnet sind.

#### Ein Steuerprogramm

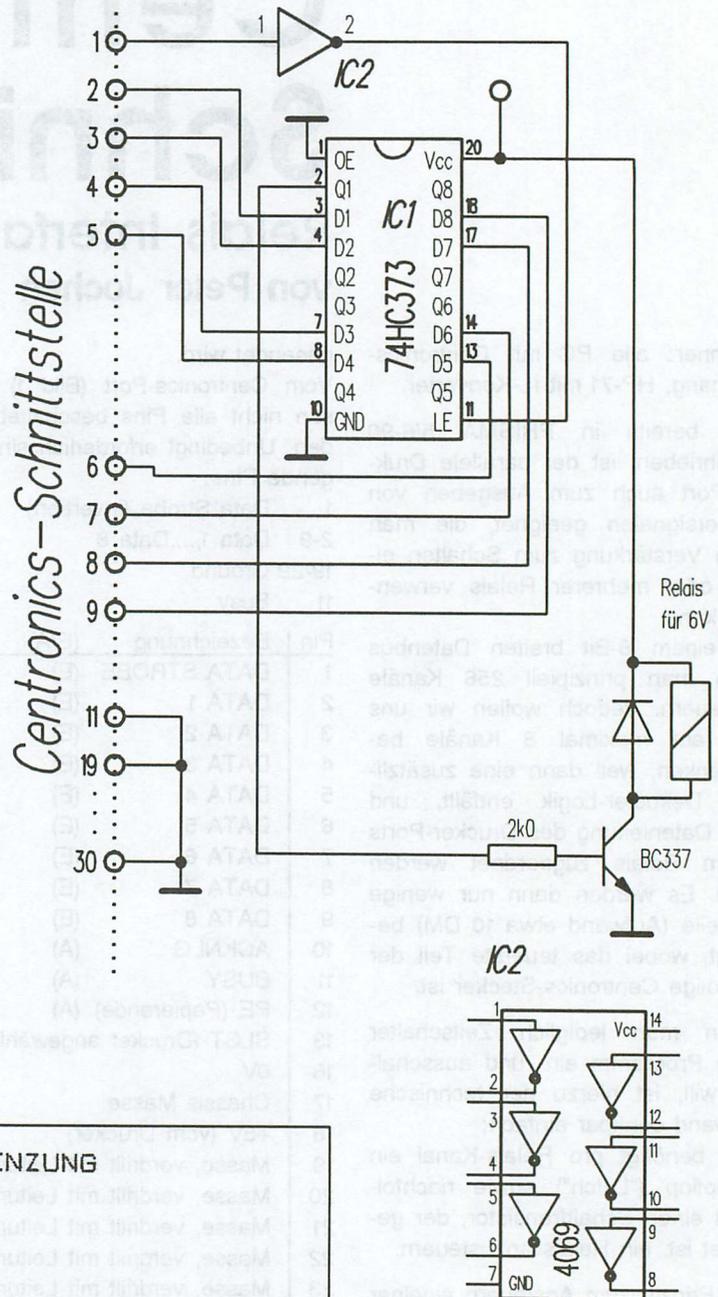
Zum Abschluß ist in Bild 4 ein kleines Steuerprogramm zum Betrieb eines Zeitschalters wiedergegeben. Nach dem Start des Programmes auf dem HP-71 mit RUN wird vom Programm die Einschaltdauer in Stunden abgefragt. In Zeile 93 wird Kanal 1 aktiviert und mit Zeile 150 nach Erreichen der vorgegebenen Dauer wieder abgeschaltet. Das Programm ist in HP-71 Basic geschrieben, kann aber leicht in andere Basic-Dialekte übersetzt werden.

```

10 ! LZ1: LADEZEIT-BEGRENZUNG
20 DESTROY ALL
30 FIX 1
60 L1=0
70 PRINT CHR$(0);
80 INPUT 'LADEZEIT IN H=';L9
90 L2=L9*60
92 BEEP
93 PRINT CHR$(1); ! K1 EINSCHALTEN
100 M2=VAL(TIME$(4,5))
110 DISP L1; ' VON ' ;L2; ' MINUTEN '
120 M1=VAL(TIME$(4,5))
130 IF M1<>M2 THEN L1=L1+1 ELSE GOTO 120
140 IF L1>=L2 THEN 150 ELSE 100
150 PRINT CHR$(0); ! K1 AUSSCHALTEN
155 BEEP
160 DISP 'AKKU GELADEN!'
170 END

```

Bild 4



#### Weiterführende Literatur:

- [1] G.Friedeman, "Control the world with HP-IL", Helder-mann-Verlag, Berlin. ISBN 0-9612174-9-9
- [2] W.Link, "Messen, Steuern und Regeln mit Basic", Franzis-Verlag, München. ISBN 3-7723-7411-5

Centronics-Schnittstelle

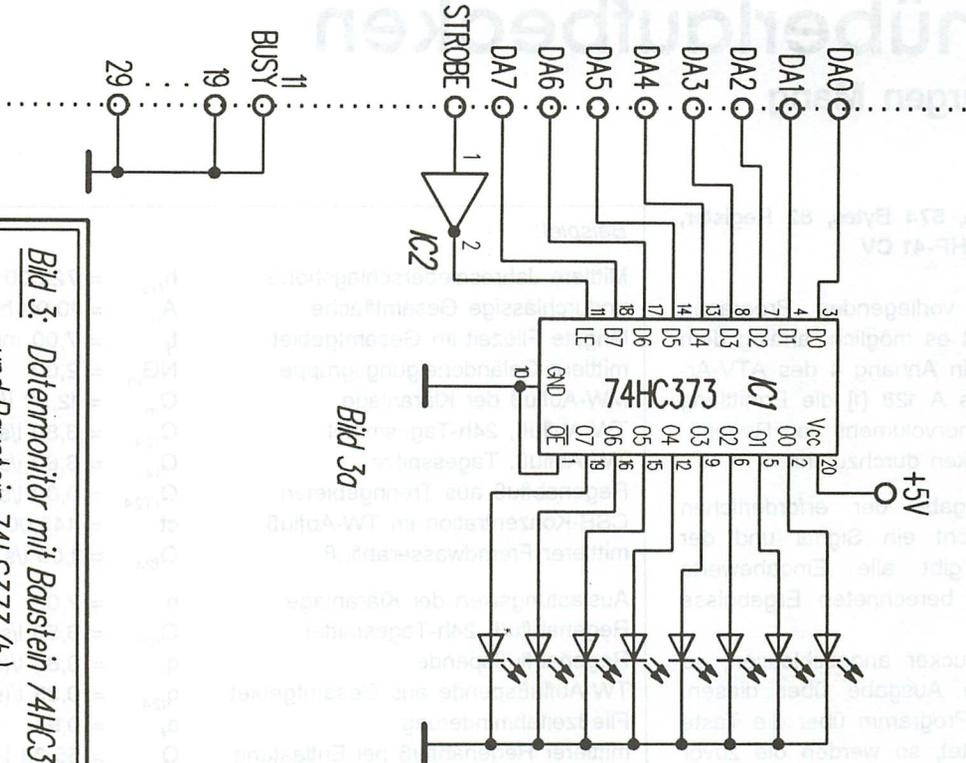


Bild 3a

Bild 3: Datenmonitor mit Baustein 74HC373 (a) und Baustein 74LS373 (b)

Centronics-Schnittstelle

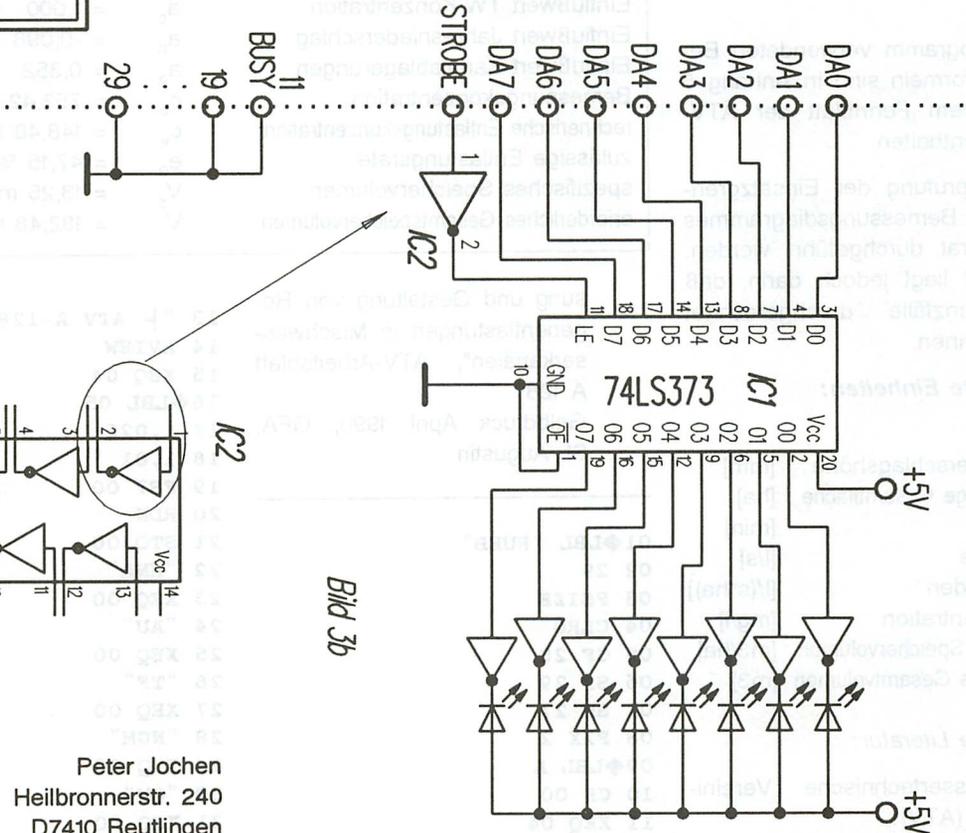
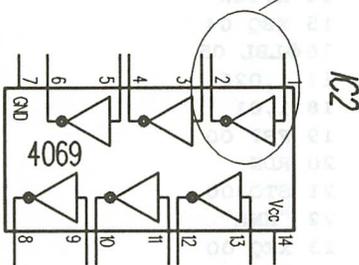


Bild 3b



Peter Jochen  
Heilbronnerstr. 240  
D7410 Reutlingen

# Bemessung von Regenüberlaufbecken

von Jürgen Mang

284 Zeilen, 574 Bytes, 82 Register,  
Size 029, HP-41 CV

Mit dem vorliegenden Programm "RUEB" ist es möglich, analog dem Formblatt in Anhang 4 des ATV-Arbeitsblattes A 128 [1] die Ermittlung des Speichervolumens von Regenüberlaufbecken durchzuführen.

Nach Eingabe der erforderlichen Werte ertönt ein Signal und der Rechner gibt alle Eingabewerte sowie die berechneten Ergebnisse aus.

Ist ein Drucker angeschlossen, so erfolgt die Ausgabe über diesen. Wird das Programm über die Taste "A" gestartet, so werden die zuvor eingegebenen Werte nicht gelöscht, sondern als Vorgabewerte angezeigt.

Die im Programm verwendeten Berechnungsformeln sind in Anhang 5 und auf dem Formblatt der ATV-Richtlinie enthalten.

Eine Überprüfung der Einsatzgrenzen des Bemessungsdiagrammes muß separat durchgeführt werden. Der Vorteil liegt jedoch darin, daß auch Grenzfälle durchgerechnet werden können.

## Verwendete Einheiten:

Mittlere

|                              |                      |
|------------------------------|----------------------|
| Jahresniederschlagshöhe      | [mm]                 |
| undurchlässige Gesamtfläche  | [ha]                 |
| Fließzeit                    | [min]                |
| Abflußwerte                  | [l/s]                |
| Abflußspenden                | [l/(s*ha)]           |
| CSB-Konzentration            | [mg/l]               |
| spezifisches Speichervolumen | [m <sup>3</sup> /ha] |
| erforderliches Gesamtvolumen | [m <sup>3</sup> ]    |

## Verwendete Literatur:

|   |                        |
|---|------------------------|
| [1] Abwassertechnische Vereinigung (ATV):   |                        |
| "Richtlinien für die Bemessung und Gestaltung von Regenentlastungen in Mischwasserkanälen", | ATV-Arbeitsblatt A 128 |

## Beispiel:

|                                       |                                     |
|---------------------------------------|-------------------------------------|
| Mittlere Jahresniederschlagshöhe      | $h_{Na} = 722,00 \text{ mm}$        |
| undurchlässige Gesamtfläche           | $A_U = 10,00 \text{ ha}$            |
| längste Fließzeit im Gesamtgebiet     | $t_f = 7,00 \text{ min}$            |
| mittlere Geländeneigungsgruppe        | $NG_m = 2,00$                       |
| MW-Abfluß der Kläranlage              | $Q_m = 12,30 \text{ l/s}$           |
| TW-Abfluß, 24h-Tagesmittel            | $Q_{t24} = 3,80 \text{ l/s}$        |
| TW-Abfluß, Tagesspitze                | $Q_{tx} = 6,60 \text{ l/s}$         |
| Regenabfluß aus Trenngebieten         | $Q_{rT24} = 0,00 \text{ l/s}$       |
| CSB-Konzentration im TW-Abfluß        | $ct = 443,00 \text{ mg/l}$          |
| mittlerer Fremdwasserabfluß           | $Q_{f24} = 1,00 \text{ l/s}$        |
| Auslastungswert der Kläranlage        | $n = 2,02$                          |
| Regenabfluß, 24h-Tagesmittel          | $Q_{r24} = 8,50 \text{ l/s}$        |
| Regenabflußspende                     | $q_r = 0,85 \text{ l/(s*ha)}$       |
| TW-Abflußspende aus Gesamtgebiet      | $q_{t24} = 0,38 \text{ l/(s*ha)}$   |
| Fließzeitabminderung                  | $a_f = 0,97$                        |
| mittlerer Regenabfluß bei Entlastung  | $Q_{re} = 55,33 \text{ l/s}$        |
| mittleres Mischverhältnis             | $m = 14,56$                         |
| xa-Wert für Kanalablagerungen         | $x_a = 13,82$                       |
| Einflußwert TW-Konzentration          | $a_c = 1,000$                       |
| Einflußwert Jahresniederschlag        | $a_h = -0,098$                      |
| Einflußwert Kanalablagerungen         | $a_a = 0,352$                       |
| Bemessungskonzentration               | $c_b = 752,42 \text{ mg/l}$         |
| rechnerische Entlastungskonzentration | $c_e = 148,48 \text{ mg/l}$         |
| zulässige Entlastungsrate             | $e_0 = 47,15 \%$                    |
| spezifisches Speichervolumen          | $V_s = 18,25 \text{ m}^3/\text{ha}$ |
| erforderliches Gesamtspeichervolumen  | $V = 182,48 \text{ m}^3$            |

sung und Gestaltung von Regenentlastungen in Mischwasserkanälen", ATV-Arbeitsblatt A 128

Gelbdruck April 1990, GFA, St. Augustin

|    |                  |
|----|------------------|
| 01 | ◆LBL "RUEB"      |
| 02 | 29               |
| 03 | PSIZE            |
| 04 | CLRG             |
| 05 | CF 28            |
| 06 | SF 29            |
| 07 | SF 27            |
| 08 | FIX 2            |
| 09 | ◆LBL A           |
| 10 | CF 00            |
| 11 | XEQ 04           |
| 12 | "RUEB-BEMESSUNG" |

|    |               |
|----|---------------|
| 13 | "  ATV A-128" |
| 14 | AVIEW         |
| 15 | XEQ 04        |
| 16 | ◆LBL 05       |
| 17 | 1,026         |
| 18 | 1,01          |
| 19 | FS? 00        |
| 20 | RDN           |
| 21 | STO 00        |
| 22 | "HNA"         |
| 23 | XEQ 00        |
| 24 | "AU"          |
| 25 | XEQ 00        |
| 26 | "TF"          |
| 27 | XEQ 00        |
| 28 | "NGM"         |
| 29 | XEQ 00        |
| 30 | "QM"          |
| 31 | XEQ 00        |
| 32 | "QT24"        |

|               |            |            |                |
|---------------|------------|------------|----------------|
| 33 XEQ 00     | 96 -       | 159 1      | 222 STO 27     |
| 34 "QTX"      | 97 /       | 160 -      | 223 36,8       |
| 35 XEQ 00     | 98 STO 11  | 161 STO 20 | 224 RCL 13     |
| 36 "QRT24"    | 99 RCL 05  | 162 RCL 04 | 225 13,5       |
| 37 XEQ 00     | 100 RCL 06 | 163 1      | 226 *          |
| 38 "CT"       | 101 RCL 08 | 164 -      | 227 +          |
| 39 ADV        | 102 +      | 165 2      | 228 RCL 13     |
| 40 XEQ 00     | 103 -      | 166 *      | 229 ,5         |
| 41 "QF24"     | 104 STO 12 | 167 1      | 230 +          |
| 42 XEQ 00     | 105 RCL 02 | 168 +      | 231 /          |
| 43 "N"        | 106 /      | 169 ,001   | 232 STO 28     |
| 44 XEQ 00     | 107 STO 13 | 170 *      | 233 RCL 27     |
| 45 "QR24"     | 108 RCL 06 | 171 430    | 234 RCL 24     |
| 46 XEQ 00     | 109 RCL 02 | 172 *      | 235 6          |
| 47 "QR"       | 110 /      | 173 RCL 14 | 236 +          |
| 48 XEQ 00     | 111 STO 14 | 174 ,45    | 237 /          |
| 49 "QT24"     | 112 50     | 175 Y^X    | 238 RCL 28     |
| 50 ADV        | 113 RCL 03 | 176 *      | 239 -          |
| 51 XEQ 00     | 114 100    | 177 2      | 240 3,84       |
| 52 "AF"       | 115 +      | 178 X<>Y   | 241 RCL 13     |
| 53 XEQ 00     | 116 /      | 179 -      | 242 *          |
| 54 "QRE"      | 117 ,5     | 180 10     | 243 3,6        |
| 55 XEQ 00     | 118 +      | 181 /      | 244 +          |
| 56 "M"        | 119 ,885   | 182 24     | 245 X<Y?       |
| 57 XEQ 00     | 120 X<Y?   | 183 RCL 18 | 246 RDN        |
| 58 "XA"       | 121 RDN    | 184 /      | 247 STO 25     |
| 59 XEQ 00     | 122 STO 15 | 185 X^2    | 248 RCL 02     |
| 60 FIX 3      | 123 RCL 13 | 186 *      | 249 *          |
| 61 "AC"       | 124 3,2    | 187 0      | 250 STO 26     |
| 62 XEQ 00     | 125 *      | 188 X<Y?   | 251 SF 00      |
| 63 "AH"       | 126 3      | 189 RDN    | 252 TONE 1     |
| 64 XEQ 00     | 127 +      | 190 STO 21 | 253 GTO 05     |
| 65 "AA"       | 128 *      | 191 RCL 20 | 254◆LBL 04     |
| 66 XEQ 00     | 129 RCL 02 | 192 RCL 19 | 255 FC? 21     |
| 67 FIX 2      | 130 *      | 193 +      | 256 RTN        |
| 68 "CB"       | 131 STO 16 | 194 +      | 257 SF 12      |
| 69 XEQ 00     | 132 RCL 08 | 195 600    | 258 "----"     |
| 70 "CE"       | 133 +      | 196 *      | 259 ACA        |
| 71 XEQ 00     | 134 RCL 06 | 197 STO 22 | 260 ACA        |
| 72 "E0"       | 135 /      | 198 107    | 261 ACA        |
| 73 XEQ 00     | 136 STO 17 | 199 RCL 17 | 262 ADV        |
| 74 "VS"       | 137 RCL 06 | 200 *      | 263 CF 12      |
| 75 ADV        | 138 24     | 201 +      | 264 RTN        |
| 76 XEQ 00     | 139 *      | 202 RCL 17 | 265◆LBL 06     |
| 77 SF 12      | 140 RCL 07 | 203 1      | 266 FS? 21     |
| 78 ADV        | 141 /      | 204 +      | 267 FMT        |
| 79 "V"        | 142 STO 18 | 205 /      | 268 RCL IND 00 |
| 80◆LBL 00     | 143 RCL 09 | 206 STO 23 | 269 "└ = "     |
| 81 FS? 00     | 144 600    | 207 70     | 270 ARCL X     |
| 82 GTO 06     | 145 /      | 208 -      | 271 FS? 21     |
| 83 RCL IND 00 | 146 1      | 209 3700   | 272 GTO 08     |
| 84 "└<"       | 147 X<Y?   | 210 X<>Y   | 273 PROMPT     |
| 85 ARCL X     | 148 RDN    | 211 /      | 274 ISG 00     |
| 86 "└>"       | 149 STO 19 | 212 STO 24 | 275 RTN        |
| 87 PROMPT     | 150 600    | 213 4000   | 276 GTO A      |
| 88 STO IND 00 | 151 RCL 01 | 214 25     | 277◆LBL 08     |
| 89 ISG 00     | 152 X<Y?   | 215 RCL 13 | 278 ACA        |
| 90 RTN        | 153 RDN    | 216 *      | 279 ADV        |
| 91 RCL 05     | 154 1000   | 217 +      | 280 ISG 00     |
| 92 RCL 10     | 155 X>Y?   | 218 RCL 13 | 281 RTN        |
| 93 -          | 156 RDN    | 219 ,551   | 282 STOP       |
| 94 RCL 07     | 157 800    | 220 +      | 283 GTO A      |
| 95 RCL 10     | 158 /      | 221 /      |                |

## RUEB

|  |  |
|--|--|
| Zeile 1 von RUEB (1-5) CCD-Barcodes Jürgen Mang      | Zeile 24 von RUEB (144-149) CCD-Barcodes Jürgen Mang |
| Zeile 2 von RUEB (6-6) CCD-Barcodes Jürgen Mang      | Zeile 25 von RUEB (150-160) CCD-Barcodes Jürgen Mang |
| Zeile 3 von RUEB (7-12) CCD-Barcodes Jürgen Mang     | Zeile 26 von RUEB (161-172) CCD-Barcodes Jürgen Mang |
| Zeile 4 von RUEB (13-22) CCD-Barcodes Jürgen Mang    | Zeile 27 von RUEB (173-184) CCD-Barcodes Jürgen Mang |
| Zeile 5 von RUEB (23-30) CCD-Barcodes Jürgen Mang    | Zeile 28 von RUEB (185-197) CCD-Barcodes Jürgen Mang |
| Zeile 6 von RUEB (31-34) CCD-Barcodes Jürgen Mang    | Zeile 29 von RUEB (198-207) CCD-Barcodes Jürgen Mang |
| Zeile 7 von RUEB (35-40) CCD-Barcodes Jürgen Mang    | Zeile 30 von RUEB (208-218) CCD-Barcodes Jürgen Mang |
| Zeile 8 von RUEB (41-45) CCD-Barcodes Jürgen Mang    | Zeile 31 von RUEB (219-229) CCD-Barcodes Jürgen Mang |
| Zeile 9 von RUEB (46-49) CCD-Barcodes Jürgen Mang    | Zeile 32 von RUEB (230-242) CCD-Barcodes Jürgen Mang |
| Zeile 10 von RUEB (50-60) CCD-Barcodes Jürgen Mang   | Zeile 33 von RUEB (243-255) CCD-Barcodes Jürgen Mang |
| Zeile 11 von RUEB (61-65) CCD-Barcodes Jürgen Mang   | Zeile 34 von RUEB (256-267) CCD-Barcodes Jürgen Mang |
| Zeile 12 von RUEB (66-69) CCD-Barcodes Jürgen Mang   | Zeile 35 von RUEB (268-279) CCD-Barcodes Jürgen Mang |
| Zeile 13 von RUEB (70-78) CCD-Barcodes Jürgen Mang   | Zeile 36 von RUEB (280-288) CCD-Barcodes Jürgen Mang |
| Zeile 14 von RUEB (79-82) CCD-Barcodes Jürgen Mang   | Zeile 37 von RUEB (289-296) CCD-Barcodes Jürgen Mang |
| Zeile 15 von RUEB (83-87) CCD-Barcodes Jürgen Mang   | Zeile 38 von RUEB (297-300) CCD-Barcodes Jürgen Mang |
| Zeile 16 von RUEB (88-92) CCD-Barcodes Jürgen Mang   | Zeile 39 von RUEB (301-305) CCD-Barcodes Jürgen Mang |
| Zeile 17 von RUEB (93-101) CCD-Barcodes Jürgen Mang  | Zeile 40 von RUEB (306-316) CCD-Barcodes Jürgen Mang |
| Zeile 18 von RUEB (102-106) CCD-Barcodes Jürgen Mang | Zeile 41 von RUEB (317-322) CCD-Barcodes Jürgen Mang |
| Zeile 19 von RUEB (107-115) CCD-Barcodes Jürgen Mang | Zeile 42 von RUEB (323-327) CCD-Barcodes Jürgen Mang |
| Zeile 20 von RUEB (116-116) CCD-Barcodes Jürgen Mang | Zeile 43 von RUEB (328-331) CCD-Barcodes Jürgen Mang |
| Zeile 21 von RUEB (117-123) CCD-Barcodes Jürgen Mang | Zeile 44 von RUEB (332-339) CCD-Barcodes Jürgen Mang |
| Zeile 22 von RUEB (124-134) CCD-Barcodes Jürgen Mang | Zeile 45 von RUEB (340-341) CCD-Barcodes Jürgen Mang |
| Zeile 23 von RUEB (135-143) CCD-Barcodes Jürgen Mang |  |

Jürgen Mang  
Lindener Straße 7  
D6751 Krickenbach

# Belastungsglieder und Volleinspannmomente

von Jürgen Mang

389 Zeilen, 617 Bytes, 89 Register,  
Size 032, HP-41 CV, X-F/M

Das vorliegende Programm "BV" ermittelt für ein beidseitig eingespannten Träger mit beliebiger Belastung die Belastungsglieder sowie die Volleinspannmomente. Als Grundlage diente das Programm "Belastungsglieder und Volleinspannmomente" aus der UPLE, der Autor war Thomas Meyer.

Das vorliegende Programm wurde um die Lastfälle Feldmoment, Stützensenkung und Temperaturbelastung erweitert. Bei mehreren Lasten werden die Ergebnisse aufsummiert und am Programmende angezeigt.

## Verwendete Einheiten

- Elastizitätsmodul [m4]
- Temperaturdifferenz [°K o. °C]
- Temperaturkoeffizient [1/°K]
- Trägerlängen und Trägerhöhen [m]
- Lasten [kN bzw. kN/m]
- Momente [kNm]
- Stützensenkung [mm]

## Tastenbelegung

### Taste Funktion

- a Neustart des Programms
- A Belastung im Feld (Streckenlasten, Einzellasten und Momente).  
Wenn a2=0 dann 2 Möglichkeiten:
- B Einzellast
- C Moment (linksdrehend ist positiv!)
- D Stützensenkung
- E Ausgabe der Ergebnisse

## Programmbeispiel

- Elastizitätsmodul =  $2,1 \cdot 10^8 \text{ kN/m}^2$
- Bauteilhöhe = 0,40 m
- Trägheitsmoment =  $0,0025 \text{ m}^4$
- Temperaturkoeffizient =  $1,0 \cdot 10^{-5} \text{ 1/}^\circ\text{K}$
- TU-TO =  $-3,0 \text{ }^\circ\text{K}$

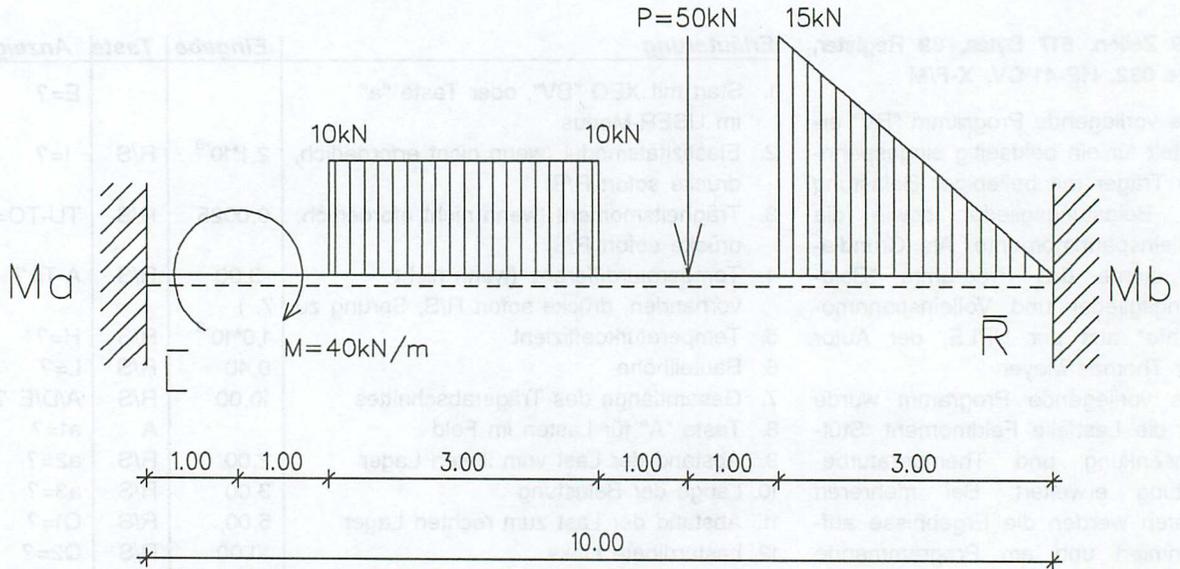
| Erläuterung   | Eingabe             | Taste | Anzeige   |
|---|---------------------|-------|-----------|
| 1. Start mit XEQ "BV", oder Taste "a" im USER-Modus                             |                     |       | E=?       |
| 2. Elastizitätsmodul (wenn nicht erforderlich, drücke sofort R/S)               | $2,1 \cdot 10^8$    | R/S   | I=?       |
| 3. Trägheitsmoment (wenn nicht erforderlich, drücke sofort R/S)                 | 0,0025              | R/S   | TU-TO=?   |
| 4. Temperaturdifferenz (wenn nicht vorhanden, drücke sofort R/S, Sprung zu 7. ) | -3,00               | R/S   | A-T=?     |
| 5. Temperaturkoeffizient  | $1,0 \cdot 10^{-5}$ | R/S   | H=?       |
| 6. Bauteilhöhe  | 0,40                | R/S   | L=?       |
| 7. Gesamtlänge des Trägerabschnittes  | 10,00               | R/S   | A/D/E ?   |
| 8. Taste "A" für Lasten im Feld   |                     | A     | a1=?      |
| 9. Abstand der Last vom linken Lager  | 2,00                | R/S   | a2=?      |
| 10. Länge der Belastung   | 3,00                | R/S   | a3=?      |
| 11. Abstand der Last zum rechten Lager  | 5,00                | R/S   | Q1=?      |
| 12. Lastordinate Links  | 10,00               | R/S   | Q2=?      |
| 13. Lastordinate Rechts   | 10,00               | R/S   | A/D/E ?   |
| 14. Taste "A" für Lasten im Feld  |                     | A     | a1=?      |
| 15. Abstand der Last vom linken Lager   | 7,00                | R/S   | a2=?      |
| 16. Länge der Belastung   | 3,00                | R/S   | a3=?      |
| 17. Abstand der Last zum rechten Lager  | 0,00                | R/S   | Q1=?      |
| 18. Lastordinate Links  | 15,00               | R/S   | Q2=?      |
| 19. Lastordinate Rechts   | 0,00                | R/S   | A/D/E ?   |
| 20. Taste "A" für Lasten im Feld  |                     | A     | a1=?      |
| 21. Abstand der Last vom linken Lager   | 1,00                | R/S   | a2=?      |
| 22. Länge der Belastung   | 0,00                | R/S   | a3=?      |
| 23. Abstand der Last zum rechten Lager  | 9,00                | R/S   | B/C ?     |
| 24. Taste "C" für Moment  |                     | C     | M=?       |
| 25. Größe des Momentes (linksdrehend ist positiv definiert)                     | -40,00              | R/S   | A/D/E ?   |
| 26. Taste "A" für Lasten im Feld  |                     | A     | a1=?      |
| 27. Abstand der Last vom linken Lager   | 6,00                | R/S   | a2=?      |
| 28. Länge der Belastung   | 0,00                | R/S   | a3=?      |
| 29. Abstand der Last zum rechten Lager  | 4,00                | R/S   | B/C ?     |
| 30. Taste "B" für Einzellast  |                     | B     | P=?       |
| 31. Größe der Einzellast  | 50,00               | R/S   | A/D/E ?   |
| 32. Taste "D" für Stützensenkung  |                     | D     | a<MM>=?   |
| 33. Stützensenkung Links  | 4,00                | R/S   | b<MM>=?   |
| 34. Stützensenkung Rechts (Da Hebung, Eingabe negativ)                          | -1,00               | R/S   | A/D/E ?   |
| 35. Taste "E" für Ergebnisausgabe   |                     | E     |           |
| 36. Belastungsglied Links   |                     |       | L=100,37  |
| 37. Belastungsglied Rechts  |                     | R/S   | R=422,01  |
| 38. Einspannmoment Links  |                     | R/S   | Ma=73,75  |
| 39. Einspannmoment Rechts   |                     | R/S   | Mb=-247,9 |
| 40. Durch nochmaliges Drücken von "R/S" kann wieder neu gestartet werden.       |                     |       |           |

**Abschließende Bemerkungen:**

- Wenn der Träger nur auf einer Seite eingespannt ist, so muß über die Belastungsglieder die Endeinganspannung berechnung werden.
- Ein Kontrollausdruck ist nicht vorgesehen.
- Wenn kein X/F-Modul vorhanden ist, so müssen die Zeilen 23 und 24 entfernt und stattdessen die Register 1 bis 13 mit der Zahl "0" belegt werden.

- Die Berechnungsformeln stammen aus der Zeitschrift "Die Bautechnik", Ausgabe 2/1981, Seite 61/62 und K.-J. Schneider: Bautabellen, 8. Auflage, Seite 4.27, Werner-Verlag

Jürgen Mang  
Lindener Str. 7  
6751 Krickenbach



```

01◆LBL "BV"          33 STO 21          65 RCL 00          97 RCL 00
02 32                34 "a2=?"        66 1                98 RCL 01
03 PSIZE             35 PROMPT        67 +                99 -
04uLBL a             36 STO 02        68 X<>Y            100 X^2
05 CLRG              37 X=0?         69 -                101 ,2
06 SF 27             38 XEQ 01        70 1                102 *
07 CF 00             39 "a3=?"        71 RCL 00           103 +
08 "E=?"            40 PROMPT        72 -                104 2
09 PROMPT            41 STO 22        73 *                105 /
10 STO 16            42 ST+ 01        74 3                106 STO 06
11 "I=?"            43 RDN           75 RCL 00           107 RCL 00
12 PROMPT            44 RCL 01        76 -                108 2
13 STO 23            45 +            77 RCL 01           109 *
14 "TU-TO=?"        46 ST+ 00        78 -                110 RCL 01
15 PROMPT            47 +            79 RCL 01           111 +
16 STO 18            48 ST* 02        80 *                112 RCL 05
17 X≠0?             49 ST/ 00        81 RCL 00           113 *
18 XEQ 06            50 ST/ 01        82 *                114 RCL 00
19 "L=?"            51 FS? 00        83 +                115 RCL 06
20 PROMPT            52 GTO 03        84 RCL 00           116 *
21 STO 20            53 "Q1=?"        85 RCL 01           117 -
22◆LBL 05            54 PROMPT        86 -                118 RCL 03
23 0,01301          55 STO 03        87 X^2              119 *
24 CLRGX             56 "Q2=?"        88 ,7              120 2
25 RDN               57 PROMPT        89 *                121 RCL 01
26 CF 00             58 STO 04        90 +                122 *
27 "A/D/E ?"        59◆LBL 00        91 6                123 RCL 00
28 AVIEW             60 RCL 00        92 /                124 +
29 STOP              61 RCL 01        93 STO 05           125 RCL 05
30◆LBL A             62 +            94 RCL 00           126 *
31 "a1=?"           63 RCL 01        95 RCL 01           127 RCL 01
32 PROMPT            64 *            96 *                128 RCL 06
    
```

|            |             |               |             |
|------------|-------------|---------------|-------------|
| 129 *      | 194◆LBL 01  | 259 RCL 30    | 324 *       |
| 130 -      | 195 RDN     | 260 X^2       | 325 RCL 16  |
| 131 RCL 04 | 196 ,01     | 261 3         | 326 *       |
| 132 *      | 197 ST- 01  | 262 *         | 327 6       |
| 133 +      | 198 ST- Y   | 263 1         | 328 *       |
| 134 RCL 02 | 199 RDN     | 264 X<>Y      | 329 RCL 20  |
| 135 *      | 200 ,02     | 265 -         | 330 X^2     |
| 136 FS? 00 | 201 STO 02  | 266 RCL 19    | 331 /       |
| 137 XEQ 02 | 202 SF 00   | 267 CHS       | 332 ST+ 15  |
| 138 ST+ 14 | 203 RTN     | 268 *         | 333 ST+ 26  |
| 139 ST+ 28 | 204uLBL 02  | 269 ST+ 15    | 334 CHS     |
| 140 RCL 06 | 205 ,02     | 270 3         | 335 ST+ 14  |
| 141 RCL 00 | 206 1/X     | 271 RCL 30    | 336 ST+ 27  |
| 142 RCL 01 | 207 *       | 272 *         | 337 GTO 05  |
| 143 +      | 208 RTN     | 273 1         | 338◆LBL 09  |
| 144 1      | 209◆LBL 03  | 274 -         | 339 -3      |
| 145 -      | 210 XEQ 10  | 275 RCL 31    | 340 /       |
| 146 2      | 211 "B/C ?" | 276 *         | 341 ST+ 26  |
| 147 /      | 212 AVIEW   | 277 RCL 19    | 342 ST+ 27  |
| 148 -      | 213 STOP    | 278 CHS       | 343 RTN     |
| 149 STO 06 | 214◆LBL B   | 279 *         | 344◆LBL 10  |
| 150 3      | 215 "P=?"   | 280 ST+ 26    | 345 RCL 21  |
| 151 RCL 00 | 216 PROMPT  | 281 3         | 346 RCL 20  |
| 152 2      | 217 STO 25  | 282 RCL 31    | 347 /       |
| 153 *      | 218 RCL 31  | 283 *         | 348 STO 30  |
| 154 RCL 01 | 219 1       | 284 1         | 349 RCL 22  |
| 155 +      | 220 +       | 285 -         | 350 RCL 20  |
| 156 -      | 221 RCL 25  | 286 RCL 30    | 351 /       |
| 157 RCL 05 | 222 *       | 287 *         | 352 STO 31  |
| 158 *      | 223 RCL 21  | 288 RCL 19    | 353 RTN     |
| 159 1      | 224 *       | 289 *         | 354◆LBL E   |
| 160 RCL 00 | 225 RCL 22  | 290 ST+ 27    | 355 "L="    |
| 161 -      | 226 *       | 291 GTO 05    | 356 ARCL 14 |
| 162 RCL 06 | 227 RCL 20  | 292◆LBL 06    | 357 AVIEW   |
| 163 *      | 228 /       | 293 "A-T=?"   | 358 STOP    |
| 164 -      | 229 ST+ 14  | 294 PROMPT    | 359 "R="    |
| 165 RCL 03 | 230 ST+ 28  | 295 RCL 16    | 360 ARCL 15 |
| 166 *      | 231 RCL 30  | 296 *         | 361 AVIEW   |
| 167 2      | 232 1       | 297 RCL 18    | 362 STOP    |
| 168 RCL 01 | 233 +       | 298 *         | 363 RCL 28  |
| 169 *      | 234 RCL 21  | 299 "H=?"     | 364 2       |
| 170 RCL 00 | 235 *       | 300 PROMPT    | 365 *       |
| 171 +      | 236 RCL 22  | 301 /         | 366 RCL 29  |
| 172 3      | 237 *       | 302 RCL 23    | 367 -       |
| 173 X<>Y   | 238 RCL 25  | 303 *         | 368 -3      |
| 174 -      | 239 *       | 304 3         | 369 /       |
| 175 RCL 05 | 240 RCL 20  | 305 *         | 370 RCL 26  |
| 176 *      | 241 /       | 306 ST+ 14    | 371 +       |
| 177 1      | 242 ST+ 15  | 307 ST+ 15    | 372 "Ma="   |
| 178 RCL 01 | 243 ST+ 29  | 308 XEQ 09    | 373 ARCL X  |
| 179 -      | 244 GTO 05  | 309 RTN       | 374 AVIEW   |
| 180 RCL 06 | 245◆LBL C   | 310◆LBL D     | 375 STOP    |
| 181 *      | 246 "M=?"   | 311 "a<MM>=?" | 376 RCL 29  |
| 182 -      | 247 PROMPT  | 312 PROMPT    | 377 2       |
| 183 RCL 04 | 248 STO 19  | 313 1000      | 378 *       |
| 184 *      | 249 RCL 31  | 314 /         | 379 RCL 28  |
| 185 +      | 250 X^2     | 315 STO 24    | 380 -       |
| 186 RCL 02 | 251 3       | 316 "b<MM>=?" | 381 -3      |
| 187 *      | 252 *       | 317 PROMPT    | 382 /       |
| 188 FS? 00 | 253 1       | 318 1000      | 383 RCL 27  |
| 189 XEQ 02 | 254 X<>Y    | 319 /         | 384 +       |
| 190 ST+ 15 | 255 -       | 320 RCL 24    | 385 "Mb="   |
| 191 ST+ 29 | 256 RCL 19  | 321 X<>Y      | 386 ARCL X  |
| 192 GTO 05 | 257 *       | 322 -         | 387 AVIEW   |
| 193 RTN    | 258 ST+ 14  | 323 RCL 23    | 388 STOP    |

## BV

Zeile 1 von BV (1-5) CCD-Barcodes Jürgen Mang



Zeile 2 von BV (6-11) CCD-Barcodes Jürgen Mang



Zeile 3 von BV (12-14) CCD-Barcodes Jürgen Mang



Zeile 4 von BV (15-21) CCD-Barcodes Jürgen Mang



Zeile 5 von BV (22-26) CCD-Barcodes Jürgen Mang



Zeile 6 von BV (27-31) CCD-Barcodes Jürgen Mang



Zeile 7 von BV (32-38) CCD-Barcodes Jürgen Mang



Zeile 8 von BV (39-45) CCD-Barcodes Jürgen Mang



Zeile 9 von BV (46-58) CCD-Barcodes Jürgen Mang



Zeile 10 von BV (59-66) CCD-Barcodes Jürgen Mang



Zeile 11 von BV (67-77) CCD-Barcodes Jürgen Mang



Zeile 12 von BV (78-88) CCD-Barcodes Jürgen Mang



Zeile 13 von BV (89-96) CCD-Barcodes Jürgen Mang



Zeile 14 von BV (97-107) CCD-Barcodes Jürgen Mang



Zeile 15 von BV (108-119) CCD-Barcodes Jürgen Mang



Zeile 16 von BV (120-132) CCD-Barcodes Jürgen Mang



Zeile 17 von BV (133-144) CCD-Barcodes Jürgen Mang



Zeile 18 von BV (145-152) CCD-Barcodes Jürgen Mang



Zeile 19 von BV (153-165) CCD-Barcodes Jürgen Mang



Zeile 20 von BV (166-178) CCD-Barcodes Jürgen Mang



Zeile 21 von BV (179-191) CCD-Barcodes Jürgen Mang



Zeile 22 von BV (192-199) CCD-Barcodes Jürgen Mang



Zeile 23 von BV (200-207) CCD-Barcodes Jürgen Mang



Zeile 24 von BV (208-215) CCD-Barcodes Jürgen Mang



Zeile 25 von BV (216-221) CCD-Barcodes Jürgen Mang



Zeile 26 von BV (222-228) CCD-Barcodes Jürgen Mang



Zeile 27 von BV (229-238) CCD-Barcodes Jürgen Mang



Zeile 28 von BV (239-247) CCD-Barcodes Jürgen Mang



Zeile 29 von BV (248-254) CCD-Barcodes Jürgen Mang



Zeile 30 von BV (255-255) CCD-Barcodes Jürgen Mang



Zeile 31 von BV (256-264) CCD-Barcodes Jürgen Mang



Zeile 32 von BV (265-273) CCD-Barcodes Jürgen Mang



Zeile 33 von BV (274-283) CCD-Barcodes Jürgen Mang



Zeile 34 von BV (284-292) CCD-Barcodes Jürgen Mang



Zeile 35 von BV (293-295) CCD-Barcodes Jürgen Mang



Zeile 36 von BV (296-303) CCD-Barcodes Jürgen Mang



Zeile 37 von BV (304-309) CCD-Barcodes Jürgen Mang



Zeile 38 von BV (310-311) CCD-Barcodes Jürgen Mang



Zeile 39 von BV (312-315) CCD-Barcodes Jürgen Mang



Zeile 40 von BV (316-326) CCD-Barcodes Jürgen Mang



Zeile 41 von BV (327-335) CCD-Barcodes Jürgen Mang



Zeile 42 von BV (336-341) CCD-Barcodes Jürgen Mang



Zeile 43 von BV (342-350) CCD-Barcodes Jürgen Mang



Zeile 44 von BV (351-354) CCD-Barcodes Jürgen Mang



Zeile 45 von BV (355-363) CCD-Barcodes Jürgen Mang



Zeile 46 von BV (364-372) CCD-Barcodes Jürgen Mang



Zeile 47 von BV (373-377) CCD-Barcodes Jürgen Mang



Zeile 48 von BV (378-381) CCD-Barcodes Jürgen Mang



# HP41 Utilities

von Dr. Martin Hochenegger

## Simulierung von Befehlen, wenn das entsprechende Modul fehlt

HP 41, Drucker

ACAXY ist im CCD-Modul enthalten:

A-Register....Text

X-Register....zu druckende Zahl

Y-Register....Druckbreite

Nach Ausführung des Programmes steht die Druckbreite im X-Register, die ausgedruckte Zahl ist gelöscht.

Wird Flag 20 gesetzt, so werden Text und Zahl durch .... verbunden.

Brot..... 3,60

Milch..... 0,91

```

01◆LBL "ACAXY"
02 X<>Y
03 STO Z
04 ACA
05 ALENG
06 -
07 CLA
08 ARCL Y
09 ALENG
10 -
11 FC? 20
12 32
13 FS? 20
14 46
15◆LBL 00
16 ACCHR
17 DSE Y
18 GTO 00
19 X<> Z
20 ACX
21 RCL T
22 END
  
```

ACLX ist ebenfalls im CCD-Modul enthalten xxx,yyy im X-Register heißt, daß Byte yyy wird xxx-mal in den Druckerspeicher gebracht.

5,049

XEQ "ACLX"

PRBUF bedeutet: .....

Im Gegensatz zum ACLX des

CCD-Moduls kann der X-Wert auch größer als 99 sein.

```

01◆LBL "ACLX"
02 ENTER^
03 INT
04 X<>Y
05 FRC
06 E3
07 *
08 CLA
09 XTOA
10◆LBL 00
11 ACA
12 DSE Y
13 GTO 00
14 END
  
```

## BIN

Das Advantage-Modul erlaubt durch die Funktion BININ die Eingabe der gesetzten und nicht gesetzten Bytes einer Dualzahl. Der Dezimalwert wird ausgegeben.

Im Programm BIN werden nach ED (Zeile 7) die gesetzten Bytes als 1, die nicht gesetzten als 0 eingegeben. Führende 0-Bytes brauchen nicht eingegeben zu werden. Zu beachten ist, daß eine 1 als gesetztes Byte erkannt wird, während alle anderen Zeichen als 0 interpretiert werden.

Nach der Eingabe kein R/S, sondern ON drücken, um aus dem ED-Modus heraus zu kommen. Die Dezimalzahl steht dann im X-Register.

```

01◆LBL "BIN"
02 FIX 0
03 "Y"
  
```

```

04 2
05 CRFLAS
06 AOFF
07 ED
08 .
09 STO 01
10 SEEKPT
11 GETREC
12 ALENG
13 STO 00
14◆LBL 00
15 49
16 ATOX
17 X=Y?
18 XEQ 01
19 DSE 00
20 GTO 00
21 RCL 01
22 "Y"
23 PURFL
24 STOP
25◆LBL 01
26 2
27 RCL 00
28 E
29 -
30 Y^X
31 ST+ 01
32 END
  
```

BIN1 und BIN2 erlauben es, wie ACOL von E. Wohler, PRISMA 85.4.19, die Positionen der gesetzten Bytes in beliebiger Reihenfolge einzugeben.

```

01◆LBL "BIN1"
02 .
03 STO 00
04 FIX 0
05 SF 28
06 "POS="
  
```

| LABEL               | BIN       | BIN1    | BIN2 | ACOL |
|---------------------|-----------|---------|------|------|
| Erforderl. Hardware | HP-41 CY  | EXT I/O | -    | -    |
| Zeitbedarf ca. [s]  | 10        | 10      | 7    | 10   |
| PPL                 | 56        | 56      | 43   | 60   |
| Wiederholung        | XEQ "BIN" | R/S     | R/S  | R/S  |
| SIZE                | ≥2        | ≥2      | ≥1   | ≥0   |

07 PROMPT  
 08 CLA  
 09 ARCL X  
 10 ALENG  
 11 STO 01  
 12 E  
 13 -  
 14 LASTX  
 15◆LBL 01  
 16 "┌,"  
 17 AROT  
 18 DSE Y  
 19 GTO 01  
 20◆LBL 02  
 21 2  
 22 ANUMDEL  
 23 E

24 -  
 25 Y^X  
 26 ST+ 00  
 27 DSE 01  
 28 GTO 02  
 29 RCL 00  
 30 END

01◆LBL "BIN2"  
 02 FIX 0  
 03 .  
 04 STO 00  
 05 "POS="  
 06 PROMPT  
 07◆LBL 00  
 08 E1  
 09 /

10 ENTER^  
 11 FRC  
 12 E1  
 13 \*  
 14 2  
 15 X<>Y  
 16 E  
 17 -  
 18 Y^X  
 19 ST+ 00  
 20 X<>Y  
 21 INT  
 22 X=0?  
 23 GTO 00  
 24 RCL 00  
 25 END

### Anderer Drucker - anderer Zeichensatz

HP41 CX, CCD, EXT I/O (bei Programmänderung durch AROT nicht nötig), Drucker, SIZE ≥ 008

Umwandlung der Umlaute und ß von einem Zeichensatz in einen anderen.

Beim Kauf eines Druckers ist es (meist) nötig, den File zu durchforsten und die Bytes der Umlaute und des zu ändern. Dieses Programm ist dazu gedacht, den ASCII-Zeichensatz, in den von ISO-German zu verwandeln. In den geradzahligen Zeilen 2 bis 14 ist das erste Byte das von ASCII, das zweite das von ISO-German. Die F2 in der Auflistung der synthetischen Zeilen kann man vergessen.

| DRUCKER |    |     |     |
|---------|----|-----|-----|
|         | IL | TJ  | DJ  |
| ä       | 22 | 204 | 123 |
| ö       | 24 | 206 | 124 |
| ü       | 26 | 207 | 125 |
| Ä       | 21 | 216 | 91  |
| Ö       | 23 | 218 | 92  |
| Ü       | 25 | 219 | 93  |
| ß       | 5  | 222 | 126 |

01◆LBL "Ae-UeΣ"  
 02 "" siehe Tabelle  
 03 ASTO 01  
 04 "" siehe Tabelle  
 05 ASTO 02  
 06 "" siehe Tabelle  
 07 ASTO 03  
 08 "" siehe Tabelle

09 ASTO 04  
 10 "" siehe Tabelle  
 11 ASTO 05  
 12 "" siehe Tabelle  
 13 ASTO 06  
 14 "" siehe Tabelle  
 15 ASTO 07  
 16 1.007  
 17 STO 00  
 18◆LBL 00  
 19 .  
 20 SEEKPT  
 21◆LBL 01  
 22 CLA  
 23 ARCL IND 00  
 24 ABSP  
 25◆LBL 02  
 26 POSFL  
 27 X<0?  
 28 GTO 03  
 29 E  
 30 DELCHR  
 31 CLA  
 32 ARCL IND 00  
 33 ATOXL  
 34 INSCHR  
 35 GTO 01  
 36◆LBL 03  
 37 ISG 00  
 38 GTO 00  
 39 END

Tabelle der verwendeten Strings:

002: F2 CC 7B  
 004: F2 CE 7C  
 006: F2 CF 7D  
 008: F2 D8 5B  
 010: F2 DA 5C  
 012: F2 DB 5D  
 014: F2 DE 7E

### Files im Extended Memory

#### HP 41 CX, CCD RENFL (REName File)

Angeregt durch die Programme von A. Jakob und O. Roth (Lit. 1) habe ich diese so zusammengeschnitten, daß ein kurzes und bündiges Programm herauskam, das nur zur Umbenennung des ersten Files dient.

01◆LBL "RENFL"  
 02 SF 27  
 03 191  
 04 "NEW NAME="  
 05 PMTA  
 06 "┌" "  
 07 7  
 08 AROT  
 09 CLX  
 10 RCL [  
 11 POKER  
 12 END

#### CLXM (CLear X-Mem)

Zum Löschen des gesamten X-Memories sind verschiedene Methoden bekannt (Lit. 2-4). Nach einer solchen Löschung ist für den Benutzer das XMEM leer, jedoch nicht für den Rechner !

Durch Umbenennung des ersten Files kommt der gesamte Inhalt wieder zum Vorschein, gleich, was man in das Alpha-Register eingibt. Das in Lit. 4) angegebene Programm zur Löschung scheint das kürzeste und bequemste zu sein. Es kann auch durchgetastet werden:

**64 ENTER CLX POKER.**

Ich habe es noch ein wenig gekürzt:

```
64 . POKER.
01◆LBL "CLXM"
02 64
03 .
04 POKER
05 END
```

Literatur:

- 1) O. Roth, PRISMA 86.5.18; A. Jakob, PRISMA 86.6.25
- 2) K.Jarret, Erweiterte Funktionen des HP 41 leicht gemacht. Deutsche Ausgabe von H. Dalkowski, Helder- mann Verlag, Berlin, 1989, S. 149.
- 3) A. McConack K. Jarret, HP 41 Advanced Programming Tips,

Synthetix, Berkeley, 1987, p. 86/87.

- 4) S. Fegert, H. Dalkowski, Eine Programmsammlung für den HP 41, Helder- mann Verlag, Berlin, 1989, S. 34.

**Synthetisches PPC Siegel  
HP 41, CCD oder Barcodeleser, IL-Drucker**

Das Programm in J. S. Dearing, Tips und Ticks und Routinen für den Taschenrechner der Serie HP 41, S. 146, Helder- mann Verlag, Berlin 1984, ist falsch. Die amerika- nische Ausgabe Calculator Tips & Routines, Corvallis Software, INC, 1981, p.89, hat den selben Fehler, obwohl deren Programmausdruck mit meinem übereinstimmt. Der Rechner bleibt bei jenem Programm

auf Zeile 5 mit der Fehlermeldung DATA ERROR stehen.

Das Programm muß richtig heißen (ohne die beiden Textzeilen):

```
01◆LBL "LG"
02 "" siehe Tabelle
03 "" siehe Tabelle
04 X<> ]
05 ACSPEC
06 X<> \
07 ACSPEC
08 X<> [
09 ACSPEC
10 PRBUF
11 END
```

*Tabelle der verwendeten Strings:*

002: 17 194 228 124 60 122 241 17  
102 62 30 61 120 249  
003: 127 17 158 29 155 191 78 135

**ACAXY**

Zeile 1 von ACAXY (1-4) CCD-Barcodes Dr.M. Hochenegger

Zeile 2 von ACAXY (5-11) CCD-Barcodes Dr.M. Hochenegger

Zeile 3 von ACAXY (12-20) CCD-Barcodes Dr.M. Hochenegger

Zeile 4 von ACAXY (21-27) CCD-Barcodes Dr.M. Hochenegger

**ACLX**

Zeile 1 von ACLX (1-6) CCD-Barcodes Dr.M. Hochenegger

Zeile 2 von ACLX (7-14) CCD-Barcodes Dr.M. Hochenegger

Zeile 3 von ACLX (15-14) CCD-Barcodes Dr.M. Hochenegger

**BIN**

Zeile 1 von BIN (1-5) CCD-Barcodes Dr.M. Hochenegger

Zeile 2 von BIN (6-13) CCD-Barcodes Dr.M. Hochenegger

Zeile 3 von BIN (14-20) CCD-Barcodes Dr.M. Hochenegger

Zeile 4 von BIN (21-31) CCD-Barcodes Dr.M. Hochenegger

Zeile 5 von BIN (32-32) CCD-Barcodes Dr.M. Hochenegger

**BIN1**

Zeile 1 von BIN1 (1-5) CCD-Barcodes Dr.M. Hochenegger

Zeile 2 von BIN1 (6-11) CCD-Barcodes Dr.M. Hochenegger

Zeile 3 von BIN1 (12-18) CCD-Barcodes Dr.M. Hochenegger

Zeile 4 von BIN1 (19-27) CCD-Barcodes Dr.M. Hochenegger

Zeile 5 von BIN1 (28-32) CCD-Barcodes Dr.M. Hochenegger

**Ae-Ues**

Zeile 1 von AE UES (1-2) CCD-Barcodes Dr.M. Hochenegger

Zeile 2 von AE UES (3-8) CCD-Barcodes Dr.M. Hochenegger

Zeile 3 von AE UES (9-13) CCD-Barcodes Dr.M. Hochenegger

Zeile 4 von AE UES (14-18) CCD-Barcodes Dr.M. Hochenegger

Zeile 5 von AE UES (19-27) CCD-Barcodes Dr.M. Hochenegger

Zeile 6 von AE UES (28-35) CCD-Barcodes Dr.M. Hochenegger

Zeile 7 von AE UES (36-39) CCD-Barcodes Dr.M. Hochenegger

**BIN2**

Zeile 1 von BIN2 (1-5) CCD-Barcodes Dr.M. Hochenegger



Zeile 2 von BIN2 (6-12) CCD-Barcodes Dr.M. Hochenegger



Zeile 3 von BIN2 (13-23) CCD-Barcodes Dr.M. Hochenegger



Zeile 4 von BIN2 (24-27) CCD-Barcodes Dr.M. Hochenegger


**RENFL**

Zeile 1 von RENFL (1-3) CCD-Barcodes Dr.M. Hochenegger



Zeile 2 von RENFL (4-5) CCD-Barcodes Dr.M. Hochenegger



Zeile 3 von RENFL (6-9) CCD-Barcodes Dr.M. Hochenegger



Zeile 4 von RENFL (10-15) CCD-Barcodes Dr.M. Hochenegger


**CLXM**

Zeile 1 von CLXM (1-3) CCD-Barcodes Dr.M. Hochenegger



Zeile 2 von CLXM (4-5) CCD-Barcodes Dr.M. Hochenegger


**LG**

Zeile 1 von LG (1-2) CCD-Barcodes Dr.M. Hochenegger



Zeile 2 von LG (3-3) CCD-Barcodes Dr.M. Hochenegger



Zeile 3 von LG (4-8) CCD-Barcodes Dr.M. Hochenegger



Zeile 4 von LG (9-14) CCD-Barcodes Dr.M. Hochenegger



Dr. Martin Hochenegger  
Heidelberger Landstraße 97  
D6100 Darmstadt 13

## Korrektur zu Maschine&Sprache Teil II PRISMA 4-6/92

Mir sind beim Abtippen des Mammartikels von Georg Hoppen leider einige scheinbar unwichtige Tippfehler unterlaufen, die bei dem einen oder anderen Interessierten zum gefürchteten MEMORY LOST geführt haben.

Also vor Maschinenspracheexperimenten immer einen Backup in einem Port oder auf dem nächstliegenden Rechner machen! - Grundsätzlich fehlt der Hinweis, daß alle Bytes-Angaben um 4 zu erhöhen sind, da die letzten 4 Ziffern ja noch mitzählen.

Seite 54, 3. Spalte:  
Es muß #1F55Dh heißen.

Seite 56, 1. Spalte unten:  
In dem Programmlisting waren 3 Fehler, in der 3. Zeile war ein \* zuviel, in der 4. Zeile muß es statt x \* heißen, in der 5. Zeile fehlt vor SUB noch ein SWAP, siehe aber das korrigierte Listing.

Seite 58, 2. Spalte oben:  
Die Zahlenkette lautet korrekt "D9D20E1632DBBF193632B2130"

```
« "String ?" { α } INPUT
"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
» »
```

Vor dem letzten B fehlte eine 2.

Bei der Montage des Programmes Blackjack sind leider einige Unterprogramme im wahrsten Sinne des Wortes unter den Tisch gefallen.

Am besten schicken Interessen einen SAFU mit einer formatierten 720kB 31/2" Diskette an mich oder Georg Hoppen zwecks Überspielung des Programms, zum Abtippen ist es so wieso fast zu lang.

mm

**Buchbesprechung****Mathe & PC**

von Irmgard Wagner und  
Werner "Tiki" Küstenmacher

ISBN 3-89390-342-9

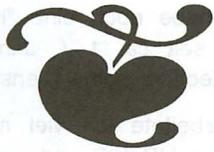
Der Untertitel "Ein heiterer Cartoon-Mathematik-Kurs" sagt schon fast alles über dieses Büchlein.

Die spielerische Aufmachung täuscht etwas über die Ernsthaftigkeit des Themas hinweg. Zumindest für Leute, die nicht täglich mit höheren Rechenarten zu tun haben, dürfte der zweifellos vorhandene Lerneffekt nützlich sein.

Der Streifzug durch die Mathewelt beginnt bei den Grundrechenarten und endet bei der Näherungsrechnung. Dazwischen liegen solch nützliche Tierchen wie das "Klapustri" oder das "Klapopustri" als Merkhilfen für die Reihenfolge der Rechnung mit verschiedenen Operatoren.

Empfehlenswert für Leute, die sich für alles andere als Mathegenies halten, mit einem dicken Wermutstropfen: Der Preis von DM 24,80 erscheint mir für ein Paperback mit 160 Seiten viel zu hoch.

dw



# Master Mind

## mit Druck- und Anzeigefunktionen und Bruchrechnen

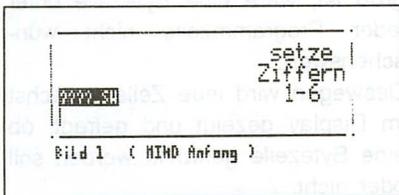
von Wolfgang Führer

Ich fange am besten mit dem an, womit alles angefangen hat, nämlich dem Master Mind Programm.

Um den Taschenrechner auch mal als Zeitvertreib benutzen zu können, habe ich das Programm von Peter Röhl (Prisma 89.04.34) abgetippt.

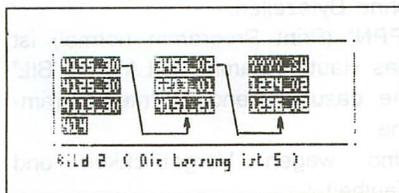
Leider ist das Spielen ohne zusätzlichen Zettel fast nicht möglich, da nur der momentane Zug angezeigt wird und damit der notwendige Rückblick auf die vorherigen Züge fehlt. Also habe ich eine Anzeigeroutine geschrieben, die kleinere Ziffern und ein 3-geteiltes Bildschirmscrolling verwendet und somit die letzten 9 Züge in der Anzeige verbleiben.

Im folgenden sieht man den Spielbeginn.



Das Eingabefeld liegt unter den invertierten ???\_SW, wobei die Fragezeichen die Eingabefelder für Zahlen markieren und SW die Spalten für schwarze und weiße Steine anzeigen.

Bild 2 zeigt einen Spielverlauf kurz vor der Lösung.



Die kleinen Ziffern erhält das Pro-

gramm aus der Liste 'ANZ'. Sie wird durch das Programm 'QANZ' erzeugt, welches nach dem aufrufen gelöscht werden kann.

Die Unterprogramme 'INFO', 'RANDO', 'CHAR', 'BLWI', 'CONV' und 'ROW' können Ihr auch direkt in 'MIND' eingeben, allerdings wird eine nachträgliche Änderung oder Fehlersuche (bei der Gesamtlänge) zur Qual.

'ZDROP' habe ich ebenfalls als Programm ausgelagert, da ich die Befehlsfolge auch anderweitig benötige. Gibt man einen String in Ebene 2 und eine Ziffer in Ebene 1 ein, so entfernt 'ZDROP' das angegebene Zeichen aus dem String.

Gestartet wird das Programm durch den Aufruf von 'MIND' und erklärt sich ansonsten von selbst.

Zu erwähnen bliebe noch, daß man durch Drücken der Taste "E" 2 Sekunden lang das Ergebnis sehen kann (für die ganz verzweifelten Momente - 'CHAR' Zeilen 10 - 13).

Wer einen Anzeigenausdruck wünscht, kann die Taste "L" drücken und erhält den Anzeigenausdruck mit Rahmen (durch 'PRL' + 'PRLR' - 'CHAR' Zeilen 14 - 17).

'PRL' kann auch durch PRLCD ersetzt werden.

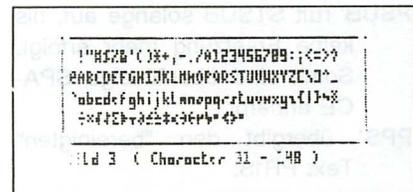
Diese Sache hatte mich dann auf den Geschmack von kleinen Zeichen gebracht.

Da es mich schon immer gereizt hat die Darstellungsmöglichkeiten der Anzeige bzw. des Druckers auszureizen, habe ich mich vom Positionieren und Vergrößern an das Verkleinern gemacht.

Ich habe einen Zeichensatz entwor-

fen, der 4 Pixel breit, 7 Pixel hoch (mit Unterlängen) und einigermaßen lesbar ist. Nach ca. 1 Woche hatte ich mich soweit an die Schrift gewöhnt, daß ich sie der Übersicht wegen bei Programmausdrucken nicht mehr missen möchte.

Da die gesamte Druckerbreite ausgenutzt wird, ist die Darstellung von 41 Zeichen pro Zeile möglich. Die Schrift sieht mit den Charaktern 31 bis 148 wie folgt aus:



Der Buchstabe N stellt leider eine Notlösung dar, weil ich ihn nicht anders von H zu unterscheiden wußte.

Aber daran, sowie an das Q, kann man sich gewöhnen.

Die Arbeit fängt damit an, 'QANF' abzutippen und aufzurufen und damit die Liste 'QDIS' (957.5 Bytes) zu erzeugen.

Erste Eindrücke bekommt man, wenn man 'SML' abgetippt hat. 'SML' macht aus einem String in Ebene 1 einen Graphikstring, der mit →LCD angezeigt werden kann. Nun sollte man sich alle Zeichen mit diesem Befehl anzeigen lassen und überprüfen.

Wenn alles in Ordnung ist (und man den Zeichensatz auch nicht ändern will) kann 'QANF' jetzt gelöscht werden.

'PR1S' soll PR1 ersetzen und ist deswegen etwas aufwendiger, weil

hier keine Worte durch das Zeilenende zerschnitten werden.

Man kann also irgendeinen String oder Programm-Recall erzeugen und mit PR1S ausdrucken lassen, ohne sich um die Zeilenlänge kümmern zu müssen.

Ist Flag 30 gesetzt, wird die strukturierte Anzeige unterdrückt und es erfolgt ein kompakterer Ausdruck, der Druckerpapier spart. Allerdings bleiben dann "Löcher" im Text, die von den führenden Leerzeichen beim Einrücken herrühren.

Um diese Löcher zu beseitigen kann man 'PPS' aufrufen.

'PPS' ruft seinerseits 'PSUB' auf, welches wieder 'STSUB' (von Ralf Pfeifer, Prisma 89.04.32) aufruft.

Im Groben sieht das folgendermaßen aus:

'STSUB' ersetzt in einem Text (in Ebene 1) die Zeichenfolge "CR SPACE SPACE" (Ebene 3) durch "CR SPACE".

'PSUB' ruft STSUB solange auf, bis keine Ersetzung mehr erfolgt. Somit sind überflüssige SPACE entfernt.

'PPS' übergibt den "bereinigten" Text PR1S.

Und weil die Bequemlichkeit siegt noch folgendes:

Alle auszudruckenden Programme innerhalb eines Ordners können mit ' begrenzt auf den Stack geschoben werden, da 'PPS' den Programmausdruck solange wiederholt, wie sich ein Objekt des Typs 6 ('Name') auf dem Stack befindet.

Und weil man auch schnell den notwendigen Flagzustand von Flag 30 vergißt als Krönung 'PPSN' (Print Programm Small Normal) und 'PPSC' (Print Programm Small Condensed).

An die Rechneranzeige habe ich natürlich auch gedacht und so bringt 'SDIS' den kleinen Zeichensatz auch auf das Display. (=34 Zeichen pro Zeile).

Der Befehl 'NULL', der aufgerufen wird, ist dabei ein String aus 137 CHR 0, der unter dem Namen 'NULL' abgelegt ist.

Also 0 CHR und 8 mal DUP +,

dann 1 137 SUB und 'NULL' STO um das zu erzeugen.

Und damit nichts vergessen wird, hier noch mal kurz eine Hierarchie. (Ein bißchen Ordnung im Kopf kann jetzt bestimmt nicht schaden.)



Hier fallen noch die Programme 'I' und 'OSIZE' auf.

'I' liefert mir das Datum als Ausdruckkopf.

'I' kann man also einfach weglassen oder etwas anderes dafür einsetzen.

'OSIZE' liefert die Länge des Objektes in Ebene 1 als reelle Zahl und ist das Programm von Peter Röhl (Prisma 90.02.39).

Und für die Leute, die erst mal sehen wollen worauf sie sich einlassen, hier 'SML' (normal 12 Zeilen) mit 'PPSN' (7 Zeilen) und 'PPSC' (4 Zeilen) ausgedruckt.

```

mit PPSN :
ZBS =====12.03.1992
'SML'
< FAST "" 1 3 PICK SIZE
FOR X OVER X X SUB MUN 148 N1H
IF DUP 10 ==
THEN 22 +
END 31 MAX 30 - QDIS SHAP GET + 0
CHR +
NEXT SHAP DROP SLOW > 183.5 BgLez

mit PPSC :
ZBS =====12.03.1992
'SML'
< FAST "" 1 3 PICK SIZE FOR X OVER X X
SUB MUN 148 N1H IF DUP 10 == THEN 22 +
END 31 MAX 30 - QDIS SHAP GET + 0 CHR +
NEXT SHAP DROP SLOW > 183.5 BgLez
  
```

Damit wäre nun das Kapitel Kleindruck beendet.

Aber ich habe noch eine "Kleinigkeit", die seit ca. 1 1/2 Jahren in meinem Rechner seine Dienste tut.

Damals arbeitete ich viel mit der Graphik des HP-28S und legte viele Graphikelemente und Symbole in Form von Strings in Programmen ab und konnte so einen schnellen Bildaufbau erreichen.

Auch mühsam, bitweise zusammengestellte Symbole wurden so abgelegt und es entstand der Wunsch, solche Bytefolgen auch auf dem Programmausdruck festhalten zu lassen und nicht durch gesonderte Listen mit Werten.

Ich will es kurz machen.

Mir ist es also gelungen mit einer Matrix von 5 mal 7 Punkten alle Zahlen zwischen 0 und 255 (theoretisch bis 299) darzustellen.

Damit war es möglich genau unter einer normal ausgedruckten Zeile eine Zeile mit den entsprechenden Bytewerten zu drucken.

Als Gerüst diente das Ausdruckprogramm 'PRP' von Ralf Pfeiffer im Prisma 89.04.36.

Da die Anzahl der Graphikstrings in einem Programm meist nicht sehr groß ist, wäre eine Bytezeile unter jeder Programmzeile nicht wünschenswert.

Deswegen wird jede Zeile zunächst im Display gezeigt und gefragt, ob eine Bytezeile gedruckt werden soll oder nicht.

Das ist bei langen Programmen zwar etwas mühsam, aber dadurch akzeptabel, daß man das Programm auch für einen normalen Ausdruck benutzen kann.

Ist Flag 1 gesetzt, so erfolgt die Bytezeilenabfrage, ansonsten druckt das Programm ohne zu fragen ohne Bytezeilen.

'PPN' (Print Programm normal) ist das Hautprogramm, 'BILA' und 'BIL' die dazugehörigen Unterprogramme.

Und wegen Vergeßlichkeit und Faulheit :

'PPNN' Print Programm Normal Normal

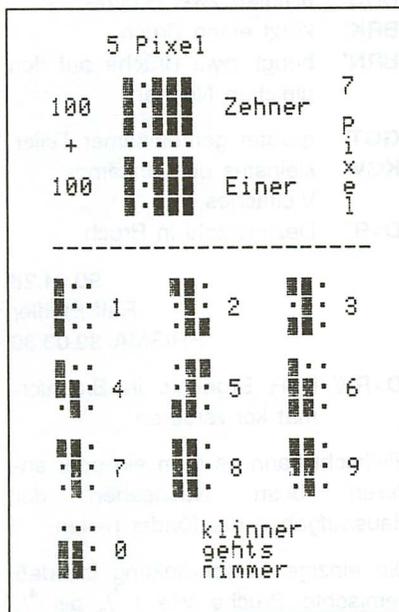
'PPNB' Print Programm Normal Bytezeilen und zur Erinnerung  
 'PPSN' Print Programm Small Normal  
 'PPSC' Print Programm Small Condensed

Damit dürfte man dann hinsichtlich seiner Ausdrücke (fast) wunschlos glücklich sein. Ich bin es jedenfalls.

Vielleicht wundert sich der ein oder andere über den Weg, den der Ausdruck nimmt. Zuerst wird die zu druckende Zeile mit 1 DISP in die Anzeige gebracht und anschließend mit LCD→ 1 137 SUB als Graphikstring isoliert und ausgedruckt. Das ist zwar nicht sonderlich schnell, aber so können Druckersteuerzeichen, die im Programm als Strings abgelegt sind, den Drucker nicht durcheinander bringen.

'IDISP' (von Peter Röhl, 90.02.39) in 'PPN' erzeugt eine invertierte Anzeige und kann durch DISP ersetzt werden.

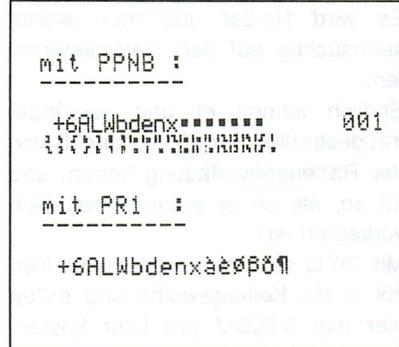
Aber jetzt noch eine kleine Zahlenkunde, um beim Anblick der Hieroglyphen auch zu einem AHA!-Erlebnis zu kommen. Die Matrix ist folgendermaßen aufgeteilt:



Obwohl die Programmausdrücke in der PRISMA manchmal etwas schwach auf der Brust sind, soll trotzdem eine Leseprobe folgen. (Dies liegt an der oft geringen Schwärzung des Thermopapiers,

wenn es nicht mehr ganz tafrisch ist, mm).

Auf den Originalausdrucken habe ich keine Schwierigkeiten bei der Entzifferung.



Ein Nachteil soll nicht verschwiegen bleiben.

Die Programme benötigen nicht nur Speicherplatz, sondern auch Zeit. Mein Rechner braucht bei vollgeschriebenen Zeilen mit 'PPSC' nur (und das meine ich auch so) ca. 9 Sekunden pro Zeile ('PPNN' ca. 2 sec/Zeile).

Wer die Programmlistings durchschaut weiß auch warum.

In fast allen Programmen tauchen an zeitkritischen Stellen der Programmaufruf FAST auf. Dies ist das SUBCODE-Programm von Peter Röhl (91.2-3.64), das die Prozessorgeschwindigkeit verdoppelt.

Vom Geschwindigkeitsrausch gefangen habe ich seinerzeit diesen Programmaufruf in fast jedem anderen Programm getätigt, bis sich mein Rechner dadurch zum Batteriefresser entwickelte und bei einem Batteriewechsel einen Gedächtnisschwund erlitt.

(Spätestens in solchen Momenten freut man sich über ordentliche Programmausdrücke.)

Aber mein Rechner war ja sowieso fast voll und ich konnte den Platz gut gebrauchen...

Aber Spaß beiseite, FAST verwende ich jetzt nur noch, wenn ich es für dringend nötig halte und schalte es auch brav mit SLOW wieder aus.

Außerdem habe ich festgestellt, daß sich der Drucker im FAST-Modus nicht ansprechen läßt. Das war

jetzt aber genug Kleinkram.

An dieser Stelle möchte ich noch ein wenig auf 'PSUB' (+STSUB) eingehen, da es mir auch beim Verkürzen von Programmen behilflich ist.

Bei mir kommen häufiger Situationen vor, bei denen ich bestimmte, nicht über die Tastatur erreichbare Zeichen(-ketten) in ein Programm einbinden möchte.

Der normale Weg ist da etwas umständlich:

- ☞ Über die Tastatur den Wert eingeben, CHR- Taste drücken und den erzeugte Charakter zu einem String addieren.
- ☞ Anschließend EDIT aufrufen und Programm Anfangs- und Endmarken setzen sowie den Programmcode.
- ☞ Sollen mehrere einzelne Zeichenketten im Programm vorkommen, dann muß der String auch noch nachträglich getrennt werden.
- ☞ Kommt dann auch noch ein CHR 0 im String vor, kann man sich von dieser Methode sowieso verabschieden, da das Programm oder der String nicht mehr editierbar ist.

Also läßt man das einfach und speichert die Strings in einer Liste oder unter 'Name' ab und ruft sie im Programm auf.

Aber dann hat man die vielen GET's im Programm oder ein aufgeblasenes Menü.

Als Beispiel diene 'PRLL' (241.5 Bytes) und 'PRLR' (168 Bytes) aus dem Mind-Paket.

Im folgendem Ausdruck sieht man wie die beiden Programme in meinem Rechner stehen und zusammen 143 Bytes kürzer sind (und noch mal als Anschauungsbeispiel für 'PPNB' dienen).

Wie geht man bei der Erzeugung solcher Programme grundsätzlich vor, denn es kommen viele CHR 0 vor?

Zunächst schreibt man wie gewohnt das Programm und setzt an die

Stelle, an der später ein String stehen soll eine Buchstabenkombination ein, die auch nicht teilweise in anderen Befehlen vorkommt.

*Ich benutze im allgemeinen die Folge "QX" gefolgt von einer Nummer, wenn mehrere Strings eingebunden werden sollen.*

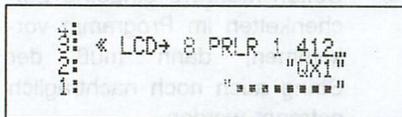
Als zweiten Schritt erzeugt man auf dem Stack die erforderlichen Strings und speichert sie unter den entsprechenden 'QXn' Namen ab.

Jetzt wird das Programm ausgetestet und Fehler beseitigt.

Der dritte Schritt besteht dann darin, die QXn Namen im Programm gegen deren Inhalt auszutauschen. Im Fall von 'PRL' wäre es folgender:

```
'PRL' RCL "QX1" 'QX1'
RCL
```

Der Stack sieht dann wie folgt aus:



Nun wird 'PSUB' ausgeführt. Als Kontrolle erhält man nach der Ausführung die Anzahl der vollzogenen Ersetzungen als Zahl in Ebene 1.

Dann weiter mit dem zweiten String:

```
DROP (Zahl löschen) "QX2"
'QX2' RCL PSUB DROP
```

Das fertige Programm dann mit STR → 'PRL' STO abspeichern. Sollte man später aber doch noch Änderungen vornehmen wollen, ist das Programm nicht verloren.

Mit 'PRL' RCL → STR DUP 1 CHR POS stellt man fest, daß CHR 1 in diesem Programm nicht vorkommt (0 in Ebene 1).

Mann ersetzt mit PSUB CHR 0 durch CHR 1 und kann das Programm wieder normal Editieren. Anschließend wieder zurückwechseln.

Jetzt noch etwas Nützliches:

Ein Programm zur "Dichtekorrektur für Kernschnitte aus natürlichen Rohölen mit einem Gehalt an aromatischen Kohlenwasserstoffen von weniger als 50 Gew % (gültig im

Bereich von 10-25°C)".

Soweit DIN 51757. Aber wofür braucht man das (außer auf der Arbeit)?

Ein Beispiel hilft die Frage klären.

Es wird Herbst und man wartet sehnsüchtig auf den Heizöllieferanten.

Endlich kommt er und verkündet freudestrahlend, daß er direkt von der Raffinerieverladung kommt und tut so, als ob er euterwarme Milch verkaufen will.

Mit 30°C rauschen nun 5000l Heizöl in die Kellergewölbe und sollen hier mal 0.50DM pro Liter kosten. Der Fahrer quittiert dann zufrieden 2500 DM und zieht von dannen.

Und jetzt die Preisfrage:

Wieviel habe ich zuviel gezahlt?

Die vorgeschriebene Liefertemperatur laut Rechnung oder Zähleraufdruck soll 20°C betragen.

Da Flüssigkeiten beim Abkühlen ihr Volumen verringern, bleibt es im Kellertank aber nicht bei diesen 5000l.

Die Dichte des Heizöls beträgt ca. 0.860 g/ml bei 20°C.

Auf 30°C umgerechnet:

```
3: 0.860 (Dichte ist)
2: 20 (Temperatur ist)
1: 30 (Temperatur soll)
```

'DKOR' liefert dann:

```
2: 0.00065 (Korrekturwert Alfa - Dichteänderung pro 1°K)
1: 0.85347 (Korrigierte Dichte)
```

Man hat also in Wirklichkeit 5000 Liter mit einer Dichte von 0.853 g/ml oder 4267.3 kg erhalten.

Bei 20°C mit einer Dichte von 0.860 g/ml hätte man aber 4300 kg bekommen. Es fehlen 32.7 kg (mit 0.860 g/ml) oder 37,98 Liter.

Also 18.99 DM zuviel gezahlt.

Außerdem kann er die 37.98 Liter nochmal für 18.99 DM verkaufen. Bei einem Zugewinn von 37.98 DM können wir nun auch verstehen, warum sich der Fahrer so gefreut hat, als er gekommen ist...

Noch eine Anmerkung:

Fahrbenzin läßt sich wegen des ho-

hen Anteils an Aromaten damit nicht umrechnen. Und im übrigen rechnen wir in unserem Betrieb nach diesen Werten in einem Temperaturbereich von 10°C bis 230°C (Bitumen) ohne Einschränkung.

## Bruchrechnen

Und noch etwas habe ich ausgebuddelt.

Als ich gerade einigermaßen mit dem Rechner klar kam und den Befehlsumfang überblicken konnte, vermißte ich auch schon etwas (kaum zu glauben bei ca. 1500 Befehlen laut Werbung).

Mein alter CASIO konnte mit Brüchen rechnen, also sollte der HP-28S das auch können.

So entstand das Bruch-Paket.

Der Bruch  $\frac{1}{3}$  ist als 1 ENTER 3 ENTER → BR einzugeben und wird in das "Format" einer komplexen Zahl umgewandelt, mit der dann weitergerechnet werden kann.

```
'→BR' Eingabeumwandlung
'BRA' addiert zwei Brüche
'BRS' subtrahiert zwei Brüche
'BRM' multipliziert zwei Brüche
'BRD' dividiert zwei Brüche
'BRK' kürzt einen Bruch
'BRN' bringt zwei Brüche auf den gleichen Nenner
```

```
'GGT' größter gemeinsamer Teiler
'KGV' kleinstes gemeinsames Vielfaches
'D>R' Dezimalzahl in Bruch
```

90.04.28  
Ralf Pfeiffer  
PRISMA 89.03.39

```
'D>RK' D>R Ergebnis in Bruchformat konvertieren
```

Vielleicht kann es dem ein oder anderen beim Nachsehen der Hausaufgaben der Kinder helfen.

Die einzige Einschränkung ist, daß gemischte Brüche wie  $1\frac{1}{3}$  als  $\frac{4}{3}$  bearbeitet werden müssen.

Die Ergebnisse von Rechnungen werden alle gekürzt ausgegeben.

Das kann man unterbinden, indem man 'BRK' in den anderen Programmen löscht und das Kürzen nach Bedarf selbst durchführt.

Zu 'D>R':

Dezimalzahl in Ebene 1 'D>R' aufrufen und mit der "NEXT" Taste die gewünschte Näherung ermitteln. Mit 'D>RK' kann das Ergebnis in einen "Bruch" umgewandelt werden.

Um aber auch noch etwas frisches bieten zu können habe ich mich noch einer Sache angenommen, die von Ralf Pfeiffer schon mehrfach angesprochen wurde:

Die Umsetzung von HP-41C Programmen auf dem HP-28S, insbesondere die indirekte Adressierung.

Die vorgestellte Lösung simuliert die Speicherregister durch eine Liste, deren Elemente abgerufen werden können. (Die Liste selbst wird unter 'REG4' abgelegt.) Die Eingabelogik bleibt aber ganz Prä-(Post?)fix.

Die Umsetzungen würden lauten:

| 41C        | 28S      |
|------------|----------|
| STO 02     | STO4     |
| RCL 02     | RCL4     |
| ST+ 02     | STR4     |
| ST- 02     | STS4     |
| ST* 02     | STM4     |
| ST/ 02     | STD4     |
| STO IND 02 | IND STO4 |
| RCL IND 02 | IND RCL4 |
| ST+ IND 02 | IND STR4 |
| USW=       | USW=     |
| SIZE 20    | SIZE4    |
| X<> 02     | X<N4     |
| X<> IND 02 | IND X=N4 |

Im übrigen stelle ich fest, daß diese Befehle auch auf dem HP-28S eine bequeme Ablagemethode für Daten darstellt, da man sie nicht mehr einzeln mit Namen abspeichern braucht und die Werte platzsparend an einem Ort untergebracht sind.

Es ist nur darauf zu achten, daß die Daten immer nur im momentanen Verzeichnis verfügbar sind.

Da ich bei dieser Gelegenheit auch mal wieder meinen HP-41C in die Hand genommen habe, hier noch mal das Dichtekorrekturenprogramm für diesen Rechner. (Das HP-28S Programm ist etwas anders und genauer, trotzdem genügen beide Umrechnungen technischen Anforderungen.)

Mindesthardware:

HP-41C + XF-Modul , 69 Register

| RHO starten und: | Anzeige |
|------------------|---------|
| Normtemperatur   | <T      |
| ENTER            | />      |
| Normdichte       | D       |
| ENTER            | /       |
| neue Temperatur  | T-IST   |
| R/S              |         |

Die Normtemperatur kann weggelassen werden, wenn sie 15°C beträgt (deswegen in der Anzeige in Klammern).

Als Ergebnis werden Dichte und Korrekturwert Alfa angegeben. z.B. "0,853 65.E-5" in Anlehnung an das Beispiel von vorhin.

Durch Drücken der Backspace Taste verschwindet die Alphaanzeige und die Dichte steht zur weiteren Berechnung im X-Register, der Korrekturwert \* 100.000 im Y-Register.

- ```

01◆LBL "RHO"
02◆LBL 04
03 15
04 "<T^>D^T-IST"
05 PROMPT
06 STO 03
07 RDN
08 STO 02
09 RDN
10 STO 01
11 "BIT. J/N ?"
12 AVIEW
13 GETKEY
14 GTO IND X
15◆LBL 41
16 RCL 02
17 .81
18 X<Y?
19 GTO 01
20 1.8 E-3
21 STO 04
22 -1.47 E-3
23 STO 05
24 GTO 03
25◆LBL 01
26 RCL 02
27 .88
28 X<Y?
29 GTO 02
30 1.27 E-3
31 STO 04
32 -7.16 E-4
33 STO 05
34 GTO 03
    
```

- ```

35◆LBL 02
36 7.35 E-4
37 STO 04
38 -1.08 E-4
39 STO 0
40◆LBL 03
41 RCL 02
42 RCL 05
43 *
44 RCL 04
45 +
46◆LBL 05
47 STO 06
48 RCL 01
49 RCL 03
50 -
51 *
52 ST+ 02
53 CLA
54 FIX 3
55 ARCL 02
56 "┌"
57 RCL 0
58 1
59 *
60 FIX 0
61 ARCL X
62 "└E5"
63 RCL 02
64 FIX 3
65 PROMPT
66 GO 04
67 LBL 25
68 6.1 E-4
69 GTO 05
70 END
    
```

- REG:01    Tempeartur ist  
           02    Dichte ist  
           03    Temperatur neu

## Ich

Zum Schluß möchte ich mich noch kurz vorstellen.

Ich heiße Wolfgang und bin 27 Jahre alt.

Beruflich bin ich in der Rohölverarbeitung (Destillation) und Bitumenproduktion tätig.

Nein, kein Dipl.-Ing., Prof. oder Techniker, sondern nur geplanter Wechselschichtarbeiter (man verzeihe den kleinen Seitenhieb auf Prisma Jan/Feb 91 S.4 Zeilen 18-20).

Mein Rechnerfieber begann 1978 mit einem CASIO Taschenrechner (6 Speicherregister, 21 Programmschritte).

1979 sattelte ich auf den HP-41CV um. Schon ein Jahr später vereinbarte mich etwas Neues.

Ein TRS-80 I kompatibler 8-Bitter (Microsoft Level 2 Basic). Durch HP-41 mit UPN und Stack interessierte ich mich damals schon früh für MMS-Forth.

1989 bin ich dann dem CCD (3497) beigetreten und habe erst mal eine geistige Aufholjagd mit dem HP-41 hinter mich gebracht (Programmier-techniken, Synthetik) und Magnetkartenleser sowie XF-Modul angeschafft.

Aber als ich so richtig fit war erschien der Artikel über den HP-28S in der PRISMA. Jetzt sind auch HP-28S und IR-Drucker mein Eigen, Freude und Hilfe. Seit gut einem Jahr nimmt mich dann noch ein Mega-ST in Anspruch, auf dem ich jetzt seit kurzem mit "32-Forthcompiler V4.05" die Programmiersprache gefunden habe, die meinen Vorstellungen entspricht.

(Forth-83 PD und Forthmacs Shareware stellten mich nicht zufrieden). Ich suche daher Unterstützung auf diesem Gebiet und würde mich über Kontakte freuen. So, daß war die Sache in eigener Sache und zum Schluß blende ich auch noch meine Telefonnummer mit ein. (Wer fragen hat.)

## RHO

Zeile 1 von RHO (1-4) CCD-Barcodes Wolfgang Führer



Zeile 2 von RHO (5-8) CCD-Barcodes Wolfgang Führer



Zeile 3 von RHO (9-11) CCD-Barcodes Wolfgang Führer



Zeile 4 von RHO (12-19) CCD-Barcodes Wolfgang Führer



Zeile 5 von RHO (20-22) CCD-Barcodes Wolfgang Führer



Zeile 6 von RHO (23-29) CCD-Barcodes Wolfgang Führer



Zeile 7 von RHO (30-32) CCD-Barcodes Wolfgang Führer



Zeile 8 von RHO (33-38) CCD-Barcodes Wolfgang Führer



Zeile 9 von RHO (39-46) CCD-Barcodes Wolfgang Führer



Zeile 10 von RHO (47-59) CCD-Barcodes Wolfgang Führer



Zeile 11 von RHO (60-65) CCD-Barcodes Wolfgang Führer



Zeile 12 von RHO (66-72) CCD-Barcodes Wolfgang Führer



Zeile 13 von RHO (73-72) CCD-Barcodes Wolfgang Führer



Zeile 14 von RHO (73-73) CCD-Barcodes Wolfgang Führer



```

ZBS ----- 09.03.1992
'PRLR'
< X
  * 27 CHR 147 CHR + 001
  0 CHR DUP + + X CHR 003
  + 0 CHR DUP DUP + +
  + X CHR DUP 1 135 005
    START OVER +
    NEXT + + 0 CHR 007
  DUP DUP + + + X CHR 009
  + PR1 DROP
  *
  *
  168 Bytes

ZBS ----- 09.03.1992
'PRLR'
< LCD+ 8 PRLR 1 412 001
  FOR A 27 CHR 149
  CHR + 0 CHR DUP + + 003
  255 CHR + 0 CHR DUP
  DUP + + + OVER A DUP 005
  136 + SUB + 0 CHR
  DUP DUP + + + 255 007
  CHR + 0 CHR DUP DUP
  + + + PR1 DROP 137 009
  STEP 16 PRLR +LCD
  *
  *
  241.5 Bytes
    
```

```

ZBS ----- 18.03.1992
'PRLR'
< X
  < " " X CHR +
  " " + X CHR DUP 1 003
  135
  START OVER +
  NEXT + + " " +
  X CHR + PR1 DROP 007
  *
  *
  122 Bytes

ZBS ----- 10.03.1992
'SDIS'
< SWAP 1 35 SUB SML 001
  " 3 PICK DISP LCD+
  DUP 1 137 6 PICK 1 - 003
  * SUB ROT NULL + 1
  137 * 1 + 548 SUB +
  +LCD
  *
  *
  157.5 Bytes
    
```

```

ZBS ----- 18.03.1992
'PRLR'
< LCD+ 8 PRLR 1 412 001
  FOR A " "
  OVER A DUP 136 + SUB 003
  + " " + PR1
  DROP 137 005
  STEP 16 PRLR +LCD
  *
  *
  144.5 Bytes

ZBS ----- 10.03.1992
'PSUB'
< 3 ROLL +STR 0 4 001
  ROLLD
  WHILE DUP 4 PICK 003
  POS
  REPEAT 4 ROLL 1 + 005
  4 ROLLD 3 PICK 3
  PICK STSUB 007
  END 3 ROLLD DROP2
  SWAP 009
  *
  *
  99.5 Bytes
    
```

# Taschenrechner

```

ZBS ===== 11.03.1992
'QANF'
« ( ) 'QDIS' STO ( ( 001
6 6 0 ) ( 0 0 0 ) (
0 23 0 ) ( 0 0 3 ) ( 003
31 10 31 ) ( 19 31 9
) ( 25 4 19 ) ( 31
27 28 ) ( 0 3 0 ) ( 005
14 17 17 ) ( 17 17
14 ) ( 21 14 21 ) (
4 14 4 ) ( 0 5 0 ) ( 009
) ( 4 4 4 ) ( 0 16 0 )
) ( 24 4 3 ) ( 14 21
14 ) ( 17 31 16 ) (
29 21 23 ) ( 17 21
31 ) ( 7 4 31 ) ( 23
21 29 ) ( 31 21 29 )
) ( 25 5 3 ) ( 31 21
) ( 10 0 23 ) ( 0 31 21
) ( 4 10 17 ) ( 10 10
10 ) ( 17 10 4 ) ( 2
25 6 ) ( 14 21 23 )
) ( 30 5 30 ) ( 31 21
10 ) ( 14 17 17 ) (
31 17 14 ) ( 31 21
17 ) ( 31 5 1 ) ( 14
17 29 ) ( 31 4 31 )
) ( 17 31 17 ) ( 9 17
15 ) ( 31 6 25 ) (
31 16 16 ) ( 31 2 31
) ( 31 4 30 ) ( 14
17 14 ) ( 31 5 2 ) (
6 9 30 ) ( 31 2 26 )
) ( 18 21 9 ) ( 1 31 1
) ( 31 16 31 ) ( 15
16 15 ) ( 31 0 31 )
) ( 27 4 27 ) ( 3 28 3
) ( 25 21 19 ) ( 31
) ( 17 17 31 ) ( 2 2
) ( 16 16 16 ) ( 0 1 2
) ( 28 20 28 ) ( 31
20 28 ) ( 28 20 20 )
) ( 28 20 31 ) ( 12 20
0 ) ( 30 5 0 ) ( 92
04 124 ) ( 31 4 20 )
) ( 0 39 0 ) ( 64 0 0
) ( 31 0 20 ) ( 31
16 0 ) ( 28 12 28 )
) ( 28 4 28 ) ( 24 20
12 ) ( 20 28 ) (
28 20 ) ( 24 4 4
) ( 16 28 4 ) ( 31
18 16 ) ( 29 16 28 )
) ( 12 16 12 ) ( 28 8
28 ) ( 20 8 20 ) (
92 80 124 ) ( 4 28
16 ) ( 4 31 17 ) ( 0
31 0 ) ( 17 31 4 ) (
6 4 2 ) ( 21 10 21
) ( 0 0 0 ) ( 4 21 4
) ( 10 4 10 ) ( 12
31 1 ) ( 0 31 2 ) (
27 21 17 ) ( 31 14 4
) ( 4 28 4 ) ( 8 21
14 ) ( 20 22 21 ) (
21 22 20 ) ( 10 31
10 ) ( 12 12 18 ) (
4 21 14 ) ( 14 21 4
) ( 30 4 6 ) ( 7 28
12 ) ( 6 6 0 ) ( 4
14 27 ) ( 27 14 4 )
) ( 6 6 0 ) ( 1 118
) ( 0 6 0 ) ( 1 118
FOR X DUP X GET 073
LIST> DROP " " 1 3
START SWAP CHR 075
SWAP +
NEXT QDIS SWAP + 077
'QDIS' STO
NEXT DROP 079
» 350 Bytes 081
    
```

```

ZBS ===== 10.03.1992
'PPSN'
« 30 CF PPS 001
» 38 Bytes 003

ZBS ===== 10.03.1992
'PPSC'
« 30 SF PPS 001
» 38 Bytes 003

ZBS ===== 10.03.1992
'PPS'
« IFERR 001
WHILE DUP TYPE 6 003
==
REPEAT I PR1 1 005
FIX DUP OSIZE +STR 007
" SWAP + 128 CHR
+ "Bytes" + SWAP STO
RCL 009
IF 30 FS? 011
THEN 10 CHR
" " + 10 CHR " " +
PSUB DROP 013
END +STR SWAP
+ PR1S CR CR DROP 015
CLMF 017
END 019
THEN 30 CF
END 021
» 253 Bytes

ZBS ===== 10.03.1992
'STSUB'
« OVER 001
IF TYPE
THEN ROT ROT DUP2 003
POS 4 ROLL ROT
ELSE DUP 005
END SIZE 3 PICK 5
PICK SIZE OVER 3 007
SWAP 0 >
IF AND 009
THEN 4 PICK 1 5
PICK 1 - SUB ROT + 4 011
ROLLD + OVER SIZE
SUB + 013
ELSE DROP DROP2 015
END
» 157 Bytes

ZBS ===== 10.03.1992
'SML'
« FAST " " 1 3 PICK 001
SIZE
FOR X OVER X X SUB 003
NUM 148 MIN
IF DUP 10 == 005
THEN 22 +
END 31 MAX 30 - 007
QDIS SWAP GET + 0
CHR + 009
NEXT SWAP DROP 011
SLOW 013
» 103.5 Bytes

ZBS ===== 12.03.1992
'PPNN'
« 1 CF PPN 001
» 30 Bytes 003

ZBS ===== 12.03.1992
'PPNB'
« 1 SF PPN 001
» 30 Bytes 003
    
```

```

ZBS ===== 12.03.1992
'PPN'
« CLLCD 001
" Ausdruck von " 3
IDISP 1 1 + X Y 003
« 1 STD 27 CHR 253
CHR + OVER +STR DUP 005
4 DISP + 27 CHR +
252 CHR + PR1 DROP 007
" OVER OSIZE +STR
+ " Bytes" + SWAP 009
RCL +STR 10 CHR +
SWAP + 10 CHR + 011
WHILE DUP DUP 10
CHR POS DUP 21 013
IF >
THEN DROP 21 015
END DUP 'X'
STO 017
REPEAT FAST 1 X
SUB DUP 1 DISP 27 019
CHR 137 CHR + LCD + 1
137 SUB + " " + Y 2 021
IF MOD
THEN Y .1001 % 023
+STR 2 4 SUB +
END SLOW PR1 025
DROP 1
IF FS? 027
THEN
"Bytes (Y)?" 2 DISP 029
IKEY "Y" ==
ELSE 0 031
END
IF 033
THEN FAST 27
CHR X 6 * CHR + 1 X 035
FOR A OVER A
DUP SUB NUM BILA + 037
NEXT SLOW
PR1 CR DROP 039
END DROP DUP X
1 + SWAP SIZE SUB Y 041
1 + 'Y' STO
END CR DROP2 043
CLMF » SLOW 045
» 697.5 Bytes 047

ZBS ===== 11.03.1992
'PRIS'
« FAST STD DUP +STR 001
1 + S XX
« FAST 003
DO S 10 CHR 32
CHR DUP + + POS DUP 005
IF 0 == 30 FS?
+ THEN DROP S 007
SIZE
END 'XX' STO 011
IF XX 41 >
THEN S 1 41
SUB 41 1 013
X SUB FOR X DUP X 015
== SWAP 10 CHR == OR 017
THEN X
'XX' STO 1 'X' STO 019
END -1
STEP DROP 021
END XX 41 MIN
'XX' STO S 1 XX SUB 023
S XX 1 + S SIZE SUB
'S' STO SML 27 CHR 025
OVER SIZE CHR + SWAP
+ PR1 027
UNTIL DROP S
SIZE NOT 029
END SLOW
» 490 Bytes 033
    
```

# Taschenrechner

| ZBS                   | 12.03.1992 |
|-----------------------|------------|
| 'BILA'                |            |
| < FAST DUP 99         | 001        |
| IF >                  |            |
| THEN 7                | 003        |
| ELSE 0                |            |
| END OVER 199          | 005        |
| THEN DROP 119         | 007        |
| END CHR 0 CHR +       |            |
| SWAP 100 + +STR DUP   | 009        |
| 2 2 SUB STR + 1 + BIL |            |
| SWAP GET LIST DROP    | 011        |
| 4 ROLL 3 3 SUB STR +  |            |
| 1 + BIL SWAP GET      | 013        |
| LIST DROP 7 ROLL 2    |            |
| 4                     | 015        |
| FOR A SWAP 16 * 7     |            |
| A - ROLL + CHR +      | 017        |
| NEXT 0 CHR +          |            |
| >                     | 019        |
| 258 Bytes             |            |
| 'SIZE4'               |            |
| < C ) 1 ROT           | 001        |
| START 0 +             |            |
| NEXT 'REG4' STO       | 003        |
| >                     |            |
| 54.5 Bytes            | 005        |
| 'IND'                 |            |
| < REG4 SWAP GET       | 001        |
| >                     |            |
| 30 Bytes              | 003        |
| 'RCL4'                |            |
| < REG4 SWAP GET       | 001        |
| >                     |            |
| 31 Bytes              | 003        |
| 'STO4'                |            |
| < REG4 SWAP ROT PUT   | 001        |
| 'REG4' STO            |            |
| >                     |            |
| 48.5 Bytes            | 003        |
| 'STA4'                |            |
| < DUP RCL4 ROT +      | 001        |
| SWAP STO4             |            |
| >                     |            |
| 43.5 Bytes            | 003        |
| 'STS4'                |            |
| < DUP RCL4 ROT -      | 001        |
| SWAP STO4             |            |
| >                     |            |
| 43.5 Bytes            | 003        |
| 'STM4'                |            |
| < DUP RCL4 ROT *      | 001        |
| SWAP STO4             |            |
| >                     |            |
| 43.5 Bytes            | 003        |
| 'STD4'                |            |
| < DUP RCL4 ROT /      | 001        |
| SWAP STO4             |            |
| >                     |            |
| 43.5 Bytes            | 003        |
| 'X=N4'                |            |
| < DUP RCL4 ROT ROT    | 001        |
| STO4                  |            |
| >                     |            |
| 41 bytes              | 003        |

| ZBS                  | 12.03.1992    |
|----------------------|---------------|
| 'BIL'                |               |
| { 0 6 6 }            | { 0 0 7 } 001 |
| { 4 7 1 }            | { 0 0 7 } 003 |
| { 0 6 3 }            | { 0 0 7 } 005 |
| { 0 6 7 }            | { 0 0 7 } 007 |
| { 0 7 7 }            | { 0 0 7 } 005 |
| 57.5 Bytes           | 007           |
| 'BRA'                |               |
| < BRN SWAP C+R DROP  | 001           |
| SWAP C+R 3 ROLL +    |               |
| SWAP R+C             | 003           |
| >                    |               |
| 49 Bytes             | 005           |
| 'BRS'                |               |
| < C+R SWAP NEG SWAP  | 001           |
| R+C BRA              |               |
| >                    |               |
| 36.5 Bytes           | 003           |
| 'BRM'                |               |
| < C+R 3 ROLL C+R 3   | 001           |
| ROLL * 3 ROLL *      |               |
| SWAP R+C BRK         | 003           |
| >                    |               |
| 54 Bytes             | 005           |
| 'BRD'                |               |
| < C+R SWAP R+C BRM   | 001           |
| >                    |               |
| 31.5 Bytes           | 003           |
| 'BRK'                |               |
| < DUP C+R GGT SWAP   | 001           |
| C+R 3 PICK / SWAP 3  |               |
| PICK / SWAP R+C SWAP | 003           |
| DROP                 |               |
| >                    |               |
| 61.5 Bytes           | 005           |
| 'BRN'                |               |
| < BRK SWAP BRK SWAP  | 001           |
| C+R 3 ROLL C+R + C D |               |
| A B                  | 003           |
| < B D GGT + E        |               |
| * A D * E / C B      | 005           |
| * E / B D * E / DUP  |               |
| 3 ROLL R+C 3 ROLL    | 007           |
| R+C SWAP             |               |
| >                    |               |
| 009                  |               |
| >                    |               |
| 011                  |               |
| 174 Bytes            | 011           |
| 'DPR'                |               |
| < ABS RND DUP 1 DISP | 001           |
| 1 0 0 1 1 6 PICK FP  |               |
| LAST IP 0            | 003           |
| DO DROP 7 PICK       |               |
| OVER * 7 ROLL + 7    | 005           |
| ROLL 5 PICK OVER *   |               |
| 5 ROLL + 5 ROLL      | 007           |
| DROP SWAP OVER MOD   |               |
| LAST DUP 1 IETE / IP | 009           |
| UNTIL "1" 8 PICK     |               |
| +STR + "/" + 6 PICK  | 011           |
| +STR + STR + DUP DUP |               |
| 3 DISP EVAL RND DUP  | 013           |
| 2 DISP 10 PICK ==    |               |
| DO KEY               | 015           |
| UNTIL                |               |
| END "NEXT" * OR      | 017           |
| END 9 ROLL 8         |               |
| DROPN CLMF           | 019           |
| >                    |               |
| 249 Bytes            | 021           |

| ZBS                  | 12.03.1992 |
|----------------------|------------|
| 'DKOR'               |            |
| < + DI TI 7          | 001        |
| >                    |            |
| IF .62 DI <          | 003        |
| THEN .00097 DI       |            |
| .62 - 649 / -        | 005        |
| END                  |            |
| IF .806 DI <         | 007        |
| THEN DROP .00069     |            |
| DI .805 - 1500 / -   | 009        |
| END                  |            |
| IF .881 DI <         | 011        |
| THEN DROP .00064     |            |
| DI .88 - 7933 / -    | 013        |
| END                  |            |
| IF 1 DI <            | 015        |
| THEN DROP .00061     |            |
| END DUP TI 7 - *     | 017        |
| DI +                 |            |
| >                    |            |
| >                    |            |
| 354.5 Bytes          | 021        |
| 'DARK'               |            |
| < +STR "/ " " " "    | 001        |
| STSUB STR + BR       |            |
| >                    |            |
| 51.5 Bytes           | 003        |
| 'CHAR'               |            |
| < + DUP ST STO       | 001        |
| 'BF' STO LCD + 1 411 |            |
| SUB DUP 'LC' STO 254 | 003        |
| CHR + +LCD           |            |
| DO                   | 005        |
| DO KEY               |            |
| UNTIL                |            |
| END 2500 .01         | 007        |
| RECP                 |            |
| IF DUP "E" ==        | 009        |
| THEN RX 4 DISP 2     |            |
| WAIT                 |            |
| END                  | 013        |
| IF DUP "L" ==        |            |
| THEN DROP PRLL       | 015        |
| "BACK"               |            |
| END                  |            |
| IF DUP "BACK" ==     | 017        |
| OVER "123456789" 1   |            |
| SS 2 + SUB SWAP POS  | 019        |
| NOT OR               |            |
| THEN DROP "" DUP     | 021        |
| 'ST' STO 'BF' STO "" | 023        |
| 4 DISP               |            |
| ELSE DUP ST SWAP     | 025        |
| + 'ST' STO STR + ANZ |            |
| SWAP GET BF SWAP +   | 027        |
| 'BF' STO             |            |
| END BF LC SWAP +     | 029        |
| 254 CHR + +LCD       |            |
| UNTIL ST SIZE SS     | 031        |
| ==                   |            |
| END                  | 033        |
| >                    |            |
| 469 Bytes            | 035        |
| 'KGV'                |            |
| < * LAST GGT /       | 001        |
| >                    |            |
| 31.5 Bytes           | 003        |
| 'GGT'                |            |
| <                    | 001        |
| DO SWAP OVER MOD     |            |
| UNTIL DUP NOT        | 003        |
| END DROP             |            |
| >                    |            |
| 40 Bytes             | 005        |
| 'BR'                 |            |
| < R+C STO            | 001        |
| >                    |            |
| 22.5 Bytes           | 003        |



# Buchbesprechung

## HP-48 noch effektiver nutzen Anwendungsprogramme für Fortgeschrittene

Prof. Jean-Michel Ferrard  
ISBN 2-908791-05-6.

Das Buch habe ich als Beilage eines Aachener Händlers zum 48S/48SX gefunden, einzeln soll es zw. 30-40 DM kosten, und mit der ISBN-Nummer sollte es jeder Buchhändler besorgen können.

Bei dem hier vorgestellten Werk handelt es sich um den 2. Teil, welcher eine reine Programmsammlung ist.

Der Autor wird als Lehrkraft für Höhere Mathematik vorgestellt, und entsprechend orientiert ist die Themenauswahl.

Der Anhang weist ca. 220 Programme auf insgesamt 282 Seiten aus, so daß man sich vorstellen kann, da dieses Buch nicht für den Anwender gedacht ist, der sich mit kommentierten Listings in die Programmierung des 48SX einarbeiten will.

Auch auf mathematische Erläuterungen verzichtet der Autor, das ganze Buch stellt also nur eine Sammlung von Listings, Beispielen (mit Laufzeiten) und Bedienungsanleitungen dar.

Die von Ferrard verwendeten schlichten Stackdiagramme finde ich sogar besser als die von HP, allerdings kann man die Qualität der Listings vergessen und überhaupt macht das Layout den Eindruck, als wollte jemand alle Möglichkeiten eines mittelmäßigen Textverarbeitungssystems nutzen.

Auch die Qualität der Programme läßt teilweise zu wünschen übrig (zu lang, Bedienung), ein Unterpunkt "Unzulässige Eingaben und Fehler" könnte einigen Programmen

auch nicht schaden.

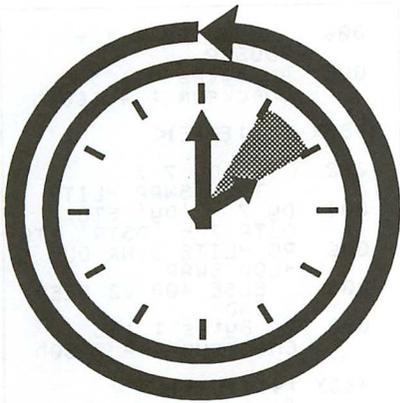
Wenn ein solches erst einmal auf einem Rechner läuft, ist es den meisten Anwendern ziemlich egal, wie es bedient wird, wenn es nur läuft. Und was J.M. Ferrard zu bieten hat soll folgende kleine Auswahl aus den folgenden 15 Kapiteln zeigen:

1. **Arithmetik**  
Diophantische Gleichungen.
2. **Komplexe und reelle Zahlen**  
Kettenbrüche, Linearisieren von sin/cos-Funktionen, Integration über Wege in der komplexen Ebene.
3. **Polynome**  
Addition, Multiplikation, Division, Verkettung, größter gemeinsamer Teiler von Polynomen, Tschebyscheff-Polynome, spezielle Integrationen.
4. **Rationale Brüche**  
Kürzen, Multiplikation und eine Vorstufe der Partialbruchzerlegung.
5. **Matrizenrechnung**  
Rang, LR-Zerlegung, ganzzahlige Inversion (geht nur bei ganzzahliger Ausgangsmatrix), Eigenwerte und Eigenvektoren, symbolische Lösungen eines Gleichungssystems.
6. **Analysis**  
Tangentengleichung in einem beliebigen Punkt für Funktionen, Gaussche Fehlerquadratmethode, Lagrange-Interpolation, nicht-lineare Gleichungssysteme (2 oder 3 Gleichungen), Runge-Kutta für DGL 1. Ordnung.
7. **Endliche Entwicklungen**  
Dieses Kapitel bietet Lösungen

rund um Reihenentwicklungen, z.B. auch ein Programm, welches besser als die Funktion TAYLR arbeitet.

8. **Geometrie**  
Berechnung von Geraden-, Ebenen- und Kreisgleichungen in verschiedenen Darstellungsformen, 2- und 3-dimensional.
9. **Differentialgeometrie**  
Bogenlänge einer 2/3-dim. Kurve, Linienintegrale, Ebene Flächen, Krümmungsradius und -mittelpunkt, Rotation und Divergenz eines Vektorfeldes.
10. **Grafische Darstellung**  
Dieses Kapitel enthält vor allem Programme, die nur auf dem 28S Sinn machen, leider fehlt den Beispielen die in der Kapitelüberschrift erwähnte grafische Darstellung, stattdessen findet der Anwender als Output-Beschreibung ein rechteckiges Kästchen mit dem Wort "Plot".
11. **Große ganze Zahlen**  
Arithmetik (+, -, \*, /, ggT und kgV) für beliebig große ganze Zahlen.
12. **Wahrscheinlichkeiten**  
Verteilungsfunktionen (Poisson, Pascal, Binomial, hypergeometrisch...), Kombinationen.
13. **Statistik mit einer Zufallsvariablen**  
Momente, Mittelwerte, Häufigkeitstabelle.
14. **Statistik mit zwei Zufallsvariablen**  
Hier wurden anscheinend einige Programme unnötigerweise vom 28S in den besser ausgestatteten 48SX übernommen.

Fortsetzung auf Seite 58



# Termin- planung



von Wilfried Steiner

"Schon wieder" werden sich einige der Leser fragen. Ja - aber diesmal komplett auch für "Nicht-Schnittstellenkabel-Besitzer" mit Erweiterungen, die dieses Programm erst zum Terminkalender machen.

*Was nützt ein Terminkalender, wenn man seine Termine irgendwo versteckt unter "Notizen" einträgt ?*

**Anstehende Termine müssen beim Aufschlagen des Kalenders sofort erkennbar sein.**

Dies ist auch schon die wichtigste Verbesserung, die ich anzubieten habe.

Desweiteren werden noch Tages- und Wochenummer zur Anzeige gebracht. Was jetzt noch fehlt, das ist die Anzeige der Feiertage (aber viel langsamer darf der Rechner dabei nicht werden).

Nun zum Programm selbst:

Aufbauend auf der Version von Günter Schapka (PRISMA 1/91), der die amerikanische Version ins Deutsche übersetzt und frisiert hat, habe ich mich mit der Anzeige der Termine und der Wochenberechnung auseinandergesetzt.

Das Programm 'MTALRM' sucht innerhalb des aktuellen Monats nach anfallenden Terminen und bringt diese als Vormittags- oder Nachmittagstermine zur Anzeige ('MHLIT'). Dabei dürfen nur je ein Vormittags- und ein Nachmittagstermin ausgewertet werden.

Da Wiederholtermine (z.B. zur täglichen Zeitkorrektur) von wichtigen Terminen ablenken, habe ich diese generell unterdrückt (Zeilen 21..24

in 'MTALRM').

Da der Rechner anscheinend der Meinung ist, daß Termine vor seiner Existenz auch nicht zu existieren haben, mußte ich Prüfung auf das Datum 1.1.1989 in den Zeilen 2..3 zufügen. Hier kann z.B. abgeändert werden, daß Vergangenheitstermine generell nicht angezeigt werden.

Nun zur Wochenummerberechnung:

Leider ist die Wochenbestimmung der ersten und letzten Tage eines Jahres recht aufwendig (ein Kollege hat sich damit eingehend befaßt - herzlichen Dank Frank), sodaß ich diese Berechnungen nur bei Änderung ausführe ('WKSET') und die Grenzen in den Variablen 'WKFRST' (erste Woche) und 'WKLST' (letzte Woche) ablege.

Wichtig ist auch der Montag der 1. Woche ('MOFRST'), der auch für das vergangene und folgende Jahr berechnet werden muß. Danach ist die Wochenbestimmung recht kurz ('WK?') und wird mit der Tagesnummernberechnung durch 'DYWK' ausgegeben.

Notwendig wurde dazu auch die Abspeicherung des aktuellen Datums in 'DSTR', was in der Basisversion nicht immer aktualisiert wurde, sowie die Überprüfung durch 'DTMAX', damit nicht auf ungültige Daten gesprungen werden kann. Die kompletten Verzeichnisse habe ich der Reihe nach ausgedruckt, wobei **APDIR**, **MASTR**, **CONTROL**, **NXT**, **APLST** und **FNDR** Directories sind.

- ☛ Gestartet wird das Programm mit 'APPT'.
- ☛ Die Cursor sind zum Wandern im Kalender.
- ☛ Mit Plus und Minus kann man einen Monat vor oder zurück springen,
- ☛ mit den linksumgeschalteten Plus- und Minustasten je ein Jahr vor oder zurück.
- ☛ Ansonsten sollte mit Softkey "GOTO" gesprungen werden.

Nun noch ein paar Hinweise für "Schnittstellenkabel-Besitzer": Gegenüber der Basisversion **entfallen** folgende Programme: M1 bis M7, CADR, DFLIP, MKSTR, GDATES, RAPPTS, GDALRM.

Folgende Programme wurden **geändert**:

GOTO, RDOSCR, REFRESH, HLIT2, RC, SETUP2, YRMIN, YRPLS, MOMIN, MOPL, PWECK, NWECK, DYMIN, DYPL, DOKEY, HLIT, MNTH, LMNTH, MMY, HEADR, DFRST, NOHEAD, MHEAD, MSSG, HOUR, DATTE, RFSH, TOPC, OVERALL, ENDPROC, GTSTR, APPTS und MYFLAGS.

Nun wünsche ich Euch noch einen verregneten Nachmittag.

```

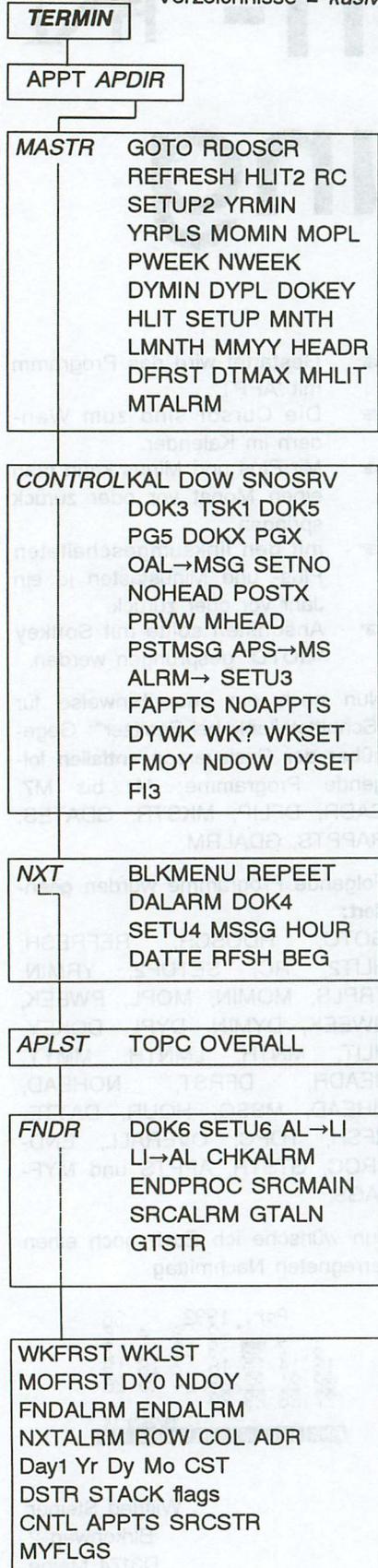
Apr. 1992
Su
16 7 16 25
17 8 17 26
18 9 18 27
19 10 19 28
20 11 20 29
21 12 21 30
22 13 22 31
23 14 23
24 15 24
25 16 25
26 17 26
27 18 27
28 19 28
29 20 29
30 21 30
31 22 31
FND GOTO RDD UPLD RAPPTS STDP
  
```

Wilfried Steiner  
Birkenweg 2  
D3174 Meine

**Verzeichnisstruktur**

(bitte unbedingt einhalten)

Verzeichnisse = *kusiv*



```

485X APPT
< APDIR MASTR
002 CONTROL NXT APLST
    FNDR RCLF flags
004 STO MYFLGS STOR
    CNTL flags STOR
006 HOME 2 MENU
    > Bytes : 116.5
008 Checksum : # 8061h

485X GOTO
< DROP2 BLKMENU
002 "datum (DD.MMYYYY):
    dann ENTER."
004 DSTR →STR -1 2
    →LIST INPUT OBJ→
006 DTMAX 'DSTR' STO
    > Bytes : 104
008 Checksum : # F7E4h

485X RDOSCR
< SETUP2 MNTH LCD→
002 MTALRM RC HLIT2
    DYWK DUP →LCD SWAP
004 > Bytes : 68
    Checksum : # 6193h

485X REFRESH
< SETUP MNTH LCD→
002 MTALRM HLIT DYWK
    DUP →LCD
004 > Bytes : 58
    Checksum : # 1C11h

485X HLIT2
< COL R→B ROW R→B 2
002 →LIST DUP2 COL 12 +
    R→B ROW 8 + R→B 2
004 →LIST SUB NEG REPL
    > Bytes : 84
006 Checksum : # 6C95h

485X RC
< Dy ADR + 7 / FP 7
002 * 3 * 6 * 5 + 'COL'
    STO Dy ADR + 7 / IP
004 1 + 8 * 1 - 'ROW'
    STO
006 > Bytes : 117
    Checksum : # 952Dh

485X SETUP2
< DUP 'DSTR' STO
002 DUP MMY DDFRST
    > Bytes : 48.5
004 Checksum : # 281Ah

485X YRMIN
< ROT DROP2 DSTR
002 .000001 - DTMAX
    RDOSCR
004 > Bytes : 53.5
    Checksum : # F8BCh

485X YRPLS
< ROT DROP2 DSTR
002 .000001 + DTMAX
    RDOSCR
004 > Bytes : 53.5
    Checksum : # 5AB9h

485X MOMIN
< ROT DROP2 DSTR FP
002 100 * DUP IP 1 -
    IF 0 ==
004 THEN FP .0001 -
    12 +
006 ELSE 1 -
    END 100 / 1 +
008 RDOSCR
    > Bytes : 134
010 Checksum : # C36Ch

485X MOPL
< ROT DROP2 DSTR FP
002 100 * DUP IP 1 +
    IF 13 ==
004 THEN FP 1.0001 +
    ELSE 1 +
  
```

```

006 END 100 / 1 +
    RDOSCR
008 > Bytes : 129
    Checksum : # 1601h

485X P W E E K
<
002 IF Dy 7 >
    THEN SWAP HLIT2
004 Dy 7 - 'Dy' STO
    DSTR 7 - 'DSTR' STO
006 RC HLIT2 DYWK DUP
    →LCD SWAP
008 ELSE 400 .2 BEEP
    END
010 > Bytes : 155
    Checksum : # 7150h

485X N W E E K
<
002 IF Dy DSTR LMNTH
    6 - <
004 THEN SWAP HLIT2 7
    Dy + 'Dy' STO DSTR
    7 + 'DSTR' STO RC
    HLIT2 DYWK DUP →LCD
    SWAP
008 ELSE 400 .2 BEEP
    END
010 > Bytes : 173.5
    Checksum : # 5C66h

485X D Y M I N
<
002 IF Dy 1 >
    THEN SWAP HLIT2
004 Dy 1 - 'Dy' STO
    DSTR 1 - 'DSTR' STO
006 RC HLIT2 DYWK DUP
    →LCD SWAP
008 ELSE 400 .2 BEEP
    END
010 > Bytes : 155
    Checksum : # 11ADh

485X D Y P L
<
002 IF Dy DSTR LMNTH
    <
004 THEN SWAP HLIT2 1
    Dy + 'Dy' STO DSTR
    1 + 'DSTR' STO RC
    HLIT2 DYWK DUP →LCD
    SWAP
008 ELSE 400 .2 BEEP
    END
010 > Bytes : 168.5
    Checksum : # 5704h

485X D O K E Y
<
002 CASE DUP 11.1 ==
    THEN DROP
004 SRCMAIN REFRESH
    11.1
006 END DUP 12.1 ==
    THEN DROP GOTO
008 REFRESH 12.1
    END DUP 36.1 ==
010 THEN DYPL
    END DUP 34.1 ==
012 THEN DYMIN
    END DUP 35.1 ==
014 THEN N WEEK
    END DUP 25.1 ==
016 THEN P WEEK
    END DUP 95.1 ==
018 THEN MOPL
    END DUP 85.1 ==
020 THEN MOMIN
    END DUP 95.2 ==
022 THEN YRPLS
    END DUP 85.2 ==
024 THEN YRMIN
    END DUP 13.1 ==
026 THEN CLEAR DSTR
    TIME 100 * IP 100 /
028 " 0 4 →LIST BEG 1
    CF REFRESH 13.1
  
```

```

030     END DUP 14.1 ==
        THEN DROP
032 OVERALL REFRESH
    14.1
034     END DUP 15.1 ==
        THEN APPTS
036 REFRESH 15.1
        END DUP 91.3 ==
038     THEN OFF
        END
040     END Yr OBJ→ 10000
        / Mo + 100 / Dy +
042 'DSTR' STO
    » Bytes : 680
044 Checksum : # CAD9h

48SX HLIT
« Dy ADR + 7 / DUP
002 FP 7 * 3 * 6 * 5 +
    DUP 'COL' STO R+B
004 SWAP 'IP' + 8 * 1 -
    DUP 'ROW' STO R+B 2 +
006 →LIST DUP2 COL 12 +
    R+B ROW 8 + R+B 2 +
008 →LIST SUB NEG REPL
    » Bytes : 168.5
010 Checksum : # 19CBh

48SX SETUP
« { "FIND" "GOTO"
002 "ADD" "UPLD"
    "APPTS" "STOP" }
004 MENU DSTR SETUP2
    » Bytes : 88.5
006 Checksum : # 851h

48SX MNTH
« DSTR DYSET WKSET
002 LMNTH KAL DROP2
    » Bytes : 52
004 Checksum : # 8C93h

48SX LMNTH
« FP 1 + DUP 32
002 DATE+ FP 1 + DDAYS
    » Bytes : 43
004 Checksum : # FFA8h

48SX MMY
« DUP IP 'Dy' STO
002 FP 100 * DUP IP
    'Mo' STO FP 10000 *
004 →STR 'Yr' STO
    » Bytes : 92.5
006 Checksum : # 9DACH

48SX HEADR
«
002 "Jan. Feb. März Apr
    . Mai Juni Juli Au
004 g. Sep. Okt. Nov. D
    ez.
006 Mo 5 * DUP 4 - SWAP
    SUB + Yr +
008 "So" +
    » Bytes : 133
010 Checksum : # BA76h

48SX DFIRST
« FP 1 + DOW
002 LASTARG 2 / 2 - DUP
    'ADR' STO 3 DROPN
004 » Bytes : 58
    Checksum : # 4FC7h

48SX DTMAX
« DUP LMNTH DUP 3
002 PICK
    IF IP <
004 THEN SWAP FP +
    ELSE DROP
006 END
    » Bytes : 58.5
008 Checksum : # C6FAh

48SX MHLIT
« SWAP IP DUP ADR +
002 7 / FP 126 * 18 +
    DUP 'COL' STO R+B

```

```

004 SWAP ADR + 7 / IP 1
    + 8 * 1 - ROT + DUP
    'ROW' STO R+B 2
006 →LIST 4 ROLL SWAP
008 DUP2 COL 1 + R+B
    ROW 1 + R+B 2 →LIST
010 SUB NEG REPL 3
    ROLL
012 » Bytes : 197.5
    Checksum : # EDC0h

48SX MTALRM
«
002 IF 1.011989 DSTR
    DDAYS 0 ≥
004 THEN DSTR FP 1 +
    DUP 32 DATE+ FP 1 +
006 SWAP
    WHILE FINDALARM
008 DUP
    IF
010 THEN RCLALARM
    LIST→ ROT DROP2 3
012 PICK 5 PICK
    IF DDAYS 0
014 >
    THEN 1
    ELSE 3
016 DROPN 0
    END
018 REPEAT
    IF 0 #
020 THEN .0001
    HMS+ 2 →LIST
022 ELSE
    IF 12 <
024 THEN DUP 1
    MHLIT 12 2 →LIST
026 ELSE DUP 6
    MHLIT 1 DATE+
030 END
    END
032 END DROP
034 » Bytes : 297
    Checksum : # 17Eh

48SX KAL
« LASTARG ROT 3
002 FREEZE 3 * HEADR
    " 1 1.01199
004 7 PICK DDAYS 148729
    + 7 MOD SUB DUP DUP
006 + +
    " 1 2 3 4 5 6
008 7 8 9 10 11 12
010 13 14 15 16 17 18 1
    9 20 21 22 23 24 25
    26 27 28 29 30 31"
012 1 4 ROLL SUB + 1 7
    FOR n 1 21 SUB
014 LASTARG + OVER SIZE
    SUB SWAP n DISP
016 NEXT DROP2
    » Bytes : 263
018 Checksum : # 95FCh

48SX DOW
« 0 TSTR 1 2 SUB
002 "MoDiMiDoFrSaSo"
    "MOTUMETHFRSASU"
004 ROT POS DUP 1 + SUB
    » Bytes : 75.5
006 Checksum : # F235h

48SX SNOSRV
« 2 GETI SWAP DROP
002 # 18CEAh SYSEVAL
    # E402h SYSEVAL
004 # 3244h SYSEVAL
    # E80Dh SYSEVAL
006 # 172Bh SYSEVAL
    DROP
008 » Bytes : 100
    Checksum : # 6EC6h

```

```

48SX DOK3
«
002 CASE DUP 91.3 ==
    THEN OFF
004 END DUP 14.1 ==
    THEN DROP DSTR
006 8 " " 0 4 →LIST BEG
    16.1
008 END
    END
010 » Bytes : 110.5
    Checksum : # 7FEAh

48SX TSK1
« LCD→ DUP ( # 0h
002 # FH ) ( # 87h
    # 17h ) SUB NEG (
004 # 0h # FH ) SWAP
    REPL →LCD
006 » Bytes : 120.5
    Checksum : # B3A8h

48SX DOK5
«
002 CASE DUP 91.3 ==
    THEN OFF
004 END DUP 25.1 ==
    THEN DROP DEPTH
006 ROLL PG5 25.1
    END DUP 25.2 ==
008 THEN DROP 1 5
    START DEPTH
010 ROLL
    NEXT PG5 25.2
012 END DUP 35.1 ==
    THEN DROP DEPTH
014 ROLL PG5 35.1
    END DUP 35.2 ==
016 THEN DROP 1 5
    START DEPTH
018 ROLL
    NEXT PG5 35.2
020 END DUP 11.1 ==
    THEN DROP 2
022 GETI SWAP DROP DUP
    RCLALARM SWAP
024 DELALARM DUP BEG
    IF 1 FC?
026 THEN STORALRM
    ELSE DROP
028 END 1 CF 16.1
    END DUP 12.1 ==
030 THEN DROP
    SNOSRV DROP 12.1
032 END DUP 13.1 ==
    THEN DROP
034 DALARM 16.1
    END DUP 14.1 ==
036 THEN DROP DSTR
    TIME 100 * IP 100 /
038 " " 0 4 →LIST BEG 1
    CF 16.1
040 END DUP 15.1 ==
    THEN DROP PRVW
042 PG5 15.1
    END
044 » Bytes : 607.5
    Checksum : # 470Ch

48SX PG5
« MHEAD SETU3 5
002 POSTX
    » Bytes : 38
004 Checksum : # 24DFh

48SX DOKX
«
002 CASE DUP 91.3 ==
    THEN OFF
004 END DUP 25.1 ==
    THEN DROP DEPTH
006 ROLL PGX 25.1
    END DUP 35.1 ==
008 THEN DROP DEPTH
    ROLL PGX 35.1
010 END DUP 11.1 ==
    THEN DROP 2

```

```

012 GETI SWAP DROP DUP
RCLALARM SWAP
014 DELALARM DUP BEG
      IF 1 FC?
016       THEN STORALARM
      ELSE DROP
018       END 1 CF 16.1
      END DUP 12.1 ==
020       THEN DROP
022 SNOSRV 12.1
      END DUP 13.1 ==
      THEN DROP
024 DALARM 16.1
      END DUP 14.1 ==
026       THEN DROP DSTR
      TIME 100 * IP 100 /
028       " 0 4 →LIST BEG 1
      CF 16.1
030       END DUP 15.1 ==
      THEN DROP PRVW
032 PGX 15.1
      END
034 END
    » Bytes : 485
036 Checksum : # 6A4h

485X PGX
« MHEAD SETUS DEPTH
002 POSTX
    » Bytes : 38
004 Checksum : # D65Ch

485X OAL→MSG
« DUP 5 GET SWAP
002 DUP 3 GET SWAP 2
GET DUP IP DUP
004 IF 10 <
      THEN "0" SWAP +
006      END " " + SWAP FP
      100 * IP DUP
008 IF NOT
      THEN DROP "00"
010      END + " " + SWAP
      + SWAP 2 →LIST
012 » Bytes : 158.5
    Checksum : # 60AAh

485X SETNO
« ( " " " " " "ADD"
002 " " "RTN" ) MENU
    » Bytes : 53.5
004 Checksum : # 9ABCh

485X NOHEAD
« DSTR DOW DSTR
002 TIME TSTR 4 12 SUB
      + " " SWAP +
004 " " Keine Termine a
      m"
006 CLLCD 1 DISP 1
      FREEZE 3 DISP 3
008 FREEZE
    » Bytes : 116
010 Checksum : # 8C70h

485X POSTX
« → d
002 « 1 d
      FOR 1 DUP 1 GET
004 1 2 + DISP 1 2 +
      FREEZE DEPTH ROLL 1
006 STEP 1 d
      START DEPTH
008 ROLL
      NEXT TSK1
010 »
    » Bytes : 102
012 Checksum : # 4A65h

485X PRVW
« CLLCD DUP 1 GET
002 DUP SIZE 7 SWAP SUB
      DUP SIZE 1 SWAP
004 FOR x DUP 1 22
      SUB x 22 / 1 + DISP
006 x 22 / 1 + FREEZE
      IF DUP SIZE 22
008 >
      THEN DUP SIZE

```

```

010 23 SWAP SUB
      END 22
012 " " "STEP" DROP ( " " "
      " " " " "RTN" )
014 MENU -1 WAIT DROP
    » Bytes : 233
016 Checksum : # FFE0h

485X MHEAD
« DSTR DOW DSTR
002 TIME TSTR 4 12 SUB
      + " " SWAP +
004 " " Termine f"
      220 CHR + "r" +
006 CLLCD 1 DISP 1
      FREEZE 2 DISP 2
008 FREEZE
    » Bytes : 135
010 Checksum : # DAA9h

485X PSTMSG
«
002 IF DEPTH 5 ≤
      THEN PGX
004 DO -1 WAIT DOKX
      UNTIL 16.1 ==
006 END
      ELSE PGS
008 DO -1 WAIT DOKS
      UNTIL 16.1 ==
010 END
    END CLEAR
012 » Bytes : 119
    Checksum : # 72D5h

485X APS→MS
« DEPTH 1 SWAP
002 START OAL→MSG
      DEPTH ROLL
004 NEXT
    » Bytes : 38
006 Checksum : # 95D4h

485X ALARM→
« 'NXTALRM' STO
002 DO NXTALRM
      IFERR RCLALARM
004 THEN DROP DSTR
      1 + 1 →LIST
006 ELSE OBJ→ DROP
      NXTALRM 5 →LIST
008 NXTALRM 1 +
      'NXTALRM' STO
010 END
      UNTIL DUP 1 GET
012 DSTR #
      END DROP
014 » Bytes : 162.5
    Checksum : # B5C3h

485X SETUS
« ( "EDIT" "ACK"
002 "DEL" "ADD" "VIEW"
      "RTN" ) MENU
004 » Bytes : 67.5
    Checksum : # DED7h

485X FAPPTS
« DSTR FINDALARM
002 DUP
      IF
004 THEN ALARM→
      ELSE DROP
006 END DEPTH
    » Bytes : 46
008 Checksum : # 9803h

485X NOAPPTS
«
002 DO NOHEAD SETNO
      -1 WAIT DOK3
004 UNTIL 16.1 ==
      END
006 » Bytes : 61
    Checksum : # AEBAh

```

```

485X DYWK
« "I" DY0 DSTR
002 DDAYS NDOY OVER -
      FIS SWAP FIS "/" +
004 SWAP + + "1" + WK?
      FIS + 1 →GROB (
006 # 50h # 33h ) SWAP
      REPL
008 » Bytes : 141.5
    Checksum : # 9E15h

485X WK?
« MOFRST DSTR DDAYS
002 7 / FLOOR 1 +
      CASE DUP 0 ==
004 THEN DROP
      WKFRST
006 END DUP WKLST >
      THEN DROP 1
008 END
    » Bytes : 110
010 Checksum : # 3816h

485X WKSET
« 1 DSTR %T FP 101
      + 1 % DUP FMOY DUP
002 'MOFRST' STO OVER
004 .000001 + FMOY
      DDAYS 7 / 'WKLST'
006 STO .000001 - FMOY
      MOFRST DDAYS 7 /
008 'WKFRST' STO
    » Bytes : 173.5
010 Checksum : # 3DC1h

485X FMOY
« DUP NDOY DUP
002 IF 3 ≤
      THEN 1 SWAP -
004 DATE+
      ELSE 8 SWAP -
006 DATE+
      END
008 » Bytes : 67.5
    Checksum : # 4508h

485X NDOY
« 1,01199 SWAP
002 DDAYS 7 MOD 1 +
    » Bytes : 35.5
004 Checksum : # 8E00h

485X DYSET
« 1 OVER %T FP 3112
002 + 1 % DUP .000001 -
      DUP 'DY0' STO SWAP
004 DDAYS 'NDOY' STO
    » Bytes : 90
006 Checksum : # 16E5h

485X FIS
« →STR " " SWAP +
002 DUP SIZE DUP 2 -
      SWAP SUB
004 » Bytes : 42
    Checksum : # 1C36h

485X BLKMENU
« ( " " " " " " " "
002 " " ) MENU
    » Bytes : 47.5
004 Checksum : # CE11h

485X REPEAT
« BLKMENU
002 "Repeat #. Then ENT
      ER"
004 " " INPUT OBJ→ (
      "week" "Day" "Hour"
006 "Min" "Sec" "None"
      ) MENU
008 "Now press repeat u
      nit"
010 3 DISP 3 FREEZE -1
      WAIT
012 CASE DUP 11.1 ==
      THEN DROP

```

```

014 4954521600 *
      END DUP 12.1 ==
016      THEN DROP
      707788800 *
018      END DUP 13.1 ==
      THEN DROP
020 29491200 *
      END DUP 14.1 ==
022      THEN DROP
      491520 *
024      END DUP 15.1 ==
      THEN DROP 8192
026 *
      END DUP 16.1 ==
028      THEN 0 *
      END
030      END SWAP DROP 4
      SWAP PUT
032 > Bytes : 414
      Checksum : # 2F80h

48SX DALARM
< 2 GET DELALARM
002 > Bytes : 17.5
      Checksum : # 8BEEh

48SX DOK4
<
002 CASE DUP DUP 17 >
      SWAP 91.3 * AND
004 THEN 1000 .2
      BEEP
006 END DUP 91.3 ==
      THEN OFF
008 END DUP 11.1 ==
      THEN DROP DATTE
010 11.1
      END DUP 12.1 ==
012      THEN DROP HOUR
014 12.1
      END DUP 13.1 ==
      THEN DROP MSSG
016 13.1
      END DUP 14.1 ==
018      THEN REPEAT
020 14.1
      END DUP 15.1 ==
      THEN DROP
022 STORALARM DROP 1 5F
      15.1
024 END DUP 16.1 ==
      THEN 1 CF SWAP
026 DROP
      END
028 > Bytes : 381
030 Checksum : # 48C5h

48SX SETU4
< ("DATE" "TIME"
002 "MSG" "RPT" "SET"
      "ABRT" ) MENU
004 > Bytes : 68.5
      Checksum : # 8FDAh

48SX MSSG
< BLKMENU 3 GETI
002 SWAP DROP * 2 +LIST
      "Meldung:"
004 dann ENTER."
      SWAP INPUT 3 SWAP
006 PUT
      > Bytes : 77.5
008 Checksum : # 7435h

48SX HOUR
< BLKMENU 2 GETI
002 SWAP DROP +STR -1 2
      +LIST
004 "Zeit (HH.MM):
      dann ENTER."
006 SWAP INPUT OBJ+ 2
      SWAP PUT
008 > Bytes : 85.5
      Checksum : # FF94h

```

```

48SX DATTE
<
002 DO BLKMENU 1 GETI
      SWAP DROP +STR -1 2
004 +LIST
      "Datum (DD.MMYYYY):
      dann ENTER."
006 SWAP INPUT OBJ+
      UNTIL DUP DUP DUP
008 IP DUP 0 > SWAP 32
010 < AND SWAP FP 100 *
      IP 13 < AND SWAP
012 100 * FP 10000 *
      1990 > AND DUP
014 IF NOT
      THEN SWAP DROP
016 CLLCD
      "Falsches Datum. Re
      gel:"
018 1 DISP 1 FREEZE
020 "0 < DD < 32" 3
      DISP 3 FREEZE
022 "0 < MM < 13" 4
      DISP 4 FREEZE
024 "1990 < YYYY" 5
      DISP 5 FREEZE
026 "Obere Taste drücke
      n"
028 7 DISP 7 FREEZE -1
      WAIT DROP
030 END
      END 1 SWAP PUT
032 > Bytes : 397.5
      Checksum : # B1EDh

48SX RFSH
< CLLCD 1 GETI
002 "Datum" SWAP + 1
      DISP 1 FREEZE GETI
004 "Zeit" SWAP + 2
      DISP 2 FREEZE GETI
006 "Msg." SWAP + 3
      DISP 3 FREEZE DROP
008 "erst Softkey druck
      en"
010 5 DISP 5 FREEZE
      SETU4
012 > Bytes : 146.5
      Checksum : # 8069h

48SX BEG
<
002 DO RFSH -1 WAIT
      DOK4
004 UNTIL DUP 15.1 ==
      SWAP 16.1 == OR
006 END
      > Bytes : 71
008 Checksum : # 8ABFh

48SX TOPC
< CLLCD
002 "ALARME ↔ PC" 1
      DISP 1 WAIT SETU6
004 > Bytes : 49
      Checksum : # 56F1h

48SX OVERALL
< DROP2
002 DO TOPC DOK6
      UNTIL 16.1 ==
004 END 'ALARME'
      PURGE
006 > Bytes : 65
      Checksum : # C8BCh

48SX DOK6
<
002 DO -1 WAIT
      CASE DUP 91.3
004 ==
      THEN OFF
006 END DUP 12.1
      ==
008 THEN DROP
010 'ALARME'
      IFERR CLLCD
      SEND

```

```

012 THEN DROP
      "NAME-FILE-I/O Prob
      lem"
014 1 DISP 1 FREEZE
016 "Check + retry" 2
      DISP 2 FREEZE
018 " OK" 2 DISP 2
      FREEZE
020 END 12.1
022 END DUP 11.1
      ==
024 THEN DROP
      AL+LI .1 1500 BEEP
026 CLLCD
      "File: ALARME ok !"
028 2 DISP 2 FREEZE
      11.1
030 END DUP 14.1
      ==
032 THEN DROP
      RECV CLLCD
034 "Liste OK !" 2 DISP
      2 FREEZE 14.1
036 END DUP 15.1
      ==
038 THEN DROP
      LI+AL .1 1500 BEEP
040 CLLCD "ALARME OK !"
      2 DISP 2 FREEZE
042 15.1
      END
044 END 1 WAIT
      UNTIL 16.1 ==
046 END 16.1
      > Bytes : 528.5
048 Checksum : # AD58h

48SX SETU6
< ("AL+LI" "SEND"
002 "" "RECV" "LI+AL"
      "ABRT" ) MENU
004 > Bytes : 69.5
      Checksum : # 37CDh

48SX AL+LI
< 1 'NA' STO
002 DO NA
      IFERR RCLALARM
004 THEN 0
      ELSE NA 1 +
006 END DUP 'NA'
      STO
008 UNTIL 0 ==
      END 1 - +LIST
010 'ALARME' STO 'NA'
      PURGE
012 > Bytes : 127
      Checksum : # 2E86h

48SX LI+AL
< "NAME" ( ( 1 0 )
002 α ) INPUT OBJ+ OBJ+
004 1 SWAP
      FOR i STORALARM
006 DROP
      NEXT
008 > Bytes : 65.5
      Checksum : # E87Fh

48SX CHKALRM
< 1 GETI TIME TSTR
002 1 12 SUB 4 DISP 4
      FREEZE DROP 3 GETI
004 SWAP DROP DUP TYPE
      2 ==
006 IF
      THEN SRCSTR POS
008 ELSE DROP 0
      END
010 IF
      THEN 1 'FNDALRM'
012 STO
      ELSE DROP
014 END
      > Bytes : 143
016 Checksum : # FA96h

```

```

48SX ENDPROC
<
002 IF ENDALRM
  THEN CLEAR CLLCD
004 "keine Termine gefu
  nden"
006 3 DISP 3 FREEZE
  "obere Taste drücke
008 n
  5 DISP 5 FREEZE -1
010 WAIT DROP
  ELSE 1 GET 'DSTR'
012 STO
  END
014 > Bytes : 144
  Checksum : # 9534h

48SX SRCMAIN
< GTSTR CLLCD
  "searching" 2 DISP
  2 FREEZE GTALN
004 SRCALRM ENDPROC
  > Bytes : 74.5
  Checksum : # C6FDh

48SX SRCALRM
< 0 'ENDALRM' STO 0
  'FNDALRM' STO
  DO NXTALRM
  IFERR RCLALARM
  THEN 1
006 'ENDALRM' STO
  ELSE CHKALRM
  END NXTALRM 1 +
008 'NXTALRM' STO
  UNTIL ENDALRM
010 FNDALRM OR
  END
  > Bytes : 174.5
  Checksum : # A721h

48SX GTALN
< DSTR FNDALARM
  'NXTALRM' STO
  > Bytes : 38
  Checksum : # 160Bh
  
```

```

48SX GTSTR
< DROP2 BLKMENU
002 "suchstring eingebe
  n
  dann ENTER drücken.
006 SRCSTR & 2 →LIST
  INPUT 'SRCSTR' STO
008 > Bytes : 105.5
  Checksum : # 848Eh

48SX CNTL
< DEPT →LIST
  'STACK' STO DATE
  'DSTR' STO REFRESH
004 DO -1 WAIT DOKEY
  UNTIL IP 16 ==
  END DROP2 STACK
  OBJ → DROP
008 > Bytes : 111.5
  Checksum : # 1A7Ah

48SX APPTS
< DEPTH DROPN
  IF FAPPTS
  THEN DEPTH ROLL
004 APS+MS PSTMSG
  ELSE NOAPPTS
  END
  > Bytes : 74
  Checksum : # 9E32h

WKFRST
52

WKLST
53

MOFRST
30.121991

DYO
31.121991

NDOY
366

FNDALRM
0
  
```

```

ENDALRM
1

NXTALRM
3

ROW
39

COL
95.0000000006

ADR
5

Day1
"PRI"

Yr
"1992"

Du
28

Mo
3

CST
( "FIND" "GOTO" "ADD"
  "UPLD" "APPTS" "stop" )

DSTR
28.031992

STACK
( )

flags
( # 30000000FF4h # 0h )

SRCSTR
"SPY"

MYFLGS
( # 300000003F4h # 0h )
  
```

Fortsetzung von Seite 52

**15. Datenbank**

Die Funktionen erlauben das Anlegen und Verwalten einer kleinen Datenbank inclusive Suchen und Sortieren.

Mit den Kalenderfunktionen des 48SX kann man jetzt seinen SF-7500 wegschmeißen (ohnehin die beste Lösung für diesen Murks).

**Fazit:**

Das Buch bietet (wenn die Preisangabe stimmt) eine preiswerte und umfangreiche Programmsammlung, die vor allem denen Freude machen wird, die sich mit Prüfungsaufgaben aus dem Bereich Mathematik und Numerik herumschlagen müssen.

Außerdem sollte das Buch all de-

nen zu Denken geben, die noch immer auf dem HP-41 programmieren, denn hier ist es einem Autor gelungen, eine große Zahl qualifizierter Programme für den 48SX zu schreiben.

Man kann den Umfang dieses Buches mit dem von "Best of PRISMA" vergleichen, obwohl das die Programme vieler Anwender von fast drei Jahren zusammenfasst, und trotzdem viel weniger bietet (als Autor der Kap. 1+4 darf ich das wohl schreiben).

Ein kleiner Tip für die Leser des Buches:

Wer sich oft von Großrechenanlagen ärgern läßt, wählt -wie der Autor- im MODES Menü des 48SX den Dezimalpunkt als Trennzeichen

(mit FM,) zwischen Vor- und Nachkommateil. Wer allerdings das Komma benutzt, kann in Namen wie z.B. C.COL keinen Punkt eingeben, da hier der 48SX bockt.

**Abhilfe:**

Vor der Eingabe des Namens auf Dezimalpunkt zurückschalten (das ist dann aber auch für RCL, VISIT oder Verwendung des Namens bei der Eingabe in andere Programme nötig) oder bei der Eingabe des Programms den Punkt durch ein Komma ersetzen (muß man dann auch in Programmen machen, die C.COL dann aufrufen sollen).

Ralf Pfeifer (116)  
 Rubensstraße 5  
 D5000 Köln 50

## SERVICELEISTUNGEN

### BEST OF PRISMA

Schutzgebühr: 10,- DM

### Nachsendedienst PRISMA

Schutzgebühr: 25,- DM für Jahrgänge 1982-86  
50,- DM 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

Erscheint von Fall zu Fall in PRISMA.

### 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.

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 Freiumsschlag)

## CLUBADRESSEN

### 1. Vorsitzender

N.N.

### 2. Vorsitzender

Alf-Norman Tietze (1909),  
Sossenheimer Mühlgrasse 10,  
6000 Frankfurt 80, ☎ (069) 34 62 40

### Schatzmeister

Dieter Wolf (1734),  
Pützerstraße 29, 6000 Frankfurt 90,  
☎ (069) 76 59 12

### 1. Beisitzer

Norbert Resch (2739),  
Tsingtauerstraße 69, 8000 München 82

### 2. Beisitzer

Werner Dworak (607),  
Allewind 51, 7900 Ulm,  
☎ (07304) 32 74

### 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 (2243), Im Winkel 1, 2061 Sülfeld,  
Horst Ziegler (2225), Schüslerweg 18b, 2100 Hamburg 90,  
☎ (040) 79 05 67 2

### Regionalgruppe Karlsruhe / Beirat

Stefan Schwall (1695), Friedrich Naumann Straße 54,  
7500 Karlsruhe 21, ☎ (0721) 74 878

### 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

Manfred Hammer(2743), Oranienstraße 42, 6200 Wiesbaden

### 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

### Beirat

Georg Hoppen (3623), Im Hubertusring 5, 4512 Wallenhorst

### Hardware 41

Winfried Maschke (413), Ursulakloster 4, 5000 Köln 1,  
☎ (0221) 13 12 97

### HP-71 Assembler (LEX-Files)

Matthias Rabe (2062), Teichsheid 13, 4800 Bielefeld,

### Mathematik

Andreas Wolpers (349), Steinstraße 15, 7500 Karlsruhe

### Programmbibliothek HP-71

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

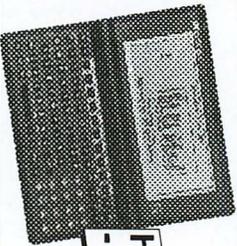
### "Clubadresse"

CCD e.V., Postf. 11 04 11, 6000 Frankfurt 1, ☎ (069) 76 59 12

# Sonderpreisliste für CCD Mitglieder

**HP 95LX PalmTop PC** 948,00 DM  
**PC Connectivity Pack** 178,00 DM  
 Summe Einzelpreise 1126,00 DM

**Im Paket 1018,00 DM**  
**Sie sparen 108,00 DM**



HP 95LX  
 - der PC in Ihrer Hand!

Zahlungsbedingungen: gegen Vorkasse oder per Nachnahme  
 jeweils zzgl. 15,- DM Versandkosten

21-04-92 Es gelten unsere AGB, Änderungen und Irrtum vorbehalten.

HP 95LX 1MB PalmTop PC 1548,00 DM  
 Mitsubishi 1MB Karte 548,00 DM  
 HP 48 S 498,00 DM  
 HP 48 SX 598,00 DM

Zubehör für HP 48 SX:

HP Serieller Anschluß an IBM 95,00 DM  
 HP Development Kit (82208B) 215,00 DM  
 HP Gleichungslöser Bibliothek 158,00 DM  
 HP 32 KB RAM Erweiterung 127,00 DM  
 HP 128 KB RAM Erweiterung 319,00 DM  
 256 KB RAM Erweiterung 548,00 DM  
 HP Programmer's Manual 38,00 DM

**HP DeskJet 500 948,00 DM**  
**HP DeskJet 500C 1598,00 DM**

Ihre Ansprechpartner:

Herr Endler  
 Herr Tubic



**H&G EDV Vertriebs GmbH**

Münsterstraße 1

☎ 0228 / 72 90 8-26 o.-27

• 5300 Bonn 1

• Fax 0228 / 72 90 8-38

