

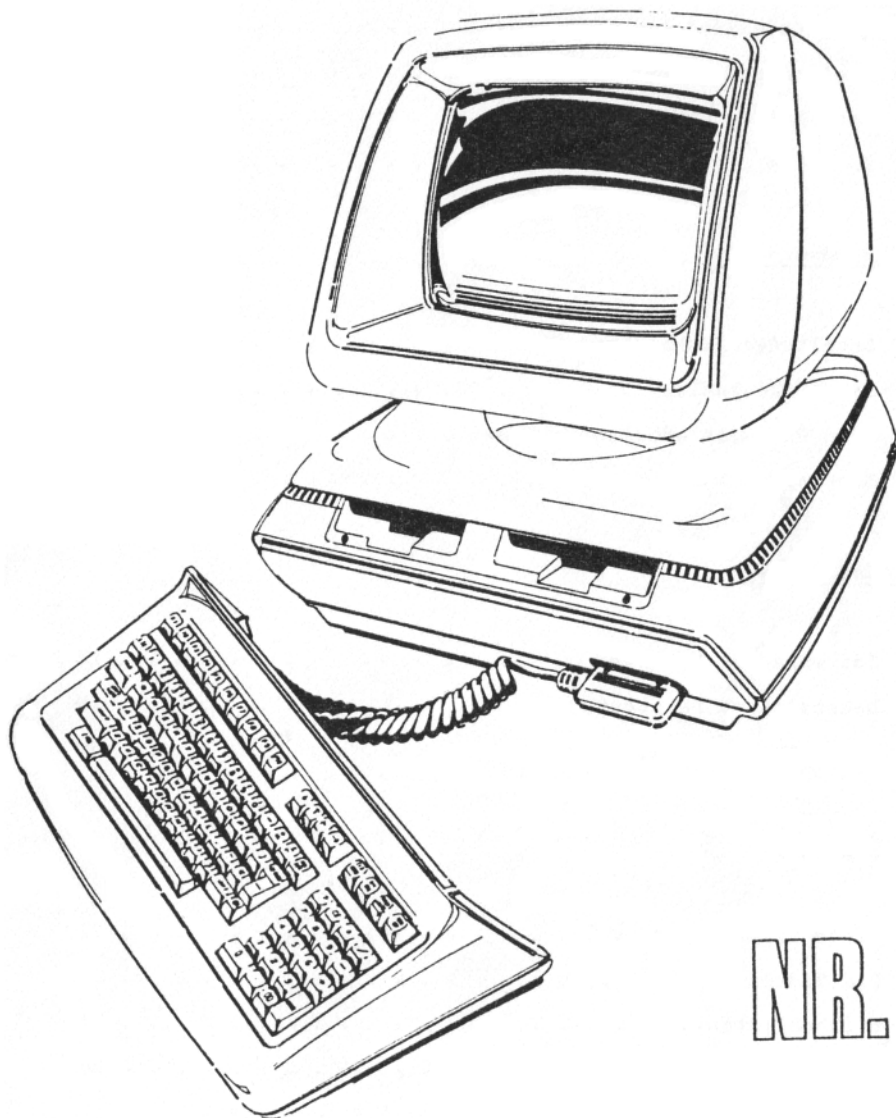
|                   |                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------|
| <b>Title:</b>     | <b>„Heute schon geust?!“ Nr. 3</b>                                                                    |
| Document name:    | hsg3.pdf                                                                                              |
| Version number:   | 1.0.0                                                                                                 |
| Format:           | DIN A5                                                                                                |
| Scanning, rework: | Ullrich von Bassewitz, uz@musoftware.de                                                               |
| PDF creation:     | Ullrich von Bassewitz, 2019-09-02 16:03                                                               |
| Download URL:     | <a href="ftp://ftp.musoftware.de/pub/uz/hsg/hsg3.pdf">ftp://ftp.musoftware.de/pub/uz/hsg/hsg3.pdf</a> |

# HEUTE SCHON GEUST ?!

protected widely throughout the world

hochoffizielles Organ der  
CBM Selbsthilfe e.V.(i.G.)

INTERNATIONAL



NR. 3

## Inhalt

|                                      |   |
|--------------------------------------|---|
| Vorwort . . . . .                    | 1 |
| Liebe Neulinge . . . . .             | 4 |
| ZU BEACHTEN . . . . .                | 5 |
| Hobby oder Arbeit . . . . .          | 6 |
| Arbeitsteilung und Adressen. . . . . | 7 |
| Regionale Treffen . . . . .          | 8 |
| Noch ältere CBM-Geräte . . . . .     | 8 |
| Zu haben . . . . .                   | 9 |

### Hardware:

|                                              |    |
|----------------------------------------------|----|
| Schaltungsbeschreibung CBM 600/700 . . . . . | 13 |
| Der dicke Fehler... . . . .                  | 15 |
| Dein PAL, das unbekannte Wesen . . . . .     | 16 |
| Echtzeit . . . . .                           | 17 |

### Datasette:

|                                  |    |
|----------------------------------|----|
| Software auf Kassette . . . . .  | 18 |
| Datensichere Datasette . . . . . | 19 |
| Super-Tape Korrekturen . . . . . | 20 |

### Video:

|                                    |    |
|------------------------------------|----|
| Bildschirmschoner . . . . .        | 21 |
| Die Videoinitialisierung . . . . . | 24 |
| Bildschirmfenster . . . . .        | 26 |
| Bildschirmflimmern . . . . .       | 28 |

## neue Projekte:

|                                                |    |
|------------------------------------------------|----|
| Sprachen . . . . .                             | 29 |
| Diskettenstationen . . . . .                   | 29 |
| Basic-Compiler . . . . .                       | 30 |
| Kernalverbesserung und -berichtigung . . . . . | 31 |
| Logikanalysator, Spracherkennung . . . . .     | 32 |

## Basic-Programmierung:

|                                            |    |
|--------------------------------------------|----|
| Kleine Tips und große Bitten . . . . .     | 33 |
| Tips und Tricks . . . . .                  | 34 |
| Nützliche Pokes . . . . .                  | 35 |
| Ein kühner Plan . . . . .                  | 36 |
| - Acht Kilo Spaghetti commentata . . . . . | 38 |
| - Tätetätä: Das Menue . . . . .            | 40 |
| - Listing . . . . .                        | 43 |

## Besprechungen:

|                                                         |    |
|---------------------------------------------------------|----|
| ema - Editor-Macroassembler . . . . .                   | 47 |
| Mal was anderes - FORTH . . . . .                       | 48 |
| Kalk Result - Ein Weichwaren-Rückblick . . . . .        | 49 |
| Textverarbeitungen . . . . .                            | 52 |
| Macro-Basic - Es muß nicht immer Pascal sein! . . . . . | 55 |
| ml01 - ein neues Betriebssystem für den 8028 . . . . .  | 57 |

## Zum Schluß:

|                               |    |
|-------------------------------|----|
| Flohmarkt . . . . .           | 59 |
| Vorschau auf Heft 4 . . . . . | 60 |
| Impressum . . . . .           | 60 |



### Lang ersehnt, mit wenig Hoffnung

haben viele von Ihnen dieses nun doch noch fertiggewordene Heft 3. Wäre es einige Wochen später geworden, könnte man es schon eher Jahrbuch nennen. Vom Umfang her kommt das auch ungefähr hin.

Ich fürchte, an dieser Stelle eine Erklärung schuldig zu sein. Vor über 20 Jahren habe ich meinen Lehrberuf Kaufmannsgehilfe aufgegeben aus Unlust am unkreativen Papierkram. Und was hat mir unser Verein beschert? Eine Papierflut ungeahnten Ausmaßes, derer ich schlicht und einfach nicht Herr werden konnte. Dabei handelt es sich hier auch noch um eine kostspielige Angelegenheit. Ein Zimmer meiner Wohnung ist mit Computerkram so voll, daß es praktisch nicht mehr zu benutzen ist, und ich mich zum arbeiten an den Küchentisch setzen muß.

Hinzu kommt, daß ich im August vorigen Jahres zum zweiten Male Vater geworden bin. Damit hat sich die räumliche Enge so zugespitzt, daß es an ein Wunder grenzt, daß meine Ehe das bis jetzt überstanden hat. Mit weltlichen Gütern bin ich nun mal nicht so sehr gesegnet. In den nächsten Wochen werde ich mit Andreas zusammen ein Büro für die Vereinsgeschichten einrichten, damit endlich ein ordnungsgemäßer Betriebsablauf herzustellen ist, und ich wieder ein Familienleben habe.

Ein weiterer Umstand, der meine Produktivität erheblich gemindert hat, ist der chronische Geldmangel. Ich habe mich schwer verkalkuliert mit der 20,-DM-Regelung für die Hefte 1-?. Dieses Heft und noch mehr ist voll aus eigener Tasche finanziert. Hoffentlich findet sich der eine oder andere bereit, sich an meinem Defizit zu beteiligen. Kaum einer kann sich vorstellen, wie mühsam es ist, wenn, um eine Bestellung zu erledigen, die einzelnen Sachen erst einmal hergestellt werden müssen, wie Photokopien oder Clubdisketten. Kann ich da nicht einfach auf Lagerbestände zurückgreifen, wird jede noch so geringe Kleinigkeit zur Last. So wird zunächst in Lagerbestände investiert werden müssen.

Der wohl wichtigste Punkt, den ich nicht geschafft habe, ist die Organisation der Arbeitsteilung. Aber die herzustellen, ist ein nicht allzu leicht realisierbares Unterfangen und auch wieder mit Arbeit verbunden. Einiges Wenige ist allerdings schon in die Wege geleitet, aber der Verein kann erst dann wirklich effizient arbeiten, wenn der größte Teil der Kommunikation von der "Zentrale" ferngehalten wird (siehe auch Seite Arbeitsteilung).

Nun aber zu dem, was geschafft worden ist.

#### Der Verein

Endlich ist es vollbracht, einen ordentlichen Verein zu gründen. Keiner in unseren Reihen scheint so ein richtiger Vereinsmeister zu sein, und für uns (in erster Linie Andreas) war es richtig Arbeit, eine Satzung zu basteln, die den Ansprüchen sowohl des Finanzamtes als auch des Registergerichtes

genügt. Nun also sind wir sogar gemeinnützig. Beiträge und Spenden sind somit steuerlich absetzbar.

Was die Finanzierung anbelangt, ist auf keinen Fall damit zu rechnen, daß der beschlossene Jahresbeitrag von DM 50,- die Kosten zu decken ausreicht. Wir sind also unbedingt auch auf Einnahmen aus dem Verkauf der Clubdisketten angewiesen. Daher sollte auch bei der Weitergabe von Clubdisketten mit freier Software (bitte nur an Mitglieder!) wenigstens ein Betrag von DM 5,- pro weitergegebene Diskette auf das Vereinskonto überwiesen werden. Sägt nicht an dem Ast, auf dem Ihr selber sitzt!

### Chicago B-Users Group und Commodore

Die Zusammenarbeit mit unserer großen amerikanischen Schwester ist abgemacht. Von den Beiträgen gehen pro Nase \$ 5,- an den CBUG, der schon wirklich viel erreicht hat, und dessen komplette Bibliothek uns zur Verfügung stehen wird. Wir zahlen für die nicht freien Programme daraus die selben Urheberrechtsabgaben wie CBUG und umgekehrt.

Die Firma Commodore hat Bereitschaft zur Zusammenarbeit signalisiert, und wir werden in Kürze versuchen, hier Regelungen zum gegenseitigen Nutzen zu treffen. Es ist anzunehmen, daß die Gemeinnützigkeit auch hier vorteilhaft ist.

### Bibliothek

Da niemand einen Anderen wußte, der diese Arbeit hätte erledigen können, habe ich mich viele Wochen damit beschäftigt, die hier eingetrudelten Disketten flüchtig auf eventuell brauchbare Programme zu untersuchen, und die Ausbeute in etwa thematisch zu ordnen. Schließlich ist die Bibliothek das Herzstück eines Computerclubs, und da hatten wir bislang doch ein reichlich schmales, wenn auch hochwertiges Repertoire. Nun, da ein hoffnungsvoller Anfang gemacht ist, suchen wir dringend jemanden, der hauptverantwortlich die Leitung der Bibliothek übernimmt.

### Grundsätzliche Probleme

Eigentlich bin ich von meinen Begabungen her nicht unbedingt der richtige Mann, die hier zu erledigenden Verwaltungsaufgaben zu erfüllen. Viel mehr habe ich die Fähigkeit, mich an Problemen festzubeißen und möglichst einfache Lösungen herbeizuführen. Zumindest, wo ich selbst keine Lösung finde, weil ich beispielsweise immer noch keinen 6502-Assembler beherrsche, kann ich Probleme und prinzipielle Lösungswege exakt genug definieren, daß andere Spezialisten rasch zu überzeugenden Lösungen kommen können. Hier hat sich insbesondere die Zusammenarbeit mit Andreas Stankewitz und Dirk Seifert bewährt.

Einige Ideen und Fehlerkorrekturen, die so zusammengekommen sind, sind in diesem Heft wiedergegeben. Die Diskmonitor-Demoversion ist als Programm nicht ernstzunehmen, insbesondere auch stilistisch. Ich hatte einfach nur ein paar Ideen,

Menuegestaltung, absolut definierbare Fenster und relativ schnelle Codewandlung in ein Programm packen wollen, und so habe ich einfach drei Tage lang drauflosgetippt. Die Benutzerführung ist bei den meisten unserer Bibliotheksprogramme ziemlich mangelhaft, und ich meine, das von mir nachprogrammierte Menue ist geeignet, so manches Programm aufzuwerten. Ich war selbst überrascht, wie gut es funktioniert.

So bald die Arbeit an diesem Heft beendet ist, werde ich mich mit Dirk Seifert darangeben, die weiter hinten kurz angesprochene Diskutility zu basteln (siehe Artikel 'Ein kühner Plan'. Einem anderen grundlegenden Ärgernis bei der Benutzung unserer Computer, der unwahrscheinlich lahmen Peripherie, werden Beine gemacht.

### Wie geht es weiter?

Am Ende, wenn auch spät, haben wir doch einiges vorzuzeigen. Damit es nicht noch einmal so lange dauert, bis Sie etwas davon merken, was so alles passiert, richten wir, Andreas und ich, uns, wie erwähnt, einen ordentlichen Arbeitsplatz ein. Damit ist die wichtigste Voraussetzung für die Arbeit erfüllt. Andreas hat schon angefangen, etwas ganz Modernes zu vollziehen, nämlich die Umstellung der Vereinsverwaltung auf EDV. Ist das nicht ein Knüller, so haben wir immer wen, der alles schuld ist. Bitte macht regen Gebrauch von der Möglichkeit, sich als Ansprechpartner bei Fragen zu bestimmten Problembereichen zur Verfügung zu stellen, damit wir uns hier auf die Arbeiten konzentrieren können, die für die Allgemeinheit zu tun sind. Hieran hängt definitiv die Existenz des Clubs und die Erscheinungsweise von "HSG?!" Was letztere anbelangt, haben wir das nächste Heft schon etwa die Hälfte zusammen. Unbedingt gehört hier aus besagten Gründen eine stattliche Liste von Ansprechpartnern hinein.

### Vereinsmeierisches

Bald hätte ich es vergessen. Bei der Gründungsversammlung ist ja gewählt worden: 1. Vorsitzender Gerd Schumacher, 2. Vorsitzender Jörg Linnig, Kassenwart Andreas Stankewitz (gehört zum Vorstand), Schriftführer Thomas Greve, Kassenprüfer Willem van Tongeren und Arnold Mertineit.

Beschlossen wurde außerdem, daß Menschen, die sich nicht durch Beitragszahlungen an den Gemeinkosten beteiligen, auch keine Leistungen des Vereins (Clubdisketten) zustehen, und daß die Zeitschrift von gewerblichen Inseraten frei bleiben soll. Der Jahresbeitrag ist auf DM 50,- und der Preis für Disketten mit freier Software auf DM 15,- festgesetzt worden. Auf Wunsch ist die Satzung zu haben, aber es steht wirklich nichts Besonderes drin.

Damit Sie nicht einschlafen, geht es jetzt zur Sache.

Viel Vergnügen

## Liebe Neulinge

Da dieses Heft (bis zum Fertigwerden eines speziellen Neu-  
lings-Heft) an alle geht, die sich neu für unseren Verein  
interessieren, möchte ich hier ein Geleit für eben diese Neu-  
linge geben.

Vor eindreiviertel Jahren wurden von Völkner zu Tiefstpreisen  
Commodores Restbestände an 600/700-Computern verkauft, Anlaß  
für viele, sich dieser Rechner anzunehmen. Da es in dieser  
Anfangsphase außer einem Schaltplan keinerlei Dokumentation  
für die Rechner gab, sahen sich die meisten schnell in der  
Situation, zwar einen scheinbar guten Computer zu besitzen,  
aber nichts damit anfangen zu können.

Um dem Informationsdefizit entgegenzutreten, haben sich  
einige Personen mittels Kleinanzeigen als Kontaktpersonen  
bloßgestellt, mit dem Ziel, vielleicht ein paar Gleichgesinn-  
te zu finden. Irgendwann lief alles bei Gerd Schumacher  
zusammen, und im Nu war die Kartei auf mehrere hundert Leute  
angewachsen und die Sache uns (insbesondere Gerd) über den  
Kopf gewachsen.

Als Antwort auf die Flut von Anfragen schrieb Gerd die 'Heute  
schon geust?! Nr. 0', auch als Vorschlag, wie eine Clubzeit-  
ung aussehen könnte. Für weitere Hefte wurde ein Kostenbei-  
trag von DM 20.- erhoben, und es folgten Heft 1 und 2, dieses  
ist das dritte.

Im Laufe der Zeit trudelten einige hundert Programme bei uns  
ein, leider selten mit Beschreibungen, und diese zu sichten  
und zu sortieren dauerte so lange, daß einige schon meinten,  
wir hielten sie vor der Allgemeinheit zurück.

Im September letzten Jahres wurde dann ein Verein gegründet,  
der gerade ins Vereinsregister eingetragen wird und bereits  
als gemeinnützig anerkannt ist. Damit sind wir jetzt auch  
offiziell in der Lage, im Vereinsinteresse Geschäfte zu täti-  
gen, so daß nicht mehr alles was Geld braucht auf Kosten von  
Gerd geht.

Nach viel zu langer Zeit (siehe Vorwort) haben wir es nun  
endlich geschafft, dieses Heft 3 herauszubringen, das das  
letzte der "20-Mark-Serie" ist. Mit diesem beginnt der Verein  
mit seinen "geschäftlichen" Tätigkeiten.

Unter anderem aus rechtlichen Gründen wurde beschlossen, daß  
Leistungen des Vereins nur an Mitglieder erbracht werden, so  
daß wir darum bitten, erst bei uns Mitglied zu werden (siehe  
Beitrittserklärung), bevor Sie irgendwelche Sachen bei uns  
bestellen.

Wir hoffen, daß Sie sich in diesem Heft gut zurecht finden.  
Wir stehen natürlich auch für Fragen zur Verfügung. Haben Sie  
aber bitte Verständnis, wenn wir telefonisch nur zu bestimm-  
ten Zeiten erreichbar sind und schriftliche Anfragen nicht  
immer oder bei allgemeinem Interesse im nächsten Heft beant-

worten.

Wir würden uns freuen, wenn Sie sich an unseren Arbeiten mit beteiligten, zu tun gibt es genug (z.B. Kontaktperson für bestimmte Anfragen, Artikel-Autoren, Disketten-Paten...).

### ZU BEACHTEN !!!

Sämtliche Leistungen des Vereins können ab sofort nur noch Mitgliedern erbracht werden (Beschluß der Gründungsversammlung). Es reicht aber, wenn uns gleichzeitig mit einer Bestellung die Beitrittserklärung zugeschickt und natürlich der Beitrag bezahlt wird. Bestellungen von Nicht-Mitgliedern werden künftig ignoriert (außer Typenräder und Farbbänder, da hier sonst die erforderlichen Stückzahlen nicht erreicht werden können).

Bei allen Diskettenbestellungen ist unbedingt das gewünschte Format anzugeben. Wenn kein Format angegeben ist, dann wird 8250-Format (SFD 1001) geliefert, ein Umtausch wegen falschem Format ist dann nicht möglich!!

Zwischen 8050 und 8250-Format ist zu unterscheiden (wegen REL-Files)!!

Für 4040-Laufwerke liefern wir 1541-Disketten (meist zweiseitig zum umdrehen). Diese sind nur lese-kompatibel und sollten nicht mit der 4040 beschrieben werden.

Für den Verein haben wir ein neues Konto eröffnet. Alle alten Konten sollen nicht mehr benutzt werden. Bis wir das Vereins-siegel des Registergerichts in der Hand halten läuft das Konto auf den Namen Gerd Schumacher. Das neue Konto heißt also:

CBM-Selbsthilfe,  
c/o Gerd Schumacher,  
KNr. 3443 57-503  
BLZ. 370 100 50

Fragen nehmen wir natürlich immer noch entgegen. Wir bitten aber weiterhin, daß unsere Sprechzeiten berücksichtigt werden.

Gerd Schumacher:  
Donnerstags 9:00 bis 22:00 Tel: 0221/448368

Andreas Stankewitz:  
Werktäglich 15:00 bis 16:00 Tel: 0221/230055  
(Ist aber nicht immer zu Hause).

Urlaub haben wir dringend nötig. Darum werden wir in der Zeit vom 16. Juli bis 21. August Bestellungen nur verzögert oder teilweise erst danach bearbeiten können.

Wir haben im Verein ein unglaubliches Potential an Fachleuten in allen Gebieten der Computerei, und gelegentlich wird hier auch wirklich einiges Beachtliche geleistet. Das meiste steht fast umsonst zur Verfügung, entweder hier durch die Zeitung oder z.B. auf Clubdisketten. Ob nun ein Entwickler aus unseren Reihen für seine Produkte Geld verlangen darf, und sogar, ob der Verein überhaupt ehemals kommerzielle Produkte für kleines Geld vertreiben sollte, darüber gehen die Meinungen auf befremdliche Art auseinander.

Einige Leute, besonders solche, die bislang noch keinem normalen Existenzkampf ausgesetzt waren, tendieren zu der Meinung, daß Software beispielsweise überhaupt nichts wert sei. Bitteschön, mögen sie doch selbst umsonst arbeiten gehen und sich meinethalben aus Mülltonnen ernähren. Grundsätzlich gehört Arbeit bezahlt, oder? Natürlich fließen die Grenzen zwischen Hobby und Arbeit, und werden individuell sehr verschieden gesetzt, auch in Abhängigkeit von der Bedürftigkeit. Mir ist herbe Kritik zu Ohren gekommen wegen des Preises für das neue Betriebssystem für den Drucker 8028 von Andreas. Er hat hieran über ein halbes Jahr intensiv gearbeitet in einem Maße, das nicht nur nach seinem Ermessen den Rahmen einer hobbymäßigen Beschäftigung übersteigt. Um für diese Arbeit und die für den Verein überhaupt Zeit zu haben, hat er einen Job an der Uni, der immerhin DM 800,- monatlich eingebracht hat, aufgeben müssen. Wohnen und essen muß er aber trotzdem. Wem wäre geholfen, gäbe es das neue Betriebssystem nicht?

Die hoffentlich vorherrschende Absicht, die Freude an eigenen Programmen mit anderen teilen zu können, bringt uns ja gemeinsam ganz schön weiter. Wenn man aber beim programmieren die Absicht verfolgt, daß etwas dabei herauskommt, das unter allen denkbaren Bedingungen (z. B. Fehleingaben, Grenzwert-Über- und -unterschreitungen, Floppyfehler) von Jedermann leicht zu benutzen ist, dann ist das schon eine ganz andere Sache, die zumeist mit richtig Arbeit zu tun hat, und zwar mit unangenehmer. Spätestens beim Verfassen einer ordentlichen Anleitung hört die reine Programmierfreude meistens auf.

Was kommerzielle Software anbelangt, hierauf können wir meines Erachtens nicht völlig verzichten, und die Gesetzgebung ist hier ebenso klar wie die moralische Haltung eines anständigen Menschen. Ich glaube, der Mehrzahl der Mitglieder ist damit gedient, daß wir zu lächerlichen Preisen so hervorragende Programme wie Protext, Macro-Basic und den EMA unters Volk bringen. Schließlich ist es eines unserer vornehmsten Ziele, hochwertige Werkzeuge zur Programmierung und für den Erfahrungsaustausch zu verbreiten.

Aber bitteschön, wenn jemand etwas Gleichwertiges umsonst anbietet, die Früchte kostenloser Arbeit ernten wir alle liebend gern. Es sei an dieser Stelle dringend ermahnt, die Clubarbeit nicht durch ungesetzliche Machenschaften zu untergraben.

## Arbeitsteilung

Einiges in Sachen Arbeitsteilung ist schon angekocht. Eine Erweiterung derselben ist dringend erforderlich. An verschiedenen Stellen dieses Heftes sind Hinweise auf einzelne Beteiligungsmöglichkeiten zu finden. Zum Behufe der Erfassung Ihrer Möglichkeiten ist ein kleines Formular beigelegt, von dem wir uns eine ganze Menge erhoffen. Im nächsten Heft wird dann eine Liste mit den entsprechenden Ansprechpartnern erscheinen. Zunächst ist die Geschichte so weit gediehen, daß die Clubdisketten zwecks Überarbeitung und Zusammenfassung von Kritiken jeweils einen "Paten" erhalten haben. Andreas Stankewitz wird sich um die Verkäufe und den Verwaltungskram kümmern.

Hier schon mal eine kleine Liste von Ansprechpartnern zu einigen Themenkreisen:

- Macro-Basic : Jörg Linnig, An der Pulvermühle 35  
5000 Köln 91, Tel. 0221/836086
- Hardware : Gerd Schumacher, Palanterstr. 12d,  
5000 Köln 1, Tel. 0221/448368 (nur Do.).
- Monitore : Detlef Kirch, Alt-Gatow 41,  
1000 Berlin 22, Tel. 030/3615061.
- Profitext : Rolf Hustedt, Hüttenstr. 3,  
5900 Siegen, Tel. 0271/89393.
- Assembler I, Supertape, 8028 (soft) :  
Andreas Stankewitz, Lindenstr. 15, 5000 Köln 1  
Tel. 0221/230055 (nur werktags 15:00 - 16:00).
- Kernal : Dirk Seifert, Arienheller 5,  
5456 Rheinbrohl, Tel. 02635/5253.
- Protex : Michael Schuster, Haintalstr. 20,  
6000 Frankfurt 56, Tel. 06101/48109.
- DFÜ : Thomas Hingott, Im Hilgersfeld 10,  
5060 Bergisch-Gladbach, Tel. 02204/67326.
- Flohmarkt : Ingo Lerch, Wasserkrüger Weg 47  
2410 Mölln 1
- Übersetzungen: Axel Schröder, Steinbrecherstr. 30,  
3300 Braunschweig, Tel. 0531/330192

### weitere Adressen:

Jürgen Krause, Bergstr. 1, 6931 Daisbach.

Günther Brühl, Steinerstr. 16, 7958 Laupheim.

Rüdiger Klenner, Fr. v. Hardenberghaus B 406,  
Learheidestr. 26, 4630 Bochum 1.

## Regionale Treffen

Ich weiß nicht, woran es liegt. Das Interesse an persönlichen Kontakten scheint ziemlich wenig ausgeprägt zu sein. In Hamburg hat mal eines stattgefunden, über das ich nichts Genaues weiß. Vielleicht findet sich ein neuer Ansprechpartner, der so etwas zuwege bringt. In Braunschweig passiert eine ganze Menge. Hier in Köln hatten wir Pech mit den Lokalitäten und die blödsinnige Vereinbarung, uns jeden vierten Samstag des Monats zu treffen. Das hat einige Verwirrung gestiftet, da manche Monate fünf Samstage haben, und gelegentlich der vierte als dritter in Erinnerung geblieben ist, worauf man sich dann am dritten traf. Daher jetzt eine weniger verwirrende Vereinbarung. Wir treffen uns fortan jeden letzten Samstag des Monats in der Gaststätte und Hotel "Schwanenstübchen", Thieboldsgasse 133. Die Thieboldsgasse geht vom Neumarkt ab (Süd-Seite, auf der Ecke befindet sich eine KVB-Geschäftsstelle).

Wegen der Urlaubszeit findet das nächste Treffen aber erst am 24. Sept. 1088 um 15:00 statt.

### Frankfurt

Es war schon für das Frühjahr geplant, mal ein Treffen in Frankfurt zu organisieren, auch für die Mitglieder weiter südlich in der Diaspora. Angesichts der anbrechenden Urlaubszeit scheint es ratsam, das auf den September zu verlegen. Zwei, drei Leute vom kölnen harten Kern würden gerne daran teilnehmen. Wer Lust hat, dorthin zu kommen, möge sein Interesse bekunden. Es erfolgt dann eine ganz persönliche Einladung, wenn es so weit ist. Das macht ja nun dank EDV keine große Mühe mehr.

### noch ältere CBM-Geräte

Gerd Schumacher

In der Praxis hat sich gezeigt, daß, obwohl wir uns in Sachen 600/700 zusammengeschlossen haben, hier eine nennenswerte Menge Anfragen bezüglich der CBM-8032/8096/8296 eingeht, auch im Rahmen der mit Commodore anlaufenden Zusammenarbeit. Sicherlich haben viele Mitglieder mit diesen Geräten ausgedehnte Erfahrungen. Hier im engeren kölnen Kreis hat die fast niemand. Was sollen wir da tun? Zum einen sind mittlerweile doch viele unserer Mitglieder auch stolze Besitzer solcher Geräte, und wir haben ganz sicher Unmengen von Programmen dafür, zum anderen scheue ich zusätzliche Arbeit, da wir uns über mangelnde Auslastung beileibe nicht beklagen können.

Ich schätze, durchschnittlich einmal am Tag mit Fragen zu diesen Geräten konfrontiert zu werden, und ich kann nur selten weiterhelfen. Gefragt sind also Experten, die bereit sind, solcherlei Anfragen zu beantworten. Wenn sich das ein paar Leute aufteilten, bedeutete das keinen nennenswerten Aufwand. Spitze wäre natürlich, wenn wir auch für die alten Geräte Clubdisketten zusammenstellen würden. Ich habe dafür bestimmt ein paar Megabytes ungenutzt herumliegen. Wer stürzt sich drauf oder kann sonstwie weiterhelfen?

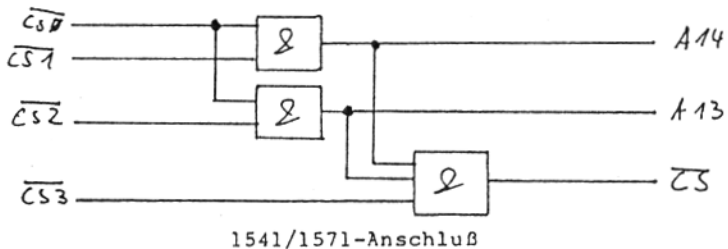


### Neues Cartridge-Modul

Es gibt ein neues RAM-Modul für die Bank 15. Es handelt sich um eine doppelseitige, nicht durchkontaktierte Leiterplatte von ca 40\*45 mm, die mit einem 32k-RAM und einem 74LS10 bestückt wird. Ohne weitere Maßnahmen sind 24k, der normale Cartridge-Bereich verfügbar. Zusätzlich, mit einer weiteren Verbindung, kann entweder der Bereich \$0800 - \$0FFF, \$1000 - \$1FFF zusätzlich benutzt werden. Wenn man das interne RAM ausblendet, kann man auch die gesamten unteren 32k der Bank 15 benutzen.

Zur Vereinfachung des Layouts - es gibt keine einzige Durchkontaktierung - sind die Daten- und die Adreßleitungen etwas durcheinandergeraten, wodurch die Verwendung von EPROMs hier ausgeschlossen ist. Zu meiner Schande muß ich gestehen, daß ich nicht an die Leute gedacht hatte, die ohne EPROM erst gar nichts machen können. Unser Mitbenutzer Wim van Tongeren hat sich vorgenommen, bis zum Herbst eine Hybridschaltung mit 32 k batteriegepuffertem RAM zu entwickeln, die dann statt des normalen eingesetzt werden kann. Damit wäre dann auch den Kassettenbetreibern gedient. Die Platine kostet unbestückt, aber gebohrt DM 5,-, voll bestückt DM 40,- (siehe Bestellschein). Leider sind die Speicher derzeit wieder ziemlich teuer.

Wie man aus vier exklusiven CS-Leitungen eine und zwei Adressen generiert:



Rolf Hustedt hat etwas entwickelt, um diese Floppies an den 600/700 anzuschliessen. Es handelt sich um ein EPROM für den Adreßbereich \$0800 bis \$0FFF und ist bei Rolf Hustedt zu haben.

### FORTH

Forth ist wohl nach C die schnellste Hochsprache, da es sehr maschinen-nah ist. Es ist stackorientiert (wie beispielsweise ein HP-Taschenrechner) und daher nicht so leicht durchschaubar. Hat man sich aber einmal daran gewöhnt, ist es schon eine phantastische Sache. Es funktioniert praktisch so, daß man sich aus den vorhandenen Befehlen eigene zusammenbaut

und kompiliert. Diese sind dann ganz normal wie der Grundwortschatz anzuwenden, etwa vergleichbar mit Macros in einem Assembler. Man schreibt sich sozusagen seine eigene Sprache.

Dr. Wolfgang Kessel hat FIG-Forth angepaßt, das unter dem Emulator 8432 läuft und als Clubdiskette verfügbar ist. Die Diskette enthält außerdem eine ordentliche Einführung in Forth, damit man, um diese Sprache kennenzulernen, sich nicht gleich in Kosten für Literatur stürzen muß. Die vorliegende Version hat noch ein paar Macken. Eine verbesserte Version ist im Herbst zu erwarten.

Rüdiger Klenner hat eine nach eigenem Bekunden vollständigere Forth-Anpassung vorgenommen, die im Normalmodus 600/700 in der Bank 15 läuft. Für Programm stehen hier zwar auch nur 16 k zur Verfügung, aber andere Bänke können für Texte benutzt werden. Rüdiger hat hier sehr viel echte Arbeit investiert. Unter anderem hat er noch Fehler im originalen Forth-Interest-Group-Quelltext beseitigt. Wichtig ist noch, daß zu diesem System sowohl ein Assembler als auch ein Recompiler gehören. Interessenten mögen sich direkt an Rüdiger Klenner wenden.

#### EPROG 27011

Einige von uns haben sich auf Verdacht den EPROG 27011 von Völkner zugelegt. Nun, nachdem sich manch einer die Zähne daran ausgebissen hat, bietet Rolf Hustedt eine selbstentwickelte Treibersoftware dazu an, sowohl zum EPROMS kopieren als auch zur Programmierung eines ganz neuen EPROMS.

#### Shugart-Bus-Floppy am 600/700

Jürgen Krause hat einen Shugart-Bus-Anschluß für den CBM 600/700 entwickelt. Auch eine Software, sie zu betreiben steht Interessenten zur Verfügung. Die Geschichte läuft zwar meinen Bemühungen um weitgehende Kompatibilität etwas zuwider, aber dennoch dürfte manch einem damit geholfen sein. Die zugehörige Software hat nichts mit der normalen CBM-mäßigen Floppyhandhabung zu tun, und ist daher nur eingeschränkt von Assembler-Programmierern verwendbar.

#### Basic umschaltbar

Auch die Mitglieder des CBUG, unserer großen Schwester, sind in Sachen Schaltungserweiterungen nicht untätig. Für unsere Verhältnisse aber sind die dortigen Entwicklungen recht teuer. Wer noch immer nicht das Basic ins parallele Mbyte-RAM laden kann gemäß meinem Vorschlag in Heft 1 Seiten 4/5, der kann von Gary Anderson eine Schaltung beziehen, die es gestattet, zwei Basic-Versionen softwaremäßig zu selektieren, deren eine (originale) fest im EPROM liegt, und deren andere entweder auch in einem EPROM, in einem EEPROM oder aber von Diskette ladbar in zwei statischen RAMs liegt. Der Preis beträgt US \$ 119,95.

## serieller C-128-Bus

Ebenfalls von Gary Anderson ist eine Hardware entwickelt worden, die den Betrieb der 1571 im schnellen Originalmodus am 600/700 ermöglicht. Der Preis beträgt US \$ 65,-. Zum Betrieb dieser Schnittstelle gibt es eine CBUG-Clubdiskette zu \$ 35,-, die bald von hier zu beziehen sein wird.

## technische Unterlagen

und Handbücher zu den alten Commodore-Geräten haben wir hier ziemlich vollständig angesammelt, um uns im Notfall auch schon mal über andere Geräte ein wenig schlau zu machen, um vernünftige Auskünfte erteilen zu können. Diese Unterlagen stehen den Mitgliedern auf Wunsch zur Verfügung. Bei Bedarf anrufen!

## Farbbänder 8028

sind manchmal schwer erhältlich. Wir haben ein gutes Angebot für Multistrike-Farbbänder. Wenn wir je Lieferung 100 Stück abnehmen, können wir sie zu DM 5,- weitergeben (siehe Bestellschein).

Wer über den Bezug der folgenden Sachen näheres wissen möchte, möge sich des Formulars am Ende des Heftes bedienen.

## Typenräder 8028 und 6400

Leider warten wir derzeit noch immer auf eine Lieferung Typenräder für den 8028, die Anfang November bestellt worden ist. Da es keine andere Quelle gibt, müssen wir uns damit abfinden. Mir geht das ganz schön an die Nerven, meinerseits wieder einen Haufen Leute vertrösten zu müssen. So etwas bindet ganz schön Energie. Trotzdem wird wohl nichts Anderes übrig bleiben, als sich weiter zu bemühen.

Eine größere Menge Typenräder für den 6400 gibt es auch noch für etwa DM 20,-. Ich werde noch mal nachhören, um welche Sorten es sich handelt, und kann dann in Kürze Auskunft geben, wenn sich jemand interessiert.

## SFD 1001    CBM 500    CBM 700    DIN-Tastenkappen

Leider hat es noch nicht geklappt, Diskettenstationen aus den USA zu besorgen. Es ist aber noch eine kleine Menge deutscher SFDs in neuen Einzelteilen aufgetaucht, die von mir montiert, gründlich geprüft und zum Preis von DM 450,- abgegeben werden können.

Eine begrenzte Anzahl DIN-Tastenkappen zu DM 40,- zzgl. Porto kann ab sofort beim Verein bestellt werden. Die Dinger sind so teuer, daß wir hier selbst auf einen kleinen Aufschlag als Anteil an den Gemeinkosten verzichtet haben, und es ist wohl

nicht angemessen, daß das Porto, wie in der Vergangenheit des öfteren von mir privat oder jetzt vom Verein übernommen wird.

Neue CBM 500 sind zu DM 75,-, CBM 700 zu DM 400,- zu haben. Für beide Geräte ist Vorkasse per Verrechnungsscheck notwendig. Nähere Information auf Anfrage.

#### EPROMMER

Es gibt zwar schon einige Eprommer für unsere Geräte, aber bislang noch keine so vollständige Komplettlösung wie die von Bernd Zimmermann. Eigentlich hatte er mit seiner Entwicklung Geld verdienen wollen. Nun aber bietet er das Gerät im Club an. Die ersten Seriengeräte werden spätestens Ende Juli fertig sein. Bei der Entwicklung wurde großer Wert auf Kompatibilität mit allen bestehenden und zu erwartenden Hardware-Erweiterungen gelegt. Die Software liegt im Cartridge-Bereich entweder als Eprom auf der Prommer-Platine oder eben in einer RAM-Erweiterung. Der Prommer wird über das weiter ausdekodierte CSI-Signal angesprochen. Die Basisadresse ist mit Jumpfern einstellbar, sodaß CSI auch für weitere Peripherieanschlüsse verfügbar bleibt.

Die Verbindung mit dem Computer erfolgt am 60-poligen Erweiterungsstecker. Hier sind zwei mechanische Lösungen vorgesehen. Entweder liegt die Platine draußen und wir über ein Flachbandkabel mit zwei Buchsenleisten verbunden. Das ist leider ziemlich teuer. Oder es wird eine Buchsenleiste eingelötet, die Platine mit der Bestückungsseite nach unten einfach intern aufgesteckt und bloß der Programmiersockel nach draußen geführt. Diese auf meinen Wunsch hinzugefügte Möglichkeit ist erheblich billiger, elektrisch sicherer und handlicher.

Es können Eproms von 2716 bis 27512 programmiert werden mit den softwaremäßig einstellbaren Spannungen 12,5 V und 21 V. Die Schaltung ist sowohl im digitalen als auch im analogen Teil definitiv nach den Regeln der Kunst aufgebaut und professionell gefertigt. Die zugehörige Software, deren Quellcode zum Lieferumfang gehört, unterstützt den Intel-Schnellprogrammialgorithmus und den Intel-Normalalgorithmus. Die Bedienung erfolgt menuegesteuert.

Alles in allem handelt es sich um ein Gerät ganz nach meinem Geschmack, das nach dem Prinzip einstöpseln und läuft zuverlässig auch von Leuten zu handhaben ist, die nichts mit Hardwarebasteleien zu tun haben wollen. Wegen seines hohen Datendurchsatzes ist das Gerät durchaus zum ernsthaften arbeiten tauglich. Ein EPROM 27512 komplett einlesen, Prüfsumme bilden, anzeigen, noch mal vergleichen dauert 7 Sekunden, das selbe mit einem 2764 eine Sekunde!

Endgültiges kann ich über den Preis noch nicht sagen. Ein Fertiggerät wird etwa DM 135,- kosten. Je nach Ausführung gibt es natürlich gewisse Abweichungen. Interessenten mögen Einzelheiten erfragen.

Dies ist die erste Folge einer vollständigen Schaltungsbeschreibung unserer Computer. Aus gegebenem Anlaß fange ich nicht in der logisch "richtigen" Reihenfolge mit Takt und CPU sondern mit der RAM-Adressierung und der Speicherauffrischung an. In diesem Schaltungsteil habe ich nämlich den wahrscheinlich einzigen echten Fehler gefunden, der verheerende Auswirkungen zeitigen kann.

Natürlich soll dies keine Einführung in die digitale Schaltungstechnik werden, und ich hoffe, für die Mehrzahl der Leser das richtige Maß an Grundkenntnissen vorauszusetzen. Anregungen sind herzlich willkommen.

Die in unseren - wie in den meisten Computern - verwendeten dynamischen Speicher vom Typ 4164 (bzw. 41256 bei Erweiterung über 256k hinaus) haben insgesamt  $2 \text{ hoch } 16 = 65536$  (bzw.  $2 \text{ hoch } 18 = 262.144$ ) Speicherstellen. Zu deren Adressierung benötigt man also 16 (bzw. 18) Adresssignale. Wie aus dem Schaltplan oder auch Abb. 1 ersichtlich, verfügen die RAM-Bausteine aber nur über die Hälfte der ansich erforderlichen Adressanschlüsse.

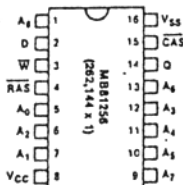


Abb. 1

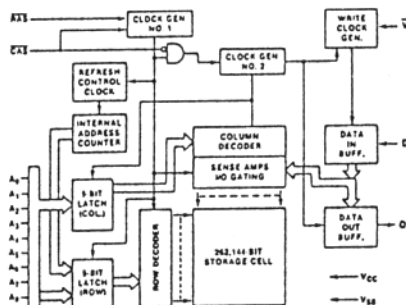


Abb. 2

Zur vollständigen Adressierung aller Speicherstellen müssen die Adresssignale also in zwei Portionen zugeführt werden. Die Speicher sind intern in einer Matrix angeordnet. Wie auf einem Schachbrett die Felder über Ziffern und Buchstaben "adressiert" sind, werden auch die Speicherzellen über ihre Spalten- und Zeilenadresse angesprochen. Im allgemeinen wird den Speichern zuerst die Zeilenadresse zugeführt. Kurz nachher muß das RAS-Signal von 1 nach 0 wechseln, um anzuzeigen, daß dieser Adressteil gültig ist. RAS bewirkt somit die bausteininterne Zwischenspeicherung der Zeilenadresse.

Anschließend wird über die selben Anschlüsse die Spaltenadresse zugeführt und kurz darauf auch deren Gültigkeit durch das Signal CAS angezeigt. Die nun anliegenden Zeilen- und Spaltenadressen werden nun dekodiert, das heißt, daß für

jeden der 256 Zustände, die ein 8-Bit Adreßteil annehmen kann, ein gesondertes Signal erzeugt wird. So ergeben sich in der Matrix  $256 \times 256 = 65536$  Kreuzungspunkte. Die Signale D (Dateneingang), Q (Datenausgang) und W (Schreiben) bedürfen wohl keiner Erklärung.

In den dynamischen Speichern werden die Informationen in winzigen Kondensatoren gespeichert. Diese verlieren ihre Ladung sehr rasch, und daher muß diese in sehr kurzen Zeitabständen aufgefrischt werden. Das heißt alle 2 (64k) bzw. 4 (256k) Millisekunden muß zum Datenerhalt jede einzelne Speicherstelle angesprochen werden. Zu diesem Zweck genügt es, nur eine halbe Portion Adressen anzulegen und deren Gültigkeit anzuzeigen. In unserem Falle handelt es sich hierbei um die Zeilenadresse.

In zwei Millisekunden müssen also alle 256 Zeilenadressen an die Speicher gelangen. Bei unserem Systemtakt von 2 MHz muß also jeder zwanzigste Taktzyklus der Speicherauffrischung geopfert werden. Der Prozessor muß solange warten.

#### Die real existierende Schaltung

Zunächst müssen die 16 Adreßsignale speichergerecht in zwei Portionen aufgeteilt werden. Die "Portionierer" finden wir auf Seite 3 des Schaltplanes links oben. Über zwei Steuereingänge (Pins 2 und 14) wird jeweils eines von vier anliegenden Adreßsignalen auf die Speicher durchgeschaltet. Warum aber vier? Nun, da sind die erwähnten Zeilen- und Spaltenadressen, die Speicherauffrischungsadresse und die Adressen von einem eventuellen Coprozessor. Es ist vorgesehen, daß eine Coprozessorplatine die RAM-Adressen selbst multiplext und auch eigene RAS- und CAS-Signale absondert.

Dem Zeitverhalten der Schaltung werde ich mich in einer anderen Folge widmen, wie gesagt, aus gegebenem Anlaß, obwohl dessen Kenntnis eigentlich hier schon zu einem tieferen Verständnis notwendig wäre. Analysieren wir erst einmal die

#### Auffrischungslogik

Rechts von den besagten Multiplexern finden wir einen Zählerbaustein vom Typ 74LS393, von dem zwei Ausgänge auf ein Und-Gatter wirken, das beim Erreichen des Wertes 20 den Wert 1 annimmt. Der Zähler wird mit PHI 1 getaktet. Hierdurch wird das oben erwähnte Anhalten des Prozessors zum Zwecke der Speicherauffrischung ermöglicht. Ein Prozessor kann aber nicht in jeder Situation angehalten werden, und so wird mit der Speicherauffrischung gewartet, bis es dem Prozessor in den Kram paßt. Ein eventueller Coprozessor wertet dann normalerweise (nicht Z80) das Signal "P2REFREQ" aus und veranlaßt den Auffrischungszyklus durch "P2REFGNT".

Der 6509 kann nur bei einem Befehlholzyklus angehalten werden. Dieser wird mit dem "SYNC"-Signal angezeigt. Also wird, wenn der Zählerstand 20 erreicht ist, der Zähler angehalten,

und beim nächsten Befehlsholzyklus wird ein Flipflop 74S74 gesetzt. Das Signal "RDY" hält den Prozessor an. Von den Wirkungen der "REFEN"-Signale sind an dieser Stelle nur drei interessant, nämlich: Der Zähler wird zurückgesetzt und fängt wieder bei Null an, ein weiterer Zähler vom Typ 74LS393, der die oben erwähnten Zeilenadressen generiert, wird inkrementiert. Dieser Zähler zählt ständig von 0 bis 255. Die besagte Auffrischungsadresse wird durch die Multiplexer 74S153 an die Speicher weitergereicht. Beim nächsten Prozessorzyklus wird das Flipflop 74S74 zurückgesetzt. Fall ein Coprozessor die Kontrolle über das dynamische RAM hat, wird das asynchrone Anhalten des 6509 dadurch verhindert, daß das "BUSY2"-Signal ein setzen des Flipflops verhindert. Der 6509 turmt dann ohnehin nur auf der Bank 15 herum.

### Der dicke Fehler...

Das hört sich eigentlich doch alles ganz plausibel an, spätestens nach dem dritten lesen, sollte man meinen. Doch weh und ach, die Schaltung funktioniert nicht ganz erwartungsgemäß! Selbst ein Total-Amputierter kann an seinen Fingern abzählen, wie viele Befehlsholzyklen ein Prozessor, sagen wir, pro Jahr tätigt, während RESET aktiv ist.

und die Folgen...

Jede Betätigung des so beliebten und seitens der Software so überaus sinnvoll bedienten roten Knöpfchens, das so brav seine Dienste als Not-Starter anbietet, kann zum Datenverlust führen, da mit Sicherheit jedes mal die notwendige Auffrischungszeit der dynamischen Speicher überschritten wird. Murphy weiß schon lange, wann so etwas passiert, üblicherweise, wenn die Arbeit wenigstens eines Tages im Speicher steht.

...wollen wir nicht ertragen.

Mein Computer verträgt bestimmt auch ein Jahr Reset; ich könnte es beschwören, habe aber keine Lust, es zu beweisen. Die Lösung setzt schon eine gewisse Fertigkeit im löten voraus, dafür kostet sie fast nichts. Das einzige, was man tun muß, ist, den RESET und das SYNC so miteinander verknüpfen, daß beide gleichrangig die Speicherauffrischung freigeben. Wenn RESET dann aktiv ist, wird jeden zwanzigsten Taktzyklus eine Auffrischung durchgeführt. Das Bildchen 3 habe ich eigens zur Veranschaulichung meines Schaltungsvorschlages angefertigt. Pinbezeichnungen: 600 (700)

### Die praktische Ausführung

Den zusätzlichen 74S00 lötet man am besten auf ein benachbartes Bauteil gleicher Größe auf. Unbenutzte Ausgänge von Gattern, deren Eingänge zur mechanischen Stabilität aufgelötet werden, kneift man am besten ab.\* Die benutzten Pins werden abgespreizt und zur Vorsicht etwas verkürzt. Beim 700er hat

man das Glück, daß die Verbindung Pin 9 U 82 durch das abkratzen einer Leiterbahn auf der Platinoberseite leicht zu unterbrechen ist, und zudem wird noch der Komfort einer brauchbaren Durchkontaktierung geboten. Wenn man es schafft, ein IC aufzulöten, gibtes keine Probleme.

Beim 600er sieht es ein wenig anders aus. Die Verbindung Pin 4 U 88 liegt auf der Unterseite, und es gibt keine brauchbare Durchkontaktierung in der Nähe. Am einfachsten ist es wohl, die Leiterbahn trotzdem durchzukratzen und ein etwas längeres Drähtchen vorzusehen. Die nächste erreichbare Durchkontaktierung findet man an der Leiterbahn, die zwischen Pin 6 und 7 von U 5 herauskommt. Etwas eleganter ist es wohl, Pin 4 von U 88 durchzukneifen und ganz hochzubiegen. Dann einfach (das genügt) alles nach Plan verlöten und sich am völlig neuen Resetgefühl ergötzen. Allerdings muß man davon ausgehen, daß Systemabstürze mit wesentlich größerer Wahrscheinlichkeit selbst verursacht sind, und wer gibt das schon gerne zu!

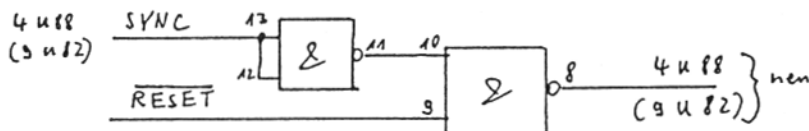


Abb. 3

\* Anm.: lieber nicht, siehe Echtzeit Seite 17/18

### Dein PAL, das unbekannte Wesen

Selbstverständlich gibt es PALs vom Typ "Black Box"; angesichts der einfachen Aufgabe, die das PAL in unseren Rechnern zu erfüllen hat, finde ich es gar nicht schwierig, dessen Funktion zu verstehen. Wenn man weiß, welche Signale die Speicher brauchen, ist dies in der Tat kein Kunststück.

Der Prozessor 6502 hat bekanntlich 16 Adressleitungen, und kann damit genau eine 64-k-Bank adressieren. Die Art und Weise, in der das geschieht, ist oben hoffentlich ausführlich genug beschrieben. Der 6509 hat zusätzlich vier Adreßerweiterungsleitungen, die ihm den Zugriff auf bis zu 16 Bänken je 64 kByte erlauben. Diese Adressen können den standardmäßigen Speicherbausteinen nicht direkt zugeführt werden. Die 16 "normalen" Adreßleitungen werden allen Speichern wie erklärt zur Verfügung gestellt. Welche Bank gerade gemeint ist, wird ausschließlich über die Signale RAS und CAS mitgeteilt. Diese also müssen entsprechend den anliegenden Bankadressen (bzw. A16 bis A19) dekodiert werden.

Wie schon erwähnt, ist vorgesehen, daß ein Coprozessor eigene RAS- und CAS-Signale erzeugen muß. Diese heißen im Schaltplan ERAS und ECAS. Ob CAS und RAS oder ECAS und ERAS an die Speicher weitergeleitet werden muß, wird durch das Signal BUSY2,



festgelegt. Es ist aktiv, wenn der Coprozessor die Kontrolle über den Speicher hat.

Eine Besonderheit stellen die Auffrischungszyklen dar. Bei der Auffrischung werden alle Speicher über RAS adressiert, und da die Adressierung unvollständig ist, kommt es zu keinem Kurzschluß auf den Datenleitungen.

Es dürfte genügen, die logischen Gleichungen der PAL-Programmierung anzugeben. Nicht jedem Leser wird die Schaltalgebra geläufig sein, und so muß ich wohl noch ein paar Takte dazu schreiben.

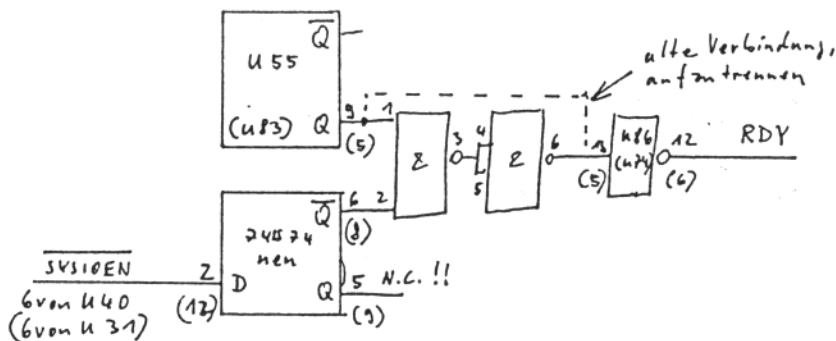
Es wird mit Punkt- und Strichrechnung gearbeitet wie in der normalen Mathematik. Punktrechnung geht vor Strichrechnung. Ausdrücke in Klammern haben Vorrang. Nur haben die Symbole eine andere Bedeutung : + heißt oder, \* heisst und, / vor einem Symbol heißt unwahr bzw 0. Mögen diese wenigen Hinweise auch unbedarfteren Mitbenutzern helfen, sich ein tieferes Verständnis für die Schaltung zu erarbeiten.

Die PAL-Gleichungen

```
/RASSEG1 = (BP0 * /BP1 * /BP2 * /BP3 + REFEN) * (/RAS * BUSY2  
+ /ERAS * /BUSY2)  
/RASSEG2 = (/BP0 * BP1 * /BP2 * /BP3 + REFEN) * (/RAS * BUSY2  
+ /ERAS * /BUSY2)  
/RASSEG3 = (BP0 * BP1 * /BP2 * /BP3 + REFEN) * (/RAS * BUSY2  
+ /ERAS * /BUSY2)  
/RASSEG4 = (/BP0 * /BP1 * BP2 * /BP3 + REFEN) * (/RAS * BUSY2  
+ /ERAS * /BUSY2)  
/CASSEG1 = (BP0 * /BP1 * /BP2 * /BP3 * /REFEN) * (/CAS *  
BUSY2 + /ECAS * /BUSY2)  
/CASSEG2 = (/BP0 * BP1 * /BP2 * /BP3 * /REFEN) * (/CAS *  
BUSY2 + /ECAS * /BUSY2)  
/CASSEG3 = (BP0 * BP1 * /BP2 * /BP3 * /REFEN) * (/CAS * BUSY2  
+ /ECAS * /BUSY2)  
/CASSEG4 = (/BP0 * /BP1 * BP2 * /BP3 * /REFEN) * (/CAS *  
BUSY2 + /ECAS * /BUSY2)
```

#### Echtzeit

Wenn der Computer auf der Bank 15 arbeitet, und das tut er meistens, ist es, wie aus dem oben gesagten hervorgeht, nicht nötig, den Prozessor für eine Speicherauffrischung anzuhalten. Da das Auftreten von Speicherauffrischungen das Zeitverhalten von Programmen beeinflußt in einer Weise, die nicht vollkommen kalkulierbar ist, kann es sinnvoll sein, dafür zu sorgen, daß man hier das Anhalten des Prozessors unterbindet. Die knapp 5%, die der Computer dadurch schneller wird, fallen wohl meistens nicht ins Gewicht. Benötigt wird ein zusätzliches Flipflop 74LS74, ein noch freies Nicht-Oder-Gatter und die noch freien Gatter des für die obige Schaltungskorrektur notwendigen 74S00. Die im folgenden Schaltbild nicht angegebenen Anschlüsse des Flipflops werden mit dem darunterliegenden Flipflop (U 86 beim 600er bzw. U 83 beim 700er) verlötet.



Pinbezeichnungen: CBM 600 (700)

### Software auf Kassette

Bittere Klagen sind uns zu Ohren gekommen, daß die immer noch zahlreichen Datasettenbenutzer unter uns nicht in ausreichendem Maße Unterstützung finden. Klar, was sollen die denn beispielsweise mit unseren Clubdisketten, denen hier so viel Raum gewidmet ist, anfangen? Normal kann man davon ausgehen, daß die Datasette nach der Anschaffung einer Diskettenstation gleich nach dem Überspielen der Programme für immer ausgedient haben wird. Uns fällt dazu nichts ein, als um Hilfe für die Datasettenbenutzer zu bitten. Erst einmal wäre es wichtig, zu erfahren, wer noch ausschließlich mit Datasetten arbeitet. So könnte man die Datasettenbenutzer zusammenbringen. Kopieren von Diskette auf Datasette ist natürlich wesentlich komfortabler als von Datasette auf Datasette, insbesondere, da wir nur einen Kassettenport an unseren Geräten haben. Wir sehen uns nicht imstande, diese Arbeit zu verrichten. Melde sich wer kann.

Hallo DATASETTE-Freunde,  
da sich die Freude am Computern u.a. proportional zur Lesbarkeit der geschriebenen Programme verhält, stelle ich hier einen Schaltungskniff zur einfachen Spurkontrolle der DATASETTE vor. Die Methode ist denkbar sicher, einfach und preiswert. Man baue die aufgeführten Bauteile nach untenstehender Anleitung ein und gleiche den Tonkopf der Datasette in "Play"-Betrieb so ab, daß der wiedergegebene Ton möglichst "hell" klingt. Diese Methode ist sicherer als die in der HSG Nr.1 beschriebene, und Ungleichmäßigkeiten der Datasette oder des Bandes sind sofort hörbar.

Benötigtes Material:

- 1) 1 Miniatur Piezo Lautsprecher (zB. Völkner:BCE 20 N 14 zu 0.95 DM, Abmessungen d=24,h=8mm).
- 2) Ein Widerstand 1.5 kOhm/0.3 Watt.
- 3) ca. 30 cm dünner flexibler Draht.
- 4) Ein Stück doppelseitiges Klebeband.

Einbau:

Die 4 Schrauben im Dassettenboden herausschrauben und den Lautsprecher mit dem Klebeband an eine freie Stelle der Innenseite des oberen Gehäusedeckels kleben. An den einen Anschluß des Lautsprechers den Widerstand anlöten, an den anderen und an das freie Ende des Widerstandes je einen Draht löten der gut bis zur Platine reicht. Die beiden Drähte werden an Pin 11 und 1 des LA 6324 angelötet, oder, falls die Datasette mit 1 oder 2 LM 358 bestückt ist, ein Draht an Pin 4 eines der IC's und den anderen Draht an Pin 1 oder 7 des gleichen oder andern IC's angelötet, je nach dem an welchem Punkt das zweitlauteste Signal (bei Play) zu hören ist.

Feststellung der IC Pin's:

Dazu muß die Platine der Datasette demontiert werden. Zuerst eine Kassette einlegen und abwechselnd RECORD und STOP drücken. An der Stelle an der sich dabei zwischen Platine und Chassis etwas bewegt sitzt der Aufnahme-Wiedergabe-Schalter. Die Blechfahne, die diesen betätigt, muß nach der Montage der Platine wieder in die Öse des Schalters ragen!!! Die Platine also jetzt demontieren (2 Schrauben), darunter schauen und nach einer Markierung (Punkt, Kerbe, o.ä.) auf den relevanten IC's suchen. Von dort aus die Pins mit 1 beginnend gegen den Uhrzeigersinn durchzählen. Platine wieder montieren.

Abgleichkassette:

... ist bei Bedarf bei mir gegen einen Unkostenbeitrag von 5 DM zu beziehen.

## Super-Tape Korrekturen

Andreas Stankewitz

Leider sind mir bei der Programmierung von Super-Tape einige Fehler unterlaufen. Da viele das Programm haben, aber nur diejenigen, die es bei mir gekauft haben, eine Korrektur zugeschickt bekommen haben, sollen die Fehler und ihre Berichtigungen hier kurz beschrieben werden.

In \$bfef muß der Sprungbefehl JMP \$bf66 (4c 66 bf) in JMP \$bf6b (4c 6b bf) geändert werden.

Damit Super-Tape auch auf dem CBM 700 lauffähig ist, muß die Abfrage der Stoptaste geändert werden:

```
be30 20 f2 bf   jsr $bff2      statt jsr $f979
be33 f0 f4      beq $be29      statt bmi $be29

bf46 20 f2 bf   jsr $bff2      statt jsr $f979
bf49 f0 1e      beq $bf69      statt bmi $bf69

bff2 20 79 f9   jsr $f979
bff5 29 01      and #$01
bff7 60         rts
```

Der Code in der Save-Routine, anhand dessen deren Vorhandensein überprüft wird, muß natürlich 43 42 4d 20 53 2d 54 lauten.

Ein weiterer Fehler ist mir bei der Initialisierung für das Schreiben mit 7200 baud unterlaufen. Folgende Speicherstellen des Super-Save in der Bank 15 müssen geändert werden:

```
:050e a0 04
:0541 a2 04 b9
:0548 88 ca
:0566 a0 08
```

So wird das neue Super-Save auf Band geschrieben:

```
:00a0 00      (Sekundäradresse gleich null setzen)
s"super-save",01,0f0400,0f0590
```

## Bildschirmschoner

Andreas Stankewitz

Das folgende Programm ist zur Schonung von Bildröhren im Dauerbetrieb gedacht. Es schaltet den Bildschirm dunkel, wenn eine bestimmte Zeit lang keine Taste der Tastatur gedrückt wurde. Das Betätigen irgendeiner Taste bewirkt das sofortige wiederaufleuchten des Schirmes mit dem alten Bildschirminhalt.

Das Programm ist relativ simpel, aber nichts desto trotz sehr wirkungsvoll. Ich will hiermit ein Beispiel für Programme geben, die am besten im Kernal-EPROM aufgehoben sind. Damit es aber auch von den "Nichtbrennern" unter uns benutzt werden kann, habe ich die Routinen zur Einbindung in die Interrupt-Routine mit angegeben.

Der Videocontroller ist so freundlich, uns ein fast perfektes Hilfsmittel an die Hand zu geben, um den Bildschirm dunkel zu schalten. Es muß nur der Inhalt eines Registers auf Null gesetzt werden. Dieses Register bestimmt, wieviele Zeilen dargestellt werden sollen. Zum Bildschirm einschalten muß dieses Register mit 25 (hex: \$19) beschrieben werden.

Nur fast perfekt ist diese Methode, da ich festgestellt habe, daß die erste Pixelzeile auf dem Bildschirm nicht ausgeschaltet wird. Das macht sich bemerkbar, wenn in der ersten Zeile inverse Zeichen stehen. Wenn das sehr stört, der kann sich in das Programm eine Routine einbinden, die zum ausschalten die erste Zeile irgendwo in den Speicher kopiert, dann mit Leerzeichen vollschreibt und beim einschalten wieder zurückschreibt.

Zur Funktionsweise des Programms:

Das eigentliche Programm beginnt ab \$07c7. Das vorherstehende dient nur der Einbindung in die Interruptroutine des Betriebssystems. Für Neulinge der Interruptprogrammierung ist diese Einbindung nach dem Programmlisting beschrieben.

Zuerst wird anhand von \$dl überprüft, ob eine Taste gedrückt ist. Wenn das der Fall ist, wird die Dunkelflag überprüft. Ist sie gesetzt (größer \$7f) dann wird der Bildschirm wieder eingeschaltet.

Wenn keine Taste gedrückt wurde, dann wird die Dunkelflag ebenfalls überprüft. Ist sie gesetzt, so heißt das, daß der Bildschirm schon dunkel ist, und es muß nur noch auf das Drücken einer Taste gewartet werden.

Wenn keine Taste gedrückt wurde und der Bildschirm noch nicht dunkel ist, dann wird der Zähler inkrementiert und anschließend mit dem Endwert verglichen. Ist der Endwert erreicht, so wird der Bildschirm dunkel geschaltet und die Dunkelflag gesetzt.

Hier das Programmlisting:

|                 |            |                   |
|-----------------|------------|-------------------|
| .,079e 78       | sei        | Interruptvektor   |
| .,079f a9 ab    | lda #\$ab  | auf die Routine   |
| .,07a1 8d 00 03 | sta \$0300 | verbiegen.        |
| .,07a4 a9 07    | lda #\$07  |                   |
| .,07a6 8d 01 03 | sta \$0301 |                   |
| .,07a9 58       | cli        |                   |
| .,07aa 60       | rts        |                   |
|                 |            |                   |
| .,07ab a5 01    | lda \$01   | Neue IRQ-Routine  |
| .,07ad 48       | pha        |                   |
| .,07ae d8       | cld        |                   |
| .,07af ad 07 de | lda \$de07 | Ist ein IRQ über  |
| .,07b2 d0 03    | bne \$07b7 | den IRQ-Sammler   |
| .,07b4 4c a2 fc | jmp \$fca2 | eingegangen ?     |
| .,07b7 c9 01    | cmp #\$01  | 100-Hz IRQ ?      |
| .,07b9 f0 03    | beq \$07be | nein, dann zum    |
| .,07bb 4c f5 fb | jmp \$fbf5 | Kernal zurück     |
| .,07be 20 13 e0 | jsr \$e013 | Tast. abfragen    |
| .,07c1 20 c7 07 | jsr \$07c7 | Bildschoner-Rout. |
| .,07c4 4c 87 fc | jmp \$fc87 | zum Kernal zurück |
|                 |            |                   |
| .,07c7 a5 d1    | lda \$d1   | Taste gedrückt ?  |
| .,07c9 f0 14    | beq \$07df | nein, inc Zähler  |
| .,07cb a5 e8    | lda \$e8   | Bild. dunkel ?    |
| .,07cd 10 0a    | bpl \$07d9 | nein, Zähler = 0  |
| .,07cf a9 06    | lda #\$06  | Bild. einschalten |
| .,07d1 8d 00 d8 | sta \$d800 |                   |
| .,07d4 a9 19    | lda #\$19  |                   |
| .,07d6 8d 01 d8 | sta \$d801 |                   |
| .,07d9 a9 00    | lda #\$00  | Zähler = 0 setzen |
| .,07db 85 e7    | sta \$e7   |                   |
| .,07dd 85 e8    | sta \$e8   |                   |
| .,07df a5 e8    | lda \$e8   | Bild schon dunkel |
| .,07e1 30 1a    | bmi \$07fd | ja, zum Kernal    |
| .,07e3 e6 e7    | inc \$e7   | Zähler erhöhen    |
| .,07e5 d0 02    | bne \$07e9 |                   |
| .,07e7 e6 e8    | inc \$e8   |                   |
| .,07e9 a5 e8    | lda \$e8   | Zähler = ende ?   |
| .,07eb c9 10    | cmp #\$10  | Endwert *         |
| .,07ed 90 0e    | bcc \$07fd | nein, zum Kernal  |
| .,07ef a9 06    | lda #\$06  | Bild. ausschalten |
| .,07f1 8d 00 d8 | sta \$d800 |                   |
| .,07f4 a9 00    | lda #\$00  |                   |
| .,07f6 8d 01 d8 | sta \$d801 |                   |
| .,07f9 a9 80    | lda #\$80  | Bilddunkel-Flag   |
| .,07fb 85 e8    | sta \$e8   | setzen            |
| .,07fd 4c 79 f9 | jmp \$f979 | Zum Kernal        |

\* Die Zeit, die vergeht, bis der Bildschirm dunkel geschaltet wird, ist (Endwert \* 2,56) Sekunden.

Das Einbinden einer Interruotroutine:

Mit der kurzen Routine ab \$079e wird der Interruptvektor auf unsere eigene Routine ab \$07ab verbogen. Tritt jetzt ein

Interrupt auf, so wird, nach dem die Betriebssystemroutine alle Register gerettet hat, zu der Adresse gesprungen, die in \$0300 und \$0301 steht.

Die neue Routine muß in unserem Fall alles das erledigen, was bisher auch von der Interruptroutine im Betriebssystem gemacht wurde. Das heißt ersteinmal, das Bankregister retten und den BCD-Modus ausschalten.

Dann wird überprüft, ob der Interruptsammler (ein TPI 6525) einen Interrupt festgestellt hat. Wenn das nicht der Fall ist, so wird zurück zum Betriebssystem verzweigt, das die Register zurückschreibt und zum Hauptprogramm zurückkehrt.

Hat der Interruptsammler den Interrupt ausgelöst, so wird überprüft, ob der 100 Hz Takt ausgelöst hat, oder ein anderer Baustein. Bei letzterem wird auch zum Betriebssystem zurückgekehrt, das dann die weitere Bearbeitung übernimmt.

Jetzt wissen wir also, der Interrupt wurde vom 100 Hz Takt ausgelöst, daß heißt, die Tastatur (auch die Kassettenrekordertasten) sollen abgefragt werden.

Mit jsr \$e013 wird die Tastatur abgefragt. Das Ergebnis dieser Abfrage wird dann in der Bildschirmschoner-Routine ausgewertet. Dann wird zum Betriebssystem zurückverzweigt, das die übrigen Abfragen übernimmt (Kassettenrekordertasten).

Die Einbindung direkt ins Kernal

Wenn der Bildschirmschoner direkt ins Kernal eingebunden wird, dann wird nur die Routine ab \$07c7 gebraucht. Diese ist voll verschiebbar und kann überall im Kernal-EPROM liegen. In der Interruptroutine im Kernal muß dann nur der jsr-Aufruf der Stoptastenabfrage auf unsere Routine geändert werden. Also statt:

```
fc84 20 79 f7 jsr $f797
```

muß

```
fc84 20 xx yy jsr $yyxx
```

eingetragen werden.

Wenn die Routine im Kernal-EPROM ab \$ed0c liegt, lautet die neue Zeile:

```
fc84 20 0c ed jsr $ed0c
```

Hier ist massig Platz für das Progrämmchen.

Der Videocontroller wird im 600/700 unterschiedlich initialisiert, abhängig davon, ob der 600er oder 700er Modus benutzt wird, ob PAL oder NTSC. Sogar zwischen DIN und ASCII findet sich noch ein Unterschied, der aber in den neuen Club-EPROMs ausgeräumt ist, da er bei manchen Monitoren zu Anpassungsproblemen geführt hat (siehe unten).

Da auf der Clubdiskette "Utilities" ein Programm zu finden ist, mit dem man mit der Bildschirminitialisierung rumspielen kann, will ich hier kurz die Register beschreiben und in der Tabelle die verschiedenen Initialisierungswerte darstellen.

Die Register des CRT-Controllers:

- R0 Gesamtzahl der Zeichen pro Zeile, hiervon hängt die horizontale Synchronisation ab.
- R1 Anzahl der dargestellten Zeichen pro Zeile.
- R2 Horizontale Synchronisationsposition, hiermit wird die links-rechts-Verschiebung der Zeilen eingestellt.
- R3 Länge des Synchronimpulses, highbyte = vsync, lowbyte = hsync.
- R4 Gesamtzahl der Zeilen, mit R5 zusammen wird hier die Bildfrequenz (vertikale Synchronisation) bestimmt.
- R5 Dient zur Feinabstimmung von vsync.
- R6 Anzahl der dargestellten Zeilen.
- R7 Vertikale Synchronisationsposition, hiermit wird der dargestellte Text in der Vertikalen verschoben.
- R8 Kontrollregister, hier wird die Arbeitsweise des CRTC festgelegt, ist zum Spielen ungeeignet.
- R9 Anzahl der Pixelzeilen pro Zeichen.
- R10+R11 Cursorgröße, R10 ist erste Pixelzeile, R11 ist letzte Pixelzeile, die den Cursorblock bestimmt. Bit 5 und 6 von R10 haben folgende zusätzliche Bedeutung:
  - 00 = kein Blinken,
  - 01 = kein Cursor,
  - 10 = Blinken mit 1/16 Bildfrequenz,
  - 11 = Blinken mit 1/32 Bildfrequenz.
- R12+R03 Video-RAM-Startadresse, ist beim 600/700er immer 0, da nicht mehr RAM als eine Bildschirmseite zur Verfügung steht.



R14+R15 Cursor-Adresse, RAM-Adresse, bei der der Cursor erscheinen soll. Dieses Registerpaar kann auch gelesen werden.

R16+R17 Lightpen-Adresse, wenn am Lightpen-Eingang ein Impuls wahrgenommen wurde, so findet sich in diesem Registerpaar die RAM-Adresse, die mit der entsprechenden Bildschirmstelle der Impulszeit übereinstimmt.

Im 600/700er werden die Register beim Reset wie folgt programmiert (alle Werte in hex):

| Reg. | 600 |      | 700 |    |
|------|-----|------|-----|----|
|      | PAL | NTSC |     |    |
| 00   | 7e  | 7f   | 6b  |    |
| 01   | 50  | 50   | 50  |    |
| 02   | 62  | 60   | 53  | *1 |
| 03   | 0a  | 0a   | 0f  |    |
| 04   | 1f  | 26   | 19  |    |
| 05   | 06  | 01   | 03  |    |
| 06   | 19  | 19   | 19  |    |
| 07   | 1c  | 1e   | 19  |    |
| 08   | 00  | 00   | 00  |    |
| 09   | 07  | 07   | 0d  |    |
| 0a   | 00  | 00   | 60  |    |
| 0b   | 07  | 07   | 0d  |    |
| 0c   | 00  | 00   | 00  |    |
| 0d   | 00  | 00   | 00  |    |
| 0e   | 00  | 00   | 00  |    |
| 0f   | 00  | 00   | 00  |    |
| 10   | 00  | 00   | 00  |    |
| 11   | 00  | 00   | 00  |    |

\*1 Seltsamer Weise ist bei 700-DIN der Text um drei Zeichen weiter rechts als bei 700-ASCII. Seltsam besonders deswegen, da jetzt die Zeichen in das hsync-Signal reinragen. Die meisten Monitore sind Pegelsynchronisiert und geben dann die letzten zwei Zeichen einer jeden Zeile im Rücklaufstrahl aus, was nicht gerade sehr schön aussieht.

600er und 700er DIN unterscheiden sich noch in der Adresse \$ea0a. Hier steht der Werte für die erste Linie des Untestreich-Cursors. Beim 600er ist sie 06, beim 700 ist sie \$0b.

Übersetzt von Rüdiger Gedenk aus der Zeitschrift "THE CBUG  
ESCAPE" Vol. 5, p. 15-16

Anmerkung: Entstanden Sommer 1986, Teil 2 Zu einem Teil ist  
dieses Programm ein Auszug aus einem Etiketten-Druckprogramm  
des Autors.

#### PROGRAMMBESCHREIBUNG

Ich nehme an, daß mittlerweile die meisten B-Maschinenbesitzer sich mit der Darstellung eines Bildschirmfensters mittels der ESCape "T" und "B" Anweisungen auskennen. "T" erzeugt die linke obere Ecke des Fensters und "B" rechte untere Ecke. Die größte Unannehmlichkeit im Gebrauch der Anweisung ist, daß der cursor sich in der Ecke befinden muß, in der die Ecke des Ausschnitts gewünscht wird, wenn der ESCape Befehl benutzt wird.

Falls das Fenster von innerhalb des Programms gewünscht wird, ist es nicht immer bequem den cursor in den vorgesehenen Eckbereich zu bringen und die ESCape Anweisungen vor dem Schreiben in das Fenster zu benutzen. Ich finde es leichter die Positionen mittels POKE von Zeile (line) und Spalte (column) im Programm zu erzeugen. Die POKES befinden sich in Bank 15 und lauten wie folgt:

```
poke220, (obere Arbeits-Zeile #)  
poke221, (untere Arbeits-Zeile #)  
poke222, (linke Arbeits-Spalte #)  
poke223, (rechte Arbeits-Spalte #)
```

Beachten Sie, daß ich diese Anweisungen als Arbeits-Positionen (working positions) gekennzeichnet habe. Die Zahl welche gePOKEt wird bestimmt Zeilen (line)- resp. Spalten (column)-Grenzen, in welchen der cursor oder die PRINT Anweisung sich bewegen kann.

#### BENUTZERHINWEISE

1.  
Beachten Sie, wenn ein Fenster definiert werden soll, daß die Maschine sowohl die oberste Zeile als auch die linksbündige Spalte als "0" und nicht als "1" erkennt.

2.  
Wenn Sie also die obere Zeile des Fensters als Zeile 5 und die untere Zeile des Fensters als Zeile 10 definieren wollen, so ist ein Fenster mit 6 Zeilen, nicht 5, zu erzeugen, entsprechend der eingeschlossenen Gesamtzahl (frozen lines). Gleiches gilt für die Erzeugung der Spalten.

3.  
Nachdem das Fenster definiert ist kann mit "print\$(19)" der cursor in die linke obere Ecke des Fensters (home) gesetzt werden. Ein "print\$(147)" Befehl löscht den Inhalt des Fensters und setzt den cursor nach HOME.

4.  
Nachdem das Fenster definiert und der cursor plazierte ist, kann letzterer dieses Fenster nicht verlassen. Das Verlassen des Fensters (break freeze) kann nur mittels eines direkten POKE-Befehls erfolgen: "printchr\$(19)chr\$(19).

5.  
Der Weg, HOME oder CLEAR Anweisungen in ein Programm zu übernehmen, wird durch PRINT, Anführungszeichen, Kontrollbuchstaben und wieder Anführungszeichen, entsprechend nachfolgenden Beispielen erreicht:

```
print"s"      - cursor nach HOME
print"S"      - löscht den Bildschirm; cursor HOME
print"ss"     - hebt die Fensterbegrenzung auf; cursor HOME
print"ssS"    - hebt die Fensterbegrenzung auf, löscht den Bildschirm; cursor HOME
```

6.  
Falls es erwünscht ist mit blinkendem cursor in einem Bildschirmfenster zu arbeiten, obgleich der Rechner auf eine GET-Anweisung wartet, kann daneben ein Fenster definiert werden, danach HOME & CLEAR geben und danach poke55296,10: poke55297,peek(212) eintasten. Dann wird die Get-Anweisung geöffnet. Der cursor bleibt eingeschaltet bis zur nächsten INPUT-Anweisung, oder bis die Befehlsfolge: poke55296,10: poke55297,24 den cursor abschaltet.

#### PROGRAMMBEISPIEL

Geben Sie das nachfolgende Programm ein und üben Sie sich mit der Positionierung des Bildschirmfensters, um ein Gefühl für die Arbeitsweise zu bekommen. Ich bin sicher, daß dieses Programm einige Anregungen zu Anwendungen in eigenen Entwicklungen gibt. Das nachfolgende Programmbeispiel kann durch Drücken der ESCape Taste beendet werden:

```
10 printchr$(147)
20 print"(6 cursor downs)"
30 printtab(21)"36 under rule chr here CTRL 4"
   (36 mal CTRL 4)
40 for n=1 to 8: printtab(20)" "tab(57)" ":next
   (first graphic is CTL- on numeric keypad, 2nd
   graphic is CTL7 on keypad)
50 printtab(21)"36 over rule chr here CTRL 3"
   (36 mal CTL3)
60 poke220,9:221,16:poke222,21:poke223,56
70 printchr$(19)
80 rem printchr$(27)"m"
90 poke55296,10:poke55297,peek(212)
100 getz$
```

```

110 ifz$=chr$(27)then140
120 rem poke210,0:poke211,0
130 printz$;:gotol00
140 end

```

Wenn es erwünscht ist , daß die Anführungszeichen innerhalb des Fensters unterdrückt werden, ist die REM-Anweisung aus der Zeile 120 zu löschen. Diese Änderung hält die Maschine davon ab in den Anführungsmodus zu gehen wenn der INsert-Befehl benutzt werden muß oder das "-"Symbol ausgedruckt werden soll.

Falls das SCROLLING während des Arbeitens im Bildschirmfenster verhindert werden soll, ist das REM in der Zeile 80 zu löschen. In den Grenzen eines schmalen Fensters ist es leicht den Text aus den Begrenzungen des Fensters zu bringen.

### Bildschirmflimmern

Gerd Schumacher

Als ich noch einen so miserablen Monitor hatte, daß es überhaupt nicht auffiel, hat mir schon mal jemand einen entsprechenden Hinweis gegeben, aber ich hatte noch kein Problembewußtsein. Gelegentlich flitzten äußerst unangenehme Streifen über den Bildschirm. Dadurch ist die inverse Darstellungsweise praktisch nicht zu gebrauchen.

Dirk Seifert hat festgestellt, daß dieses Phänomen immer beim Zugriff auf den Videocontroller auftritt. Das Signal 'notcrtcs' verursacht diesen Unfug. Es ist an ein Flipflop angeschlossen, dessen Ausgänge laut Schaltplan unbeschaltet sind. In Wirklichkeit ist aber der Pin 9 mit Pin 13 von U 56 (beim 700er U 96) verbunden. Dieses NAND dient eigentlich nur zur Invertierung von display-enable vom Videocontroller, ist aber aus nicht erfindlichen Gründen mit dem besagten FlipFlop verbunden.

Abhilfe ist leicht zu schaffen, indem man die Verbindung des Pin 13 von U 19 im 600er bzw. von U 54 im 700er unterbricht und den Pin mit dem benachbarten Pin 12 verbindet. Ein neues, königliches Glotzgefühl!

Eine Idee hätte ich ja für die Verwendung des nutzlosen Flipflops. Mit ein wenig löten ließe sich eine 40-Zeichen-Darstellung erreichen, aber wer will die schon!

## neue Projekte

### Sprachen

Andreas ist ein Quelltext für Lisp, die am weitesten verbreitete Sprache für künstliche Intelligenz zur Verfügung gestellt worden. Es ist noch nicht klar, wie hoch hier die Urheberrechtsabgaben sein werden. Jedenfalls ist das Ding sauber programmiert, und eine vollständige Anpassung, die vermutlich zwei Bänke benutzen wird, könnte noch in diesem Jahr fertig werden.

Professor Harald Lindner hat in Aussicht gestellt, einen Pascal-Compiler zu schreiben, sofern seine Zeit es erlaubt. Drücken wir ihm die Daumen.

In den USA hat jemand eine C-Compiler in Arbeit. Das wäre natürlich die obstsaltete Krönung, insbesondere, da C-Programme im allgemeinen am transportabelsten sind, und da es sich hierbei wohl um die effizienteste Hochsprache überhaupt handelt.

### Diskettenstationen

Es gibt für die 1541, am C-64 betrieben, eine unüberschaubare Menge von Floppybeschleunigungen. Für die großen Floppies 8050, 8250 und SFD gibt es lediglich Speedos, das preislich für die meisten von uns völlig indiskutabel ist. Von ungefähr kommt es nicht, daß sich offenbar sonst noch niemand darangemacht hat, diese Diskettestationen zu verbessern. Das riecht nämlich penetrant nach ernsthafter Arbeit, von der man annehmen kann, daß sie so gut wie keine Früchte tragen wird. Mit der 1541 haben sich wegen des riesigen Marktes einige Entwickler eine goldene Nase verdienen können.

Nun läuft trotzdem die Weiterentwicklung der DOS 2.7-Geräte auf Hochtouren. Ein großer Teil der Ergebnisse wird hier veröffentlicht werden, ein anderer Teil (EPROMS und so) wird zu vereinsüblichen public-domain-Preisen erhältlich sein, und die Gesamtentwicklung wird vermutlich etwa DM 150,- kosten.

Einige der angepeilten Verbesserungen sind u.a.:

Verkürzung der Zugriffszeit um etwa eine Sekunde bei 8250 und 8050, etwas weniger bei 8250 LP und SFD

Vierfache Lesegeschwindigkeit

Zehnfache Schreibgeschwindigkeit

Drei-Minuten-Backup

Subdirectoryfähigkeit

Verhinderung der Dejustierung der 8250 LP und SFD bei mechanischem Anschlag, indem die vorgesehene Kennung der Spur Null durch die vorhandene Lichtschranke einen mechanischen Anschlag verhindert.

Ansprechen der SFD als logisches Doppellaufwerk 8050

Im nächsten Heft werde ich mich ausführlich mit den Disket-

tenstationen beschäftigen. Hier aber schon mal ein interessanter Tip:

Die Aufzeichnung sequentieller Dateien erfolgt bei Geräten mit DOS 2.7 mit einem Blockversatz von 5, da das DOS die Daten nicht so schnell an den Computer übertragen kann, wie sie von der Diskette gelesen- oder darauf geschrieben werden können. Für jeden Kanal stehen zur Kommunikation zwischen dem Bus- und dem Diskcontroller 2 RAM-Bereiche von jeweils 256 Byte zur Verfügung. Diese beiden Puffer werden von den Prozessoren abwechselnd benutzt. Während der eine beispielsweise in Puffer 6 schreibt, liest der andere aus Puffer 7, und dann werden die Puffer umgekehrt benutzt.

Findet nun der Disk-Controller, wenn er den nächsten Block übertragen müßte, keinen Puffer vor, in dem er seine Daten unterbringen kann, muß die Diskette eine zeitraubende Ehrenrunde drehen. Ich habe herausgefunden, daß der standardmäßige Blockabstand zu knapp bemessen ist, als daß der Buscontroller mit seiner Übertragung schritthalten könnte. Es werden viele Ehrenrunden gedreht.

Mit folgender kleinen Routine kann der Blockversatz auf optimale 12 umprogrammiert werden, was bei Schreiboperationen einen Geschwindigkeitsvorteil von 30% und beim lesen von sage und schreibe 60% !!! ausmacht. Des Geschwindigkeitsvorteils wird man teilhaftig bei allen prg, usr und seq - Dateien, die auf diese Art aufgezeichnet worden sind. Das "Programm":

```
open 15,gr,15 :
print#15,"m-w"chr$(244)chr$(16)chr$(1)chr$(12) :
close 15
```

"m-w" bedeutet schreiben in den Speicher der Diskettenstation.

chr\$(244)chr\$(16) bezeichnet die Adresse \$10F4 im Speicher

chr\$(1) ist die Anzahl der zu übertragenden Bytes

chr\$(12) ist der neue Blockversatz

Der Blockversatz in der 4040 ist auch per RAM-Zugriff veränderbar. Vermutlich beträgt hier der optimale Blockversatz 8, was eine 20%ige Geschwindigkeitssteigerung bei sequentiellen Leseoperationen bedeuten würde. Wer es ausprobiert, möge mir Bescheid geben. Hier müssen die chr\$ folgender Werte übertragen werden: 68, 67, 1, x, wobei x auszuprobieren ist.

Die 8280 hat schon ihren optimalen Blockversatz von 10, während sich bei der 2031 bis zu 16% herausholen lassen mit einem Blockversatz von 17. Hierfür werden die Werte 105, 0, 1, 17 an die Floppy übermittelt.

Basic-Compiler

Für unsere Computer gibt es meines Wissens drei Basic-Compiler:

1. Petspeed, als sehr effizient und pingelig mit der Syntax

bekannt von der restlichen PET-Familie her. Welche Eigenschaften er auf unseren Geräten hat, ist mir nicht bekannt. Er ist für US \$ 100,- in Amerika erhältlich.

2. DTL-Compiler, auch bekannt von den Verwandten unserer Rechner her. Er ist dem Petspeed ähnlich, allerdings meines Wissens mit einem Qualitätsabfall. Auch er erzeugt einen sehr knappen Code für eine von beiden Basic-Versionen. Er kennt nicht die Befehle, die der 600/700 mehr hat als der 8032. Sein gegenwärtiger Normalpreis beträgt US \$ 80,-. Hier sind noch ein paar Restexemplare zu DM 100,- verfügbar.

3. Austrospeed Er soll der beste Basic-Compiler für unsere Geräte sein, nur ist leider auch sein Preis so weit jenseits von Gut und Böse, daß seine Anschaffung für unsereinen kaum in Betracht kommt. Außerdem ist es der einzige, der lediglich für die 256k-Basic-Version erhältlich ist. Mein Bemühen, ihn zu einem zeitgemäßen Preis in unsere Bibliothek aufnehmen zu können, war leider nicht von Erfolg gekrönt.

Die normalerweise hohen Preise für solche Compiler lassen sich durch den enormen Arbeitsaufwand und die relativ kleinen Mengen, die man von einem solchen Programm absetzen kann, durchaus erklären.

Vor dem Hintergrund, daß eh allenfalls ein geringes Anerkennungshonorar zu erwarten ist, um den erhöhten Kaffee- und Zigarettenverbrauch zu finanzieren, ist die Neuentwicklung eines Basic-Compilers in Arbeit. Minimalziel der Entwicklung ist: Lauffähigkeit mit beiden Basic- und Kernalversionen, weitgehende Verträglichkeit mit Macro-Basic und auch eventuellen weiteren Basicerweiterungen, volle Unterstützung des gesamten Befehlssatzes und unkomplizierte Handhabung. Sollte alles gutgehen, wird der Preis deutlich unter DM 100,- liegen, da es sich um ein echtes Liebhaberprojekt handelt mit dem Ziel, alles bisher Dagewesene in den Schatten zu stellen. Ein zeitlicher Rahmen kann naturgemäß bei einem solch komplexen Vorhaben nicht gesteckt werden.

#### Kernalverbesserung und -berichtigung

Bisher hat es erstaunlich wenig Resonanz gegeben auf das Ersuchen, Vorschläge zur Verbesserung des Kernals zu machen, natürlich am liebsten gleich mit konkreten Vorschlägen. Die RS 232-Verbesserung war schon ein ein echter Knüller, jedenfalls für diejenigen, bei denen Probleme mit dieser Schnittstelle aufgetreten sind. Nun habe ich in Heft 2 den Vorschlag gemacht, eine entsprechende Softwareänderung auch für den einwandfrei arbeitenden 65C51 vorzunehmen. Dieser Vorschlag ist nachfrageseitig auf offene Ohren gestoßen, aber nur dort. Wer hat so etwas zu bieten?

Es gibt viele Kleinigkeiten zu verbessern, und es ist angebracht, daß sich einige Kernalspezialisten zusammentun, um einen neuen, voll kompatibelen Selbsthilfe-Standard kreieren. Vermutlich werden nur wenige von uns je über eine Coprozessorkarte verfügen, und man könnte die Koprozessorroutinen

weitgehend aus dem Kernal entfernen. Sie taugen sowieso nicht besonders viel. Diese bei Bedarf in den Cartridge-Bereich nachzuladen, dürfte der Zumutungen eine der geringsten sein.

Einiges ist natürlich auch vom individuellen Geschmack abhängig. Beispielsweise die Tastenwiederholgeschwindigkeit ist vielen von uns viel zu gering. Wer einen größeren Speicher hat, dem ist gewiß mit einem schnelleren Speichertest nach Reset gedient. Eine halbe Minute kann ja so lang sein. Je nach Monitor benötigt man unterschiedliche Video-Initialisierungen, die man mit dem Videotool von Frank Vandecruys auf unserer Utilities-Diskette ermitteln kann, usw. usf... Es wäre natürlich das größte, wenn wir des fertigbrächten, eine Bibliothek von Optionen für individuelle Kernalversionen zu erstellen, natürlich im Rahmen der vollen Kompatibilität. Dirk Seifert und Andreas Stankewitz suchen weiter nach Mitarbeitern für dieses Projekt, damit es möglichst bald zum Abschluß gebracht werden kann.

Eine ähnliche Aktion zur Beseitigung von Macken im Basic würden wir sehr begrüßen. Frisch ans Werk!

#### Logikanalysator, Spracherkennung

Dies sind zwei Projekte, für die Jürgen Krause noch Komplizen sucht. Er hat alle Voraussetzungen, solche Projekte durchzuführen, aber für einen allein gibt es da doch etwas zu viel zu tun.



## Kleine Tips und große Bitten

Wie schon an anderer Stelle erwähnt, ist es ein bedeutsamer Unterschied, ob man ganz privat für sich alleine programmiert, oder ob man die Bedürfnisse anderer Anwender eines Programms voll berücksichtigt.

Schließlich haben wir uns zusammengefunden, um einander an den Früchten unserer Arbeit laben zu lassen, weil geteilte Freude doppelte Freude ist. Nun, ein paar Kleinigkeiten, die geeignet sind, die Freude anderer zu mehren, machen nicht viel Mühe, und ich möchte hier darum bitten, die kleinen Tips zu beherzigen.

1. Es ist äußerst unangenehm, wenn die Bildschirmdarstellung verschwindet, weil ein Programmautor sich nicht vorstellen kann, daß es Mitbenutzer gibt, die ein ASCII-Kernal benutzen. Also sollte vor dem Umschalten auf den alternativen Zeichensatz beispielsweise:

```
if peek(57352) = 4 then print chr$(27)+"z"  
(Wenn DIN, dann alternativer Zeichensatz).
```

Desweiteren sollten bei der Benutzung der Grafikzeichensätze die DIN-Anwender nicht vergessen werden.

2. Ich habe in diesem Heft einen ziemlich akzeptablen Vorschlag zur Menuegestaltung gemacht. Wem es dennoch beliebt, die Funktionstasten für die simple Abfrage von Programmverzweigungen zu benutzen, der möge einfach den Funktions-tastenvektor verbiegen mit `poke 949,249:poke 950,244`. Die Funktionstasten produzieren dann `chr$(224)` bis `chr$(243)`.

3. Hat man irgendwelche Grundeinstellungen verändert, sollte beim Verlassen des Programms der ursprüngliche Zustand wiederhergestellt werden, das heißt auch, daß sämtliche vom Einschaltzustand abweichenden Editorfunktionen rückgängig gemacht werden sollten. Der Funktionstastenvektor ist bei DIN auf `$E957` und bei ASCII auf `$E92A` zurückzusetzen mit `poke 950,233` und bei ASCII `poke 949,42`, bei DIN `poke 949,87`, falls man ihn nach der obigen Methode verbogen hat, und die ursprünglichen Tastenbelegungen, die oft in mühsamer Kleinarbeit erstellt waren, stehen wieder zur Verfügung. Sollte die default-Bank verstellt sein, bitte ich, sie beim verlassen eines Programms wieder auf 15 zu setzen.

4. Sehr viele Basicprogramme laden Assembler-Programmteile nach. Hier empfiehlt sich in jedem Fall eine programmäßige Abfrage nach der Gerätenummer, von dem das Haupt- bzw. Startprogramm geladen worden ist, mit `bload "Quatsch", u(peek(159))`.

Um das unschöne Nachladen solcher Programmteile zu vermeiden, gibt es ja noch die Möglichkeit, sie in Datass zu packen. Ein Programm hierfür von Pierre Teubner ist auf der Basic-Utilities-Diskette zu finden. Allerdings dauert die Data-Pokerei tierisch lange, ewige 4,7 Sekunden für ein Kilobyte.

Ich habe eine Macro-Basic-Routine in Arbeit, die das in 0,2

Sekunden erledigt. Die Programme werden - in listbarer Form !  
! - verschlüsselt und in Strings gepackt, die nach der Entschlüsselung am Stück auf ihren Platz geschoben werden.

### Tips und Tricks

Jörg Linnig

#### Profitext - CBM 8028 - Umlaute

Bei der Verwendung des Profitextmoduls mit dem Drucker CBM 8028 kann es dazu kommen, daß die Umlaute verloren gehen. Und zwar immer dann, wenn eine Gerätenummer angegeben wird. Wir nur F1 RETURN eingeben, dann funktioniert es.

#### Inverse (reverse) Directory-Einträge

Möchte man einen Directory-Eintrag invers darstellen, z.B. um zu verdeutlichen, daß jetzt der Inhalt einer anderen Diskette hinzukopiert wurde, so läßt sich das relativ einfach bewerkstelligen. Und zwar saved man ein entsprechendes File (z.B. einen leeren Speicher, oder irgendwelche Informationen in rem-Zeilen) wie folgt ab: dsave "SHIFT-SPACE RVS Filename".

Der Directory-Eintrag beginnt dann mit zwei Gänsfüßchen gefolgt von dem invers dargestellten Filenamen inclusive dem invers dargestellten 'prg'. Daß das 'prg' auch invers dargestellt wird, läßt sich glaube ich nicht verhindern, da die Eingabe von RVS-OFF keine Wirkung hat. Aber vielleicht gibt es ja auch noch andere Möglichkeiten solcher Manipulationen.

#### Strukturierte Programmierung in Basic, Großbuchstaben und Umlaute in Kommentarzeilen

Auch in Basic läßt sich strukturiert programmieren. Dies verlangt allerdings wesentlich mehr Disziplin ab als das in anderen Programmiersprachen wie z.B. Pascal der Fall ist. Eine große Hilfe hierbei ist Macro-Basic. Aber auch unabhängig davon läßt sich optisch schon einiges machen. Auch mit CBM-Rechnern lassen sich Programmzeilen einrücken. Und zwar dadurch, daß die einzurückende Zeile mit einem Doppelpunkt beginnt. Folgendes Beispiel soll das verdeutlichen:

```
10 rem" Demo - Programm
20 for i=0 to 25
30 :   for j=i to 25
40 :       a(i,j)=a(j,i)
50 :   next j
60 next i
70 end
```

Gleichzeitig soll dieses Beispiel die Möglichkeit verdeutlichen, Großbuchstaben und Umlaute in rem-Zeilen zu verwenden, nämlich durch die Verwendung von Gänsefüßchen.

#### Unzulässige Zeichen im Directoryeintrag

Wem ist das nicht schon passiert, daß er bei der Verwendung des Profitext-Moduls den Doppelpunkt hinter dem Klammeraffen vergessen hat, oder hat sonst irgendwie einen Directoryeinttrag erzeugt, der sich nicht mehr so ohne weiters löschen läßt? Eine Möglichkeit, das DOS zu überlisten besteht darin, alle für das DOS ungültigen Zeichen im Filenamen durch ein '?' zu ersetzen. Auf diese Art und Weise bleiben einem dann meist irgendwelche Byte-Manipulationen auf der Diskette erspart.

#### Directory auf Drucker ausgeben

Wenn man das Directory schwarz auf weiß haben möchte, dann sollte eigentlich folgender Weg gangbar sein:

```
open 4,4 : cmd 4 : directory print#4 : close 4
```

Leider stören sich Floppy und Drucker gegenseitig so, daß manchmal ein paar Zeichen verloren gehen. Darum muß auf den Erhalt des Basic-Textes verzichtet und das Directory in den Basic-Speicher geladen werden:

```
load"$",8 open 4,4 : cmd 4 : list print#4 : close 4
```

Damit kommen wir wieder zu der umständlichen Art des C-64, sich Directories anzusehen, aber so geht wenigstens nichts verloren.

Damit wären wir auch gleich dabei zu beschreiben, wie sich Basic-Texte und Directories als sequentielle Files auf Diskette schreiben lassen:

```
open 8,8,8,"filename,s,w" : cmd 8 : list print#8 : close 8
```

#### Nützliche Pokes

Rainer Ehrh

Stop Taste Ein/Aus.....Poke 789,249/250

List Befehl Ein/Aus.....Poke 647,137/138

Warmstart.....SYS 32786

Get Befehl: Cursor sichtbar machen.....  
.....bank15:poke55296,10:poke55297,peek(212)

Get Befehl: Cursor aus.....bank15:poke55296,10:poke55297,15

Bildinhalt auf Schirm zentrieren:....(default fuer XX = 80)..  
.....bank15:poke55296,2:poke55297,XX

Wichtig: Die Pokes mit den Adressen 55296 und 55297 müssen in einer Zeile stehen.

## Ein kühner Plan

Run Homepage Load See Print Hardcopy Other Extensions Monitor Copy Quit

**RUN BASIC-program**

## CONCLUSION

UNIVERSAL CON-SELESTHILLE DISK-UTILITY

| unit 8 | drive 8 | name             | diskkill | 1       | 0x200          | 10  | 01  | dos 2c     | 2482           | blocks | free              |
|--------|---------|------------------|----------|---------|----------------|-----|-----|------------|----------------|--------|-------------------|
| 1      | h       | hft run          | 220      | 10-     | filecopy       | 2.0 | 430 | 10-        | regenerate     | 660    | disk sektor 620.1 |
| 2      | l       | go .scr          | 230      | 2.0     | q              |     | 440 | file       | protector      | 660    | sector.pgm        |
| 3      | o       | directory .dok   | 240      | 2.0     |                |     | 450 | loesch     | schutz         | 660    | sector.ctl        |
| 4      | u       | allgemein        | 250      | bas.    | unitcopy       |     | 460 | umschalten |                | 670    | t's 1001          |
| 5      | p       | in.Dok ! .bas    | 260      | disk    | copy.bas       |     | 470 | change     | address        | 680    | 3/ display t & s  |
| 6      | o       | 10-dirsoft       | 270      | 3/      | copy all       |     | 480 | ds=        | disk           | 690    | display t&s       |
| 7      | o       | 3/filemaster     | 280      | copier  | bank1          |     | 490 | turbo      | dos 2.7        | 700    | 3/ t&s            |
| 8      | o       | disk handling    | 290      | copier  | bank4          |     | 500 | TURBO      | SFD            | 710    | Verzeichnis       |
| 9      | o       | disk helper      | 300      | file    | copy           |     | 510 | Monitor    | +              | 720    | disk verz. 4040   |
| 10     | o       | disk hand        | 310      | ub      | filecopy       |     | 520 | disk       | mon. celle     | 730    | 3/ disk-verzeichn |
| 11     | o       | 1388/diskmaster  | 320      | convert | copy           |     | 530 | disk       | mon. Künne     | 740    | disk verz.single  |
| 12     | o       | file wandlung    | 330      | IS=     | copy v4.2      |     | 540 | 3/         | disk mon. künn | 750    | 3/ master directo |
| 13     | o       | name + id change | 340      | 3/      | backup sfd=sfd |     | 550 | IS=        | flomo v1.2     | 760    | 3/ master index   |
| 14     | o       | disk los         | 350      | block   | kopie          |     | 560 | IS=        | flomo 620 v1.0 | 770    | 4040              |
| 15     | o       | disk manager     | 360      | bm      | blockcopy      |     | 570 | IS=        | flomo 620/15   | 780    | 4040 bam view     |
| 16     | o       | complete dir     | 370      | bm      | copy           |     | 580 | IS=        | flomo 620/3.1  | 790    | 4040 belegung     |
| 17     | o       | 3/ disk log.     | 380      | loesch  | u.ao           |     | 590 | IS=        | flomV1.3 druck | 800    | 4040 repair18s1   |
| 18     | o       | 3/ tartadresse   | 390      | loesch  | hilfe          |     | 600 | disk       | mon 8          | 810    | 4040 revive disk  |
| 19     | o       | dir=print        | 400      | 10      | filekill       |     | 610 | 3/         | clm-v1.0 !     | 820    | 4040 turbo        |
| 20     | o       | kopierhilfe      | 410      | disk    | file=retter    |     | 620 | 3/         | diskmon/comp   | 830    | 4040 t&s          |
| 21     | o       | copy unit=unit   | 420      | mscrsch |                |     | 630 | disk       | memory dump    | 840    | 4040 copy         |

Next Last Track Sect Read Write Edit **STATUS** + - > < Print Memory Bam Other Quit

first of act, links, bhm, directory track 39 \$ 27 sector 10 \$ 0H

[illegible][illegible]

Gerd Schumacher

Es ist mir ein tiefes Bedürfnis, Sie mit einem meines Erachtens wichtigen Programmiervorhaben bekanntzumachen, für dessen Verwirklichung ich etliche Mitglieder zu gewinnen hoffe. Es handelt sich um ein universelles Programmpaket zur Diskettenhandhabung, speziell für unsere CBM 600 und 700, das,

modular aufgebaut, wirklich sämtliche sinnvollen Routinen für diesen Zweck zusammenfaßt.

Jedes mal, wenn ich ein normales Directory sehe, ärgert mich dessen Unübersichtlichkeit und umständliche Handhabung. Hier Abhilfe zu schaffen ohne Zeitverlust war der Anlaß, mir weitergehende Gedanken zu machen. Das Problem, das Directory möglichst schnell in den Speicher zu laden, ist endlich gelöst. Zunächst habe ich es mit Macro-Basic in 14 Sekunden gemacht, dann habe ich Andreas genötigt, für diesen Zweck eine Assembler-Routine zu schreiben, die ebenso wie meine kleine Macro-Basic-Routine der Block-Verkettung folgt. Die 8,3 Sekunden, die dieses Programm braucht, haben mich auch nicht ganz befriedigt. Wenn das DOS einen Directoryeintrag gefunden hat, überträgt es die Daten, der Blockverkettung folgend, schließlich viel schneller. Dirk Seifert hat die Routine, die das tut, endlich gefunden. Hiermit braucht man nur schlappe 4,8 Sekunden. Die Zeitangaben beziehen sich auf ein rasselvolles Directory auf der SFD 1001 bzw. 8250 LP mit 224 Einträgen, und wer hat das schon! Bis der 224. Eintrag in einem solchen Directory auf dem Bildschirm erscheint, vergehen bei der normalen Directory-Funktion übrigens 25 Sekunden, die es zu unterbieten gilt.

Die Hardcopy eines Directories, wie ich es mir vorstelle, ist am Anfang dieses Artikels zu bewundern. Die Anzeige der Blockanzahl ist normalerweise verhältnismäßig unwichtig, und so wird zunächst nur eine laufende Nummer, der Dateityp (nur Anfangsbuchstabe, Großbuchstabe bei Schreibschutz) und der Dateiname angezeigt. Optional ist, da die Information ja schon im Rechner vorliegt, auch die Anzeige eines erweiterten Directories einschließlich 1.Spur und 1.Sektor möglich. In beiden Anzeigearten kann man nun ohne weiteren Diskettenzugriff blättern.

So hat man zunächst einmal einen guten Überblick über den Disketteninhalt, und eine vorzügliche Grundlage zur Erstellung oder Anpassung der verschiedensten Hilfsprogramme, die über ein später zu beschreibendes Menu ausgewählt werden können. Als Grundeinstellung ist vorgesehen, durch die Eingabe der angezeigten laufenden Dateinummer und return ein Basicprogramm zu laden und zu starten.

Die Grundroutine soll weiterhin u.a. das Listen und Drucken von Dateien ohne Programmverlust, das Laden von Dateien an beliebige Speicherstellen, umfassende Dateinformationen (Start- und Endadresse, Dateilänge) sowie die Möglichkeit, weitere Programme anzuhängen und auszuführen beinhalten. Beispielsweise könnte ich mir vorstellen, daß die verschiedensten Kopierprogramme in dieses Paket eingebunden werden.

Der Umfang der hier anstehenden Arbeiten macht eine Arbeitsteilung unbedingt notwendig. Ich glaube, ein wirklich ordentliches Konzept entwickelt zu haben, dessen vollständige Beschreibung derzeit noch von weniger allgemeinem Interesse sein dürfte. Wer an einer Mitwirkung ernsthaft interessiert ist, möge sich vertrauensvoll an mich wenden.

Leider ist unser Standard-Basic nicht geeignet, in akzeptabler Zeit die hier erforderlichen Funktionen auszuführen. So würde allein schon die Umwandlung des in den Speicher geladenen Directorys in eine Stringvariable per 'peek' ungefähr doppelt so lange dauern wie die Anzeige eines normalen Directorys. Andererseits steht der Programmierung der zeitkritischen Programmteile in Assembler unser relativ kleiner standardmäßig verfügbarer Speicher in der Systembank entgegen. So dürfte es am sinnvollsten sein, eine kleine Macro-Basic-Erweiterung hinzuzufügen.

### Acht Kilo Spaghetti commentata

Das im folgenden beschriebene Programm könnte durchaus als Negativbeispiel für die Qualität freier Software dienen. Dennoch erdreisten wir uns, auch noch vier Seiten Basic-Spaghetti abzudrucken. Eigentlich wollte ich nur ein paar sinnvolle Programmierkniffe und ein Beispiel für Programmergonomie durch gute Menüführung und ein ordentliches Bildschirm-Layout geben. Dann hatte ich die blöde Idee, das alles 'mal eben' zu einer Art Programm zusammenzumatschen, damit man sich etwas darunter vorstellen kann. Das so entstandene Machwerk ist also eher Hand- als Kopfarbeit und über alle Maßen fehlerbehaftet. Dennoch enthält es bemerkenswerte Einzelteile mit denen ich hoffe, Ihnen eine Freude machen zu können.

In absehbarer Zeit wird es einen richtigen Diskmonitor als Bestandteil der oben beschriebenen universellen Diskutility geben. Der wird dann ordentlich programmiert und trotz Basic doppelt so schnell. Mache sich also niemand die Mühe, die abgedruckte Demoversion zu debuggen, es lohnt sich nicht! Wer eine Clubdiskette bestellt, dem wird das Programm auf Wunsch zusätzlich draufgespielt; einzeln ist es nicht verfügbar.

Auf die Idee, mir über einen Diskmonitor Gedanken zu machen, bin ich gekommen, da ich bemerkte, daß alle Basic-Lösungen wegen ihrer Lahmheit kaum zum arbeiten zu gebrauchen sind, und daß mir weder aus eigener Anschauung noch vom Hörensagen ein Programm bekannt ist, das meinen Mindestanforderungen in jeder Hinsicht gerecht wird. Die mir bekannten kommerziellen Programme beispielsweise gestatten keinen Zugriff auf eine 8050-Diskette mit der 8250. Die 8280 wird von solchen Programmen nie unterstützt, und einer wirklich defekten Diskette, z.B. mit überschriebener BAM, kann man im schlimmsten Fall kein einziges Bit mehr entlocken, obwohl sie vom Basic aus einwandfrei zu lesen ist. Außerdem war ich mit keinem Editor wirklich zufrieden, und so glaube ich, den denkbar besten geschrieben zu haben, wenigstens im Ansatz.

### Die Beschreibung von oben bis unten

Da der 8028 keine Grafikdarstellung kennt, sind die Steuerzeichen in eckige Klammern gesetzt. Etwas zu kurzgeratene Abkürzungen sind:

[cu d] = cursor down    [cu r]    = cursor right  
[cu l] = cursor left    [cu u]    = cursor up  
[sp]    = space            [sh ...] = shift ...

Zeile 50: Dimensionierungen gehören aus stilistischen Gründen und - im Falle CBM 500-700 wegen gelegentlicher Probleme mit der "garbage collection" an den Programmanfang. "tl%" und "sl%" sind vorgesehen, später einmal Dateiverkettungen rückwärts verfolgen zu können. "ml%" beinhaltet Erläuterungen zu den verschiedenen Menueoptionen, "m0" die Position der Leerzeichen in der Menuezeile, "c%" die Hexzahlen 00 bis FF, "ch" die Tabelle zur ASCII-Darstellung von peek-Werten, "fd%" Diskettentyp und DOS-Version, "dh" Screencode der Hex-Ziffern 0 bis H. Die Codewandlung über Tabellen ist so viel schneller als eine rechnerische, daß man die verhältnismäßig lange Zeit zur Erstellung derselben durchaus akzeptieren kann. Übrigens ist dieses Verfahren sogar noch wesentlich schneller als die entsprechenden Macro-Basic-Befehle, die ja beim CBM 600/700 durch die - wenn auch sehr schnelle - Bankumschaltungsroutine gebremst werden.

Zeile 100: Ist es nicht hübsch, aufeinanderfolgende Inputs in eine Zeile zu packen und zu löschen, wenn sie nicht mehr gebraucht werden?

Zeile 130: Die Laufwerkserkennung ist noch unvollständig. Demnächst wird auch die SFD nicht mehr als 8250 erkannt werden. Beim ersten mal müßte es auch mit der 8280 klappen. Die 2031 wird richtig erkannt, sonst habe ich nichts getestet.

Zeile 210 bis 230: wird durch einen Macro-Basic-Befehle ersetzt werden. Zusätzlich steht dann die Anzahl der freien Blöcke zur Verfügung.

Zeile 270 baut einen wunderhübschen Rahmen um die in HEX und in ASCII darzustellenden Zeichen.

Zeile 290: Der Inhalt eines Sektors wird in den Bildschirmspeicher geschrieben statt, wie üblich, in Variablen. Die Speicherstellen 55248 bis 55296 werden normalerweise nicht dargestellt (BS-Zeile 26). Beim 600er kann man sie aber sichtbar machen, indem man 'poke 55297,25' durch 'poke 55297,26' ersetzt. So kann man sich den gerade aktuellen Block zusätzlich zur hex- und ASCII-Darstellung noch im Screencode ansehen. Sollte dies aber unerwünscht sein, kann die Darstellung der letzten drei Zeilen durch 'poke 55297,22' ganz unterbinden. So ist es im übrigen auch möglich, den Bildschirmspeicher ganz unauffällig für Assemblerprogramme zu benutzen.

Nun wird angezeigt, welcher Spur/Sektorwert aktuell ist. Als erstes wird stets der erste Directoryblock gelesen und angezeigt. Danach werden die Variablen für das Hauptmenue erstellt und zum Menue-Unterprogramm gesprungen. Auf letzteres komme ich am besten sofort zu sprechen.

## TÄTERÄTÄ: DAS MENUE

Ein wesentlicher Faktor für die Handhabbarkeit von Programmen ist eine vernünftige Menüführung unter Erhaltung notwendiger Informationen aus dem Programm mit einer möglichst ruhigen Bildschirmdarstellung. Ich finde die Erzeugung ganzer Menueseiten in laufenden Programmen - es sei denn beim Programmstart - im allgemeinen hinderlich. Außerdem sollte ein prägnanter Zusammenhang zwischen der gewünschten Funktion und der zu deren Auslösung notwendigen Eingabe bestehen. Oftmals wird, wohl um der vermeintlich einfachen programmmäßigen Auswertung willen, bei unseren Programmen die Eingabe einer Ziffer verlangt. Zahlen sind mir zwar ebenso lieb wie Buchstaben, aber ein sinnvoller Zusammenhang zwischen Funktion und Ziffer dürfte hier allenfalls irgendwo in parapsychologischen Abgründen auszumachen sein.

Abgesehen von einigen Anwendungen, für die unser Computer nicht vorgesehen ist, ist die vorzustellende Methode Mäusen, Freudenknüppeln und allen verwandten Hilfsmitteln für Körperbehinderte haushoch überlegen. Ich muß sagen, ich war fürbaß erstaunt, wie wenig aufwendig es war, diese Menueform, die ich beim "Superdiskdoc" abgesehen habe, in Basic nachzuprogrammieren.

Die erste Zeile gibt jeweils Stichworte für die möglichen Programmverzweigungen an. Mit den Tasten 'cursor rechts' und 'cursor links' kann man im Menue sozusagen "blättern". Das jeweils aktuelle Stichwort wird invers dargestellt und kann mittels 'return' aktiviert werden. In der zweiten Zeile werden Erläuterungen zu den Stichworten gegeben.

Die Auswahl über cursor ist natürlich nur zum blättern im Menue sinnvoll. Sonst werden die Funktionen durch Eingabe des ersten Buchstaben der Stichwörter ausgelöst. Zur Vermeidung von Unheil durch Tippfehler kann zusätzlich eine Bestätigung durch 'return' eingebaut werden.

Die dritte Zeile ist für Meldungen aus dem Programm und für Eingaben vorgesehen. Im vorzustellenden Beispiel enthält die vierte Zeile Informationen über die angesprochene Diskettenstation und die zu bearbeitende Diskette. Bevor Sie sich dranmachen, das Listing abzutippen, kann Ihnen die Hardcopy des vierspaltigen Directorys eine ungefähre Vorstellung vermitteln.

Mögen auch manchen Programmteilen kryptographische Qualitäten anhaften, der Menueteil dürfte durchaus geeignet sein, auch von Durchblick- Anfängern adaptiert zu werden.

Die zu übergebenden Parameter (Zeilen 820-870 und 900-950)

m0 = Position der Leerzeichen vor und hinter den Stichworten in der Menuezeile, wird für Invertierung gebraucht.

m0\$ = Menuezeile



m4\$ = Anfangsbuchstaben der Stichwörter, Gemeine und Versalien

m7\$ = 'return', 'cursor links' und 'cursor rechts'

Die pokes in Zeile 870 und 950 stellen den Cursor ab.

In Zeile 980 werden die beiden oberen Zeilen dargestellt, und dann zum Unterprogramm Tastaturabfrage verzweigt. Eine eventuelle Eingabe wird zunächst auf return und Cursor links-rechts überprüft und anschließend auf m4\$. Ist eines dieser Zeichen eingegeben worden, wird ein entsprechender Wert in 'm5' hinterlassen, der die laufende Nummer der ausgewählten Funktion darstellt.

Wenn 'return' erkannt wird, erfolgt ein Rücksprung zur Sprungleiste in den Zeilen 880 oder 960. Wird mit einer Cursortaste 'geblättert', erfolgt lediglich eine aktualisierte Anzeige und ein Sprung zur erneuten Eingabe.

In Zeile 1060 wird festgelegt, daß im Falle der Festlegung eines neuen 'm5'-Wertes durch eine gültige Buchstabentaste einen Rücksprung zur Sprungleiste, alle größeren aber nach der Aktualisierung der Anzeige eine neue Eingabe erwarten entweder zur Korrektur der Eingabe oder zur Bestätigung durch return.

Die Zeilen 310 bis 720 beinhalten Aktionen, zu denen das das Hauptmenue führt; die Zeilen 1080 bis 1130 werden vom Untermenue aus bedient.

Erwähnenswert sind vielleicht noch die Zeilen 360ff zur wahlweisen hex-dez-Eingabe. Da ist auch noch eine Macke drin. Vielleicht weiß ja noch wer eine elegantere Methode.

### Bildschirmfenster

Nun noch ein paar Worte zum Editor. Einigermaßen nützlich dürfte meine Methode zur absoluten Definition von Bildschirmfenstern sein (Zeile 750). Die Parameter 'w1' und 'w3' reichen von 0 bis 79, die Parameter 'w2' und 'w4' von 0 bis 24. Der Cursor steht jeweils oben links.

### Die Darstellungsweise

Die Darstellungsweise ist weitgehend identisch mit der des Superdiskdoc von Precision Software. Die optional zusätzliche Darstellung im Bildschirmcode und die genaue Angabe der Diskettenstation und DOS-Version ist auf meinem Mist gewachsen. Künftig soll im Editiermodus noch die Nummer und der Wert des jeweiligen Bytes unter dem Cursor zusätzlich dezimal dargestellt werden. Die für Meldungen aus dem Programm und Inputs benutzte Zeile drei soll ansonsten die Angaben über die nächste und vorherige Spur sowie über den nächsten und vorherigen Sektor enthalten. Vor den angezeigten Hex-Werten wird jeweils ein Leerzeichen gedruckt. Ist das Byte null eines Blocks = 0, was bekanntlich angezeigt, daß es sich um den letzten Block einer Datei handelt, werden die Leerzeichen vor den von der

Datei nicht benötigten Bytes invertiert.

Nun wird zunächst der jeweilige Block auf Wunsch sichtbar in den Bildschirmspeicher geschrieben. (Zeile 550) Als nächstes wird ein Fenster in dem vorbereiteten Rahmen für die Hex-Darstellung eröffnet, und mit den Zeilen 570 oder 580 die Hex-Darstellung vorgenommen. Die Zeile 590 schließlich besorgt die Darstellung in ASCII, wobei die nicht druckbaren Zeichen in Punkte umgewandelt werden.

### Hex- und ASCII-Eingabe

In aller Bescheidenheit geruhe ich, den Editor als einen mitelmäßigen Knüller zu bezeichnen, es muß ja nicht immer einer von den ganz großen sein. Was mich bei den meisten Programmen dieser Art stört, ist, daß man überhaupt Zugriff auf die Leerzeichen zwischen den Hexwerten hat, und daß nur zu oft Fehleingaben zugelassen werden. Nun, mit Fenster öffnen, und schließen, zusätzlich inverser Darstellung aller eingegebenen Werte, auch in ASCII, habe ich als Anfänger hierfür gewiß zwei Stunden und elf Zeilen gebraucht. Für geübte Hacker ist es wohl auch etwas langsam, aber immerhin. Wie man dazu kommt, wie u.a. der Programmierer des TIM-Monitors, ähnlichen Komfort in Assemblerprogrammen wegzulassen, ist mir ein Rätsel.

Als ich den Editorteil für die ASCII-Eingabe gebastelt habe, war ich schon ein wenig geübter. Er braucht lediglich vier Zeilen, und befriedigt in der Geschwindigkeit auch recht hohe Ansprüche.

Zwischen beiden Editiermodi kann mit der 'Pfeil nach links - Taste' bzw. dem Unterstreichungsstrich (DIN) hin- und hergeschaltet werden. Return führt zurück zum Hauptmenue.

### Schlußbemerkungen

Das vorgestellte Programm ist wirklich nur ein blasser Abglanz dessen, das ich in Kürze zu verfassen gedenke. Ich fand es einmal ganz reizvoll, ein einigermaßen schnelles Programm mit dem Standard-Basic zu schreiben für eine Aufgabe, zu der dieses sich nur sehr mangelhaft eignet. Die Version 1 wird wesentlich kürzer, durchschnittlich etwa doppelt so schnell, sauber programmiert, mehr Funktionen enthalten, mehr Informationen darstellen und obendrein noch zuverlässig arbeiten. Beachten Sie bitte, daß im vorgestellten Demo-Programm keine einzige Fehlerabfrage enthalten ist!

Was übrigens die Geschwindigkeit anbelangt, von der Spur/Sektor-Angabe bis zur Vollendung der gleich dreifachen Darstellung werden schlappe acht Sekunden gebraucht. Das nächste mal werden es knapp drei, und das mit Basic!

Dieses Programm taugt nicht zum Arbeiten !! Es soll lediglich einen Eindruck von der Qualität eines Programmpakets vermitteln, das derzeit entwickelt wird. Die nächste Version wird unvergleichlich besser.

```

30 print"[2xhome][clr-home][3xsp]DISKMONITOR CBM 600/700
   von Gerd Schumacher fuer die CBM-Selbsthilfe e.V."
40 print"[rvs][3xsp]Version 0.001 gedacht als Unterprogramm
   einer universellen Diskutility[6xsp][rvs-off]":
   rem ***** zuerst alle dimensionierungen und variablen *****
50 dim t1$(4000): dim sl$(4000): dim ml$(12): dim m0(12): dim
   c$(255): dim ch(255): dim fd$(9): dim dh(15): print chr$(2
   7)"h"
60 b=176: for a=0 to 9: dh(a)=b: b=b+1: next: b=193: for a=10 to
   15: dh(a)=b: b=b+1: next
70 a$="0123456789ABCDEF": a=0: for b=1 to 16: c$(a)=mid$(a$,
   b,1): for c=1 to 16: c$(a)=mid$(a$,b,1)+mid$(a$,c,1): a
   =a+1: next: next: co$="abcdef"
80 for a=0 to 31: ch(a)=174: next: for a=32 to 63: ch(a)=a+128:
   next: for a=64 to 95: ch(a)=a+64: next: for a=96 to 127:
   ch(a)=a+64: next: for a=128 to 159: ch(a)=174: next:
90 rem ***** endlich sind die arrays fertig *****
100 print chr$(27)"m"chr$(27)"q[home][cu d]": input"unit"
   ;un: print spc(10)"[cu u]";: input"drive";dr: print"[cu u]"
   chr$(27)"j"chr$(27)"q"
110 rem ***** identifiziere laufwerk *****
120 open l5,un,l5: open l,un,2,"#"
130 print#l5,"m-r"chr$(255)chr$(255): get#l5,c$: c=asc(c$)
140 if c=254 then fd=1: else if c=226 then fd=2: else if c=213
   then fd=4: else if c=242 then fd=5: else if c=255 then fd=8
150 print#l5,"m-r"chr$(234)chr$(16): get#l5,cb$: cb=asc(cb$):
   if cb=0 then fd=4
160 print#l5,"m-r"chr$(172)chr$(16): get#l5,cc$: cc=asc(cc$):
   if cc=1 then fd=6: else if cc=2 then fd=7
170 fd$(1)="2031 dos 2.6": fd$(2)="2040 dos 1.2":
   fd$(3)="4040 dos 2.1": fd$(5)="8050 dos 2.5":
   fd$(4)="4040 dos 2.7": fd$(6)="8050 dos 2.7":
   fd$(7)="8250 dos 2.7"
180 fd$(8)="8280 dos 3.1"
190 if fd=8 then 220: else if fd>4 then 230
200 rem ***** header und id lesen *****
210 he$="": print#l5,"ul";2;dr;18;0: print#l5,"b-p";2;13
   8: for a=0 to 25: get#l,x$: if x$="" then x$=chr$(32): else
   he$=he$+x$: next: tr=18: se=1: goto 240
220 he$="": print#l5,"ul";2;dr;38;20: for a=0 to 25: get#l,x
   $: if x$="" then x$=chr$(32): he$=he$+x$: next: tr=38: se=
   0: goto 250: else he$=he$+x$: next: tr=38: se=0: goto 250
230 he$="": print#l5,"ul";2;dr;39;0: for a=0 to 25: get#l,x$
   : if x$="" then x$=chr$(32): he$=he$+x$: next: tr=39: se=1
   : else he$=he$+x$: next: tr=39: se=1
240 print"[rvs][21xsp]"chr$(27)"j unit";: printusing
   "##";un;: print"[2xsp]drive "dr;" "fd$(fd)"[2xsp]";
250 print mid$(he$,7,18)"id "right$(he$,2)"[9xsp]blocks
   free[3xsp][rvs-off]"
260 rem ***** kompletter rahmen *****
270 print"[2xhome][4xcu d][rvs][10xsp]";: for a=0 to 15:
   print c$(a)" ";: next: print"[3xsp]"a$"[3xsp]";:
   for a=0 to 240 step 16: print"[rvs][3xsp][rvs-off] "c$(a)"

```

```

[rvs][2xsp]"spc(49)"[2xsp]"spc(18)"[2xsp]";: next:
for a=1 to 8: print"[10xsp]";: next: print"[rvs-off]"
280 rem ***** 32 byte loeschen, anzahl zeilen,
    erster sektor ***
290 for a=55248 to 55296: poke a,32: next: poke 55296,6: poke 55
    297,25: gosub 530: gosub 820: goto 550: end
300 rem ***** eingaben + meldungen *****
310 get o$: if o$="" then 310: else if o$=chr$(13) then goto 870
320 print"[2xhome][2xcu d] not yet available[4xsp]";:
    goto 310
330 if peek(55008) then tr=peek(55008): se=peek(55009): gosub
    530: goto 550: else print"[2xhome][2xcu d] end of
    file-links";: goto 310
340 print"[2xhome][2xcu d] track[2xsp]";: gosub 380: t
    r=a: gosub 530: goto 870
350 print"[2xhome][2xcu d] sector ";: gosub 380: se
    =a: gosub 530: goto 870
360 rem ***** unterprogramm wahlweise hex- o
    der dez-eingabe ****
370 print"[cu u][8xcu r]";: goto 390
380 gosub 390: if a<256 then return: else if a<255 then goto 370
390 a=1: x$="": inputx$: if left$(x$,1)="$" then x$=right$(x
    $,2): goto 420
400 for b=1 to len(x$): if instr(left$(a$,10),mid$(x$,b,1))
    =0 then a=0
410 next:if a=1 then a=val(x$): return: else if a=0 then goto 370
420 y$=left$(x$,1): gosub 440: if b<0 then 370: else if b=>0 then
    a=b*16
430 y$=right$(x$,1): gosub 440: if b<0 then 370: else if b=>0
    then a=a+b: return
440 b=instr(co$,y$): if b then y$=chr$(asc(y$)+128)
450 b=instr(a$,y$)-1: return: rem ***** das war es schon *****
460 gosub 530: goto 550
470 print"[2xhome][2xcu d] Schluss mit 'ret'[3xsp]"
    : getx$: if x$="" then 470: else if x$<>chr$(13) then 940:
    else for a=1 to 15: close a: next: print chr$(27)"l"chr$(27)
    "g": print"[2xhome][clr-home][3xsp]ENDE": poke 55296,6: poke
    55297,25: new
480 rem ***** sektor schreiben *****
490 print"[2xhome][2xcu d] disk-data will be modified ! confirm
    by 'ret'";
500 getx$: if x$="" then 500: else if x$<>chr$(13) then 940
510 print#15,"b-p: ";2;0;: for a=55008 to 55263: print#1,chr$
    (peek(a));: next: print#15,"u2";2;dr;tr;se;: goto 870
520 rem ***** anzeige track sector + inputzeile loeschen *****
530 print"[2xhome][cu d][rvs]";tab(44);"[2xsp]track ";:
    printusing"###";tr;: print"[2xsp]"c$(tr)"[2xsp]
    sector ";: printusing"###";se;: print"[2xsp]"
    c$(se)"[rvs-off]": print chr$(27)"q": return
540 rem ***** sektor lesen *****
550 print#15,"u1";2;dr;tr;se;: for a=55008 to 55263: get#1,x$:
    if x$="" then x$=chr$(0): poke a,asc(x$): next: else poke
    a,asc(x$): next
560 w1=9: w2=5: w3=56: w4=20: gosub 750: b=0: if peek(55008)
    then 580
570 for a=55008 to (55008+peek(55009)): print" c$(peek
    (a));: next: for a=a to 55263: print"[rvs] [rvs-off]"
    c$(peek(a));: next: goto 590

```

```

580 for a=55008 to 55263: print "c$(peek(a));: next
590 w1=61: w3=76: gosub 750: for a=55008 to 55263: if peek(a)and
96 then print chr$(peek(a));: else print ".";
600 next: print "[2xhome]": goto 870
610 rem ***** editor hex-eingabe *****
620 w1=9: w2=5: w3=56: w4=20: gosub 750
630 print " ";: poke 55296,10: poke 55297,peek(212): rem
*** cursor an bei get ****
640 v=peek(203): x=(v-7)/3: y=int(x)+(peek(202)-5)*16: bl
$=c$(peek(55007+y))
650 getw$: if w$="" then 650: else if instr(co$,w$) then w$=chr$(
asc(w$)or128)
660 if w$="[cu d]" then printw$;: goto 640: else if w$="[cu u]"
then printw$;: goto 640: else if w$="[cu l]" then 700: else
if w$="[cu r]" then 700
670 if w$=chr$(13) then print "[2xhome]": goto 870: else if w$=
"_" then 770: else if instr(a$,w$)=0 then 640
680 if x=int(x) then bl$=left$(bl$,1)+w$: gosub 710: print "[rvs]
[cu l]"bl$"[rvs-off][cu r]";: goto 640
690 bl$=w$+right$(bl$,1): gosub 710: print "[rvs]"bl$"[rvs-off]
[cu l]";: goto 640
700 printw$;: if peek(203)/3-int(peek(203)/3)=0 then print
w$;: goto 640: else goto 640
710 a=(instr(a$,left$(bl$,1))-1)*16+instr(a$,right$(bl$,1))-1:
poke 55007+y,a
720 poke peek(202)*80+53308+x,ch(a): return
730 rem ***** bildschirmfenster *****
740 rem w1 links w2 oben w3 rechts w4 unten ***
cursor oben links ***
750 poke 201,(w1+53248+80*w2)/256: poke 200,w1+53248+80*w2-
peek(201)*256: poke 202,w2: poke 203,w1: print chr$(27)
"t[home]";: poke 223,w3: poke 221,w4: return
760 rem ***** editor ascii-eingabe *****
770 w1=61: w3=76: gosub 750: poke 55296,10: poke 55297,peek(212)
780 v=peek(203)-60: y=v+(peek(202)-5)*16: v=v*3+53255+80
*peek(202)
790 getw$: if w$="" then 790: else if instr("[cu d][cu u][cu l]
[cu r]",w$) then printw$;: goto 780: else if w$="_" then 620:
else if w$=chr$(13) then 940
800 if asc(w$)and96 then poke 55007+y,asc(w$): x=int(asc(w$
)/16): poke v,dh(x): x=asc(w$)-int(asc(w$)/16)*16: poke
v+1,dh(x): print "[rvs]"w$"[rvs-off]";: goto 780:else goto780
810 rem ***** die daten fuer das hauptmenue *****
820 data53248,53253,53258,53263,53269,53276,53281,5328
7,53293,53298: restore 820: for m1=1 to 10: readm0(m1): next:
m0$=" Next Last Edit Track Sector
Read Write Other Quit"
830 m1$(1)=" read next sector[26xsp]": m1$(2)=" read last sector
[25xsp]": m1$(3)=" edit (_ to switch ascii <--> hex)[7xsp]"
840 m1$(4)=" select other track[3xsp]dec. or hex $xx[6xsp]":
m1$(5)=" select other sector[2xsp]dec. or hex $xx[6xsp]":
m4$="nletsrwoqNLETsrwoq"
850 m1$(6)=" read selected sector[21xsp]": m1$(7)=" write sector
[30xsp]": m1$(8)=" select other unit, drive,or disk[8xsp]"
860 m1$(9)=" leave program[31xsp]": return
870 poke 55296,10: poke 55297,16: m9=len(m4$): m5=1: m7$=chr$(
13)+chr$(29)+chr$(157)
880 gosub 980: gosub 1000: on m5 goto 330,320,620,340,350,460,

```

```

480,900,470
890 rem ***** die daten fuer das untermenue *****
900 get o$: if o$="" then 900: else if o$<>chr$(13) then goto 870
910 data 53248,53255,53260,53266,53274,53280: for m1=1 to 6:
    read m0(m1): next: m0$=" Floppy Unit Drive Address Menue ":
    m1$(1)=" change disk 'return' when ready"
920 m1$(2)=" other unit[24xsp]": m1$(3)=" other
    drive[23xsp]": m1$(4)=" change unit address[15xsp]"
930 m1$(5)=" back to main menue[25xsp]": m4$="fudamFUDAM"
940 rem ***** untermenueaufruf *****
950 poke 55296,10: poke 55297,16: m9=len(m4$): m5=1: m7$=chr$(
    13)+chr$(29)+chr$(157)
960 gosub 980: gosub 1000: on m5 goto 1080,1100,1110,1070,1130
970 rem ***** anzeige zwei menuezeilen, loeschen inputzeile ***
980 print"[2xhome]"m0$chr$(27)"q": print"[rvs]"m1$(m5)"
    [rvs-off]": print chr$(27)"q[cu u]": for a=m0(m5) to m0(
    m5+1): poke a,(peek(a)or128): next: return
990 rem ***** hier die auswahl ueber tastatur *****
1000 get m8$: if m8$="" then 1000: else if m8$<>" then ma=instr
    (m7$,m8$)
1010 if ma=0 then 1050
1020 if ma=1 then return: else if ma=2 then m5=m5+1:
    else if ma=3 then m5=m5-1
1030 if m5=0 then m5=m9/2: else if m5>(m9/2) then m5=1
1040 gosub 980: goto 1000
1050 ma=instr(m4$,m8$): if ma=0 then 1000: else if ma>m9/2 then
    ma=ma-m9/2
1060 m5=ma: if ma<7 then gosub 980: return: else gosub 980:
    goto 1000
1070 rem ***** eingaben untermenue *****
1080 print"[2xhome][2xcu d] insert new disk
    ! return when ready": get o$: if o$="" then 1080:
    else if o$<>chr$(13) then 950
1090 print"[2xhome][2xcu d]"chr$(27)"t[clr-home]": gosub 1120:
    if mid$(he$,3,1)="c" then 230: else if mid$(he$,3,1)
    ="a" then 210: else if left$(he$,1)("&" then 220
1095 goto 1130
1100 print"[2xhome][2xcu d] ";: input"unit number";o:
    if o then un=o: for o=1 to 15: close o: next: print"[2xhome]"
    [2xcu d]"chr$(27)"t[clr-home]": goto 120: else goto 950
1110 dr=abs(dr-1): goto 1090
1120 for o=1 to 15: close o: next: open 15,un,15: open 1,un,2,"#":
    return
1130 gosub 810: goto 870

```

Ich will hier ein bißchen über mein Erfahrungen mit dem ema-Editor-Macroassembler berichten. Zunächst zum Editor.

Der Editor ist ein vernünftiger Programmeditor, der einem das Leben bei der Erstellung von Quelltexten nur selten schwer macht. Der Cursor befindet sich immer im Einfüge-Modus, was meistens auch sinnvoll ist. Eine Umschaltung auf Überschreib-Modus gibt es leider nicht. Da der Editor sich in keiner Weise an den Standard-Editor der 600/700er anlehnt, sind manche cursor-Bewegungen etwas gewöhnungsbedürftig. So kann man nicht von der Mitte einer Zeile direkt in die Mitte der nächsten gelangen, den cursor drängt es ständig an den Rand. Aber das ist wirklich nicht so schlimm.

Alle Befehle lassen sich über eine esc- oder ctrl-Kombination erreichen. Sämtliche üblichen Befehle zur Blockbearbeitung sind vorhanden. Der Befehl "Freien Speicherplatz anzeigen" zeigt einem, daß man viel Platz zum wüten hat, mehr als 50000 Zeichen. Ohne Kommentare paßt da ein 8k-Betriebssystem rein! Dann werden die Funktionen Blockanfang/-ende markieren, cursor an Textanfang oder -Ende setzen aber etwas nödelig, der Einfügemodus wird aber kein bißchen langsamer!.

Einige wenige Dinge des Editors sind mir allerdings so auf die Nerven gegangen, daß ich einen Eingriff in das Programm nicht gescheut habe. Die Erfolge dessen sind im b0x-ema auf der Diskette Assembler I v2.0 verewigt.

Zunächst haben mich die Haken, die das Zeilenende markieren sollten, gestört. Für Leute, die einen zeilenorientierten Editor nicht gewöhnt sind, mag das ja ganz gut sein, mir hat es die Übersichtlichkeit der Texte genommen, darum habe ich dem Editor die Haken genommen.

Dann hat der Ausstieg aus dem Editor (ctrl-x) nicht mehr einen Basic-Kaltstart zur Folge, sondern einen Reset, der im allgemeinen einen Warmstart nach sich zieht.

Nachdem ich mit unbeabsichtigtem Drücken der ctrl-z Taste (ganzen Text löschen) die Arbeit von zwei Stunden fast zunichte gemacht habe (der Text ließ sich noch vom Monitor aus retten), habe ich die benannte Funktion, ctrl-x und ctrl-j (Block löschen) auf die entsprechenden Großbuchstaben-Tasten gelegt.

Die einzige Funktion, die mir wirklich manchmal fehlt ist die, eine ganze Zeile löschen zu können.

Die Neubelegung der Funktionstasten ist jetzt möglich, da im b0x-ema ein sanfter Ausstieg (siehe Oben) und wieder Einstieg (d.h. ohne Textverlust) möglich ist. Die Tasten F3 bis F8 können von Basic aus neu definiert werden.

Nun zum schönsten Teil, der Assembler. Er ist der schnellste,

der mir je untergekommen ist. Kurze Routinen sind fast schneller assembliert, als man auf die RETURN-Taste gedrückt hat (aber nur fast). Bei langen Dateien, die nicht in den Textspeicher passen, ist die Floppy das tempobestimmende Glied. Die Assemblierungszeit kann vernachlässigt werden.

Es wird auf Diskette oder in die Bank 15 assembliert, letzteres hat den Vorteil, daß man bei genügend RAM in dieser Bank die Routinen sofort austesten kann, und den Nachteil, daß einem das ROM manchmal im Wege ist. Vielleicht erbarmt sich ja mal einer und verpaßt dem Assembler eine Bank-Anweisung.

Mit dem Editor lassen sich unter der Verwendung der tab-Taste sehr übersichtliche Source-Texte erstellen (mit vielen Kommentaren!), die der Assembler mühelos frisst, da er tab und space gleichermaßen bearbeitet.

Die Macrofähigkeiten des Assemblers habe ich noch gar nicht ausprobiert, weil ich immer zu faul bin, mir die Macros zu schreiben. Aber die bedingte Assemblierung läßt es sogar zu, sich Macro-Bibliotheken zu schreiben, sodaß sich hoffen läßt, daß wir irgendwann einmal eine Diskette mit hoffentlich vielen, vielen Macros anbieten können (z.B. Basic-ähnliche Befehle wie open, load, print...).

Zu guter letzt kann ich nur sagen, daß ich mit dem ema im großen und ganzen sehr zufrieden bin und empfehle ihn wärmstens weiter.

Da fällt mir noch ein zu sagen, daß Dirk Seifert und ich mit der Hilfe von Frank Grotelüschen derzeit im ema herumpörkeln, um ihn leistungsfähiger und noch bedienungsfreundlicher zu machen. Wir werden sehen, was dabei herauskommt und unsere Ergebnisse sicher nicht verheimlichen.

#### Mal was anderes - FORTH

Rüdiger Klenner

Wer sich schon etwas länger mit seinem 600/700er beschäftigt hat, wird sicher auch schon mal an die Grenzen von Basic gestoßen sein. Zwar lassen sich hier Programme recht rasch erstellen, und solange sie noch klein sind auch leidlich gut übersehen und warten (basteln...) aber so manches läßt sich hier schlecht machen. Eine RAM-Floppy in Basic zu schreiben wäre sowohl unpraktisch wie auch einfach viel zu langsam. Soetwas aber in Assembler zu erstellen ist ziemlich mühsam und zeitraubend. Außerdem sieht es hier mit dem schnellen Umbasteln schon viel schlechter aus. Der Kompromiß, ein in Basic geschriebenes Programm in Maschinensprache zu kompilieren ist auch eine Krücke, denn Basic ist zu maschinenfern und damit das Kompilat im Vergleich zur Lösung in Assembler wesentlich ineffizienter...



FORTH ist ein Betriebssystem und eine Sprache. FORTH ist beliebig (wirklich beliebig) erweiterbar, flexibel (von der Bit-Manipulation an einem Ausgabeport bis zur kommerziellen Anwendung) und es ist sehr kompakt. Eine Seite Forth-Programm entspricht einem mehrfachen an Basic oder Assemblercode. Forth ist schnell, schneller als Pascal-Kompilate und manchmal so schnell wie Assemblercode, da in Forth wesentlich effizienter als in Maschinencode programmiert werden kann. Forth ist auch eine Entwicklungssprache, in der sie ein Programm bereits vor der Implementation entwerfen (!). Dabei ist Forth eine Interpretersprache, sie können also Programme oder Programmfragmente unmittelbar austesten. Forth kompiliert neue Worte und führt dann den kompilierten Code aus. Forth verkörpert auch viele Prinzipien, die in modernsten Sprachen eingebaut sind, wie etwa strukturiertes Design, Modularität, Verstecken von Informationen... Forth hat deshalb Ähnlichkeit mit C, Modula 2, Lisp.

Außerdem ist Forth recht einfach (?) zu implementieren und deshalb habe ich vor einiger Zeit damit angefangen, Forth für unsere Rechner verfügbar zu machen. Vor ein paar Tagen habe ich nun eine Vorversion zum Laufen gebracht, die sich bei den ersten Tests als wirklich ziemlich schnell erwiesen hat. Es bleiben aber doch noch einige Arbeiten zu tun, so z.B. die Anpassung der Diskettenschnittstelle, und auch ein Editor und der fest eingebaute Assembler fehlen noch. So wird es noch etwas dauern, bis Forth in brauchbarer Form verfügbar sein wird.

In der Dezemberausgabe von PASCAL INTERNATIONAL begann eine Serie über FORTH. Damit die Leser die Beispiele auch selbst nachvollziehen können, war der Ausgabe ein Basic-FORTH-Interpreter beigelegt. Natürlich ist dieser entsetzlich langsam und funktioniert nur im Direktmodus, aber zum Lernen von Forth ist er ganz nützlich. Ich habe ihn deshalb abgetippt und etwas erweitert und verbessert und füge ihn diesem Artikel bei, so daß er demnächst für alle Interessierten verfügbar ist.

Eine wirklich gute Einführung in Forth bietet:  
Leo Brodie, Programmieren in Forth, München Wien 1984;  
Außerdem:  
Loe Brodie, In Forth denken, München Wien 1986;  
A.Goppolt/R.Bouteiller, Forth: Ein Programmiersystem ohne Grenzen, Moers 1985.

### Kalk Result - Ein Weichwaren-Rückblick

Jörg Linnig

Dies Programm ist nicht, wie der Name es vielleicht vermuten ließe, das Resultat einer fortgeschrittenen Verkalkung und führt auch nicht zu einer solchen. Um es genau zu sagen, es handelt sich hierbei um eine professionelle Tabellenkalkulation.

Vorauschickend muß ich anmerken, daß ich bis jetzt noch mit keiner anderen Tabellenkalkulation gearbeitet habe und von daher das Programm sicherlich nicht vergleichend beurteilen kann. Von CBM-Anwendern, die schon mit anderen Tabellenkalkulationen (z.B. Visicalc) gearbeitet haben, habe ich mir sagen lassen, daß Calc Result das beste sei, was es an Tabellenkalkulation für CBM Rechner gibt.

Besonders gefiel mir an diesem Programm die Vielsprachigkeit. In einem Menu kann man die gewünscht Sprache (z.B. 0=Deutsch) anwählen. Danach kommen alle Help-Texte in deutscher Sprache. Insgesamt sind acht Sprachen integriert.

Windows sowie ein Multiuserbetrieb sind ebenso möglich wie das Laden von Visicalc-Files oder eine grafische Darstellung (Balkendiagramm) der Tabellen. Einem User stehen bis zu 32 Seiten mit jeweils bis zu 254 Zeilen und 63 Spalten zur Verfügung. Die Ergebnisse der einzelnen Seiten lassen sich auf die jeweils nächste übertragen. Ich glaube Platzprobleme dürften hier eigentlich nicht mehr auftreten.

Was ich allerdings vermisste ist die Möglichkeit, auch von einer Seite auf eine andere zu kopieren. So lassen sich nur komplette Seiten kopieren. Innerhalb einer Seite sind dem Kopieren, Duplizieren und Verschieben fast keine Grenzen gesetzt. Es sei denn, man kennt sich nicht damit aus. Es bedarf schon etwas Übung, bis man hier den Durchblick hat.

Hier muß das englische Handbuch doch ziemlich oft bemüht werden. Das Handbuch ist im Übrigen ganz brauchbar. Alle Funktionen werden zuerst kurz erläutert und anschließend mit Übungsbeispielen belegt. Wer dem Englischen nicht so mächtig ist: Es gibt auch ein deutsches Handbuch (nicht verzagen, Gerd fragen).

So konsequent Calc Result im Großen und Ganzen auch programmiert ist (dies ist nicht immer selbstverständlich - siehe Protext!), so hat auch dieses Programm einige unliebsamen Eigenarten. Wenn man Calc Result mit einem Einzellaufwerk betreiben will, so geht das nur, indem man so tut, als hätte man eine Festplatte angeschlossen

Will man den CBM 8028 anschließen, und hält man sich dabei an das Handbuch bzw. an die Bildschirmausgabe (diese sind schon widersprüchlich), so wird man vergeblich darauf warten, irgendwelche Zeichen zu Papier zu bringen. Vier verschiedene Druckeranpassungen stehen laut Handbuch zur Verfügung. Der CBM 8028 soll angeblich in dem Druckertyp 4 enthalten sein. Es tut sich nur nichts. Die eigentliche nicht vorhandene Nummer 5 führt jedoch zum Ziel. Allerdings auch nur zum Teil, denn bei 'Hardcopy' wird zwar alles ordnungsgemäß zu Papier gebracht, aber im regulären Druckmodus sind auf einmal die Umlaute verschwunden. Hier habe ich bis jetzt noch keine Lösung gefunden.

Das Sichern der Daten auf Diskette ist auch ein Kapitel für sich. Im Prinzip ist es nicht notwendig, die Seiten abzuspeichern, da beim Verlassen des Programms alle Änderungen abge-

speichert werden. Beim Starten des Programms wird automatisch der ursprüngliche Zustand wieder hergestellt. Sogar der Cursor steht an derselben Stelle. Abspeichern ist jedoch dann sinnvoll, wenn man eine Tabelle ändern, den alten Zustand aber noch für später sichern möchte. Neben dem normalen Format ist es auch möglich, in dem sogenannten DIF-Format Tabellen abzuspeichern und wieder zu laden. Das Diskettenhandling ist im allgemeinen sehr sicher. Mir sind, selbst nach Bedienungsfehlern, noch keine Tabellen unwiderruflich verloren gegangen. Allerdings wird diese Sicherheit auch bezahlt - und zwar mit Zugriffszeiten. So ergibt sich da so manche Kaffeepause.

Was ich an diesem Programm sehr vermisse ist die Möglichkeit, unterschiedliche Spaltenbreiten zu definieren. Die definierte Spaltenbreite gilt immer für alle Spalten gemeinsam. Das größte Manko dieses Programmes ist aus meiner Sicht, daß es unmöglich ist, Felder, die NA (not avalible) oder ERROR sind nicht abzufangen zu können. So kann man zwar mit vielen Tricks auch iterative Probleme lösen, aber eine falsche Eingabe kann die gesamte Tabelle zerstören. Diese "Kleinigkeit" schmälert den Wert dieses Programmes aus meiner Sicht erheblich.

Ebenso ist die Ausgabe auf den Drucker für meinen Geschmack etwas zu kompliziert, bzw. erfordert eine zu lange Tastenfolge. Hier wäre es vielleicht auch angebracht gewesen, die Möglichkeit vorzusehen, Tastenfolgen in Form von Macros abzulegen.

Die Verfügbarkeit von mathematischen Funktionen ist dafür wieder vorbildlich. Das Programm besitzt einen eigenen Formel-Interpreter, der bei einer fehlerhaften Eingabe dies sofort reklamiert. Überflüssige Klammern werden automatisch eliminiert. Neben den normalen mathematischen Funktionen gibt es Funktionen wie Prozent, Summe, Anzahl, Mittelwert, Minimal- und Maximalwert. Außerdem sind if..then..else - Verzweigungen möglich.

Zwei bis jetzt noch nicht erwähnte Leistungsmerkmale sind die Möglichkeiten, den Inhalt von Spalten in ansteigender oder abfallender Reihenfolge alphabetisch oder numerisch sortieren zu lassen, sowie der Kommunikations-Modus, bei dem mit anderen Computer Daten auszutauscht werden können.

Ich bin der Meinung das Calc Result eine sehr brauchbare Tabellenkalkulation mit einem sehr großen Leistungsumfang ist. Wünsche bleiben sicherlich offen, aber welches Programm ist schon perfekt. Vielleicht gibt es ja noch den einen oder anderen Trick.

Ich würde mich freuen, wenn sich auch andere Anwender kritisch oder ergänzend zu Wort melden würden.

## Textverarbeitungen

Gerd Schumacher

An dieser Stelle möchte ich lediglich einen kurzen Überblick über die für unsere Rechner existierenden Textverarbeitungen geben.

Am weitesten dürfte Profitext verbreitet sein. Es ist recht leicht zu handhaben. Dem Anspruch, ein professionelles Textsystem darzustellen, wird es aber keineswegs gerecht. Der gegenwärtige Ausverkaufspreis von ca 100,- ist wohl noch gerechtfertigt, zumal in Grenzen (nur ein Gerät gleichzeitig) die Benutzung der seriellen CBM-Schnittstelle unterstützt wird. Mehr oder minder handelt es sich aber bloß um eine Art Briefschreibprogramm.

Meines Erachtens ist man dafür mit TEXT von Günther Brühl wesentlich besser bedient. Es ist die einzige Textverarbeitung, die den hervorragenden eingebauten Editor unserer Geräte benutzt. Die Editorfunktionen sind sogar sinnvoll erweitert. Statt Anleitungen zu wälzen, kann man hier einfach so drauflos tippen, wie man es vom Basic oder Monitor her gewöhnt ist. Das ist ein nicht zu unterschätzender Vorzug. Das Programm ist beim Autor zu DM 50,- zu haben. Zwei Nachteile von TEXT seien nicht verschwiegen: Es benötigt unbedingt das Basic 256k, und das Speicherformat ist nicht universell. Dennoch, als einfaches Texteingabeprogramm für umgewöhnungsfaule ist es hervorragend. Wenn Sie es haben, werden Sie von Profitext gewiß nichts mehr wissen wollen.

Unser Profitext-Experte Rolf Hustedt hat gleich zwei mal zugeschlagen. Zunächst gibt es von ihm ein umsonstenes Texteingabeprogramm auf unserer Clubdiskette "Basic-Utilities". Dieses Programm hat das gleiche Speicherformat wie Profitext, und eignet sich vornehmlich für Diskettenbriefe, die mit einem winzigen Leseprogramm zugänglich werden - und, wie gesagt - auch Profitexte. Ausdrucken kann man immer zwei Bildschirmseiten auf ein Blatt.

Rolf Hustedts Programm "Schreibmaschine" ist da schon wesentlich leistungsfähiger. Es läuft auf allen 600/700, hat einige Möglichkeiten zur Peripherieanpassung, und hat die selbe Tastenbelegung für deutsche Sonderzeichen wie Profitext, falls man mit dem ASCII-Kernal arbeitet. Profitext ist es zumal wegen seiner größeren Flexibilität mindestens ebenbürtig. Es ist als Clubdiskette mit einer Beschreibung zu DM25,- zu haben. Übrigens steht der Autor beim Auftreten irgendwelcher Probleme mit Rat und Tat zur Seite.

Auf der Clubdiskette "Ernsthafte Anwendungen" finden sich zwei weitere Textprogramme in Basic, die ich aber noch nicht ausprobiert habe.

Ich weiß nicht, ob es ein großer Verlust ist, eine der ganz alten Textsysteme für CBM-Rechner, Comtext bzw. Wordcraft ist anscheinend nicht mehr verfügbar. Es hat einmal eine englische Version gegeben. Comtext ist vollständig menuegeführt,

was gewiß auch seine Lästigkeiten beinhaltet, und es ist die einzige CBM-Textverarbeitung, die Blocksatz mit Proportional-schrift erlaubt. Vom äußeren Erscheinungsbild her ist auch die Bildschirmausgabe die angenehmste. Formatierungsangaben werden sofort ausgeführt, und Steuerzeichen erscheinen nicht im Text, sondern lediglich der erste Buchstabe, bei dem ein Steuerzeichen wirksam wird, wird invers dargestellt. Die jeweiligen Steuerzeichen können in einer separaten Statuszeile betrachtet werden. Irgendwie ist das schon toll, wenn beispielsweise Text zentrieren eingegeben wird, und dann der Cursor sofort in die Bildschirmmitte springt, und der eingegebene Text gleich bei der Eingabe symmetrisch auf dem Bildschirm erscheint.

Leider ist Comtext insgesamt wesentlich weniger leistungsfähig als die anderen professionellen Textsysteme für CBM. Die eigentlich ganz tollen Ideen, die hier verwirklicht sind, werden mit einem meines Erachtens zu hohen Preis, einer teilweise unerträglichen Lahmheit bezahlt. Für das moderne "What you see is what you get" reicht die Rechenleistung einfach nicht aus. Es können beim Einfügen in längere Texte Sekunden pro Zeichen vergehen! Gleichwohl würde ich mich kümmern, wenn eine nennenswerte Zahl von Mitgliedern an Comtext interessiert wäre.

Das weit verbreitete Wordpro ist auf unseren 600/700 nicht lauffähig. Es ist zwar schon ein altes Schätzchen, trägt aber alle wesentlichen Merkmale einer modernen Textverarbeitung. Dennoch wäre eine Anpassung an unsere Rechner flüssiger als flüssig, nämlich überflüssig, da wir etwas Besseres haben.

Superscript, beim C-64 auch als Easyscript bekannt, ist in der deutschen Fassung nur als Bestandteil von Superoffice, also mit einer hervorragenden Datenbank als integriertes Paket erhältlich gewesen. Ich habe noch keinen ganz genauen Überblick, aber es ist weitestgehend befehlskompatibel zu Wordpro. Im Vergleich zu Protext kann man sagen, daß die Befehlsstruktur erheblich leichter zu durchschauen und zu behalten ist. Superscript-Texte werden als normale sequentielle Files abgespeichert - normalerweise natürlich mit Steuerzeichen -, wodurch es problemlos möglich ist, andere sequentielle Files in die Textverarbeitung einzuziehen und hier zu bearbeiten. Die englische Version beinhaltet eine rasante Textkorrektur mit drei Wörterbüchern, englisch, amerikanisch und 'mid-Atlantic'. Die interessanteste Möglichkeit ist natürlich die der Kombination mit Superbase. Das einzig wirkliche Übel bei Superscript ist, daß Silbentrennungen lediglich nach explizit vorgegebenen Vorschlägen vorgenommen werden können. Die Ausdrücke beleidigen gelegentlich das Auge eines Ästheten zutiefst, zumal wir in der deutschen Sprache so entsetzlich lange Wörter haben. Ich hoffe, daß unsere Gemeinnützigkeit uns bei dem Bemühen, das Programmpaket Superoffice (deutsche Version) unseren Mitgliedern zugänglich zu machen, helfen wird. Die englische Version ist leider nur begrenzt einsetzbar. Superoffice gibt es übrigens auch für den C 128. Wie aus berufenem Munde zu hören ist, ist die Qualität dieser Version gegenüber der Version 600/700 erheblich schlechter.

Bei allen Unzulänglichkeiten, die dem Programm anhaften, muß man sagen, daß Protex mit Abstand die beste Textverarbeitung ist, die für die ganze PET-Sippschaft existiert, zumindest, wenn man mehr will, als einfach nur Briefe schreiben. Diese Zeitung beispielsweise wird seit Heft 1 damit erstellt. Für das Druckergebnis ist es beispielsweise sehr angenehm, daß beim Blocksatz nicht einfach wie bei Profitext und Superscript weitere Leerzeichen eingefügt werden, sondern daß der zusätzlich benötigte Zwischenraum auf die Zwischenräume zwischen allen Wörtern einer Zeile gleichmäßig verteilt wird. Die Programmierung von Druckertreibern ist verhältnismäßig leicht, da hierfür ein eigener Editor vorgesehen ist. Es läßt sich praktisch jeder Drucker ansteuern. Mir ist zwar noch nie zu Ohren gekommen, daß es jemand geschafft hat, Protex zum Absturz zu bringen, aber so ganz fehlerfrei ist es nicht. Bei den Fehlern aber handelt es sich lediglich um kleine Lästigkeiten, wie beispielsweise, daß man nach dem Einfügen einer Zeile nicht auch ein Zeichen einfügen oder löschen kann, ehe man nicht irgendein anderes Zeichen eingegeben hat. Solcherlei Ungereimtheiten treten gelegentlich auf, und können ganz schön nerven. Gewiß ist Protex von den hier erwähnten Programmen das gewöhnungsbedürftigste, da nach Meinung vieler Anwender die Logik des Befehlsaufbaus schwer zu ergründen ist. Wenn man es einmal gewöhnt ist, fällt so etwas aber nicht mehr auf. Auch mit den kleinen Macken kann man leben. Eine Macke, die ich sehr unangenehm finde, ist die abschließend automatische Silbentrennung, die durchschnittlich einen Fehler pro Seite produziert, aber besser als gar keine. In dieser Beziehung halte ich die Profitext-Lösung für die beste, wo jede Trennung erfragt wird. Selbstverständlich beinhaltet Protex eine lernfähige Textkorrektur. Die Funktionstasten sind mit Phrasen und Macros programmierbar. Wie Superscript und Comtext enthält es Rechenmöglichkeiten und die Möglichkeit zum Dateneintrag beispielsweise für Serienbriefe. Als Besonderheit ist noch eine eingebaute Terminalfunktion zu erwähnen.

Ich habe für mein Protex noch DM 300,- berappen müssen. Jetzt sind wir in der Lage, es zu DM 70,- einschließlich Handbuch anzubieten. Endlich ein zeitgemäßer Preis, zu dem vermutlich nichts Vergleichbares auch für andere Rechner zu haben ist.

Gerd Schumacher

Der Enthusiasmus einiger Mitglieder hat mich einst bewogen, Macro-Basic anzuschaffen. Wie schon früher erwähnt, hat es mich auf Anhieb so sehr überzeugt, daß ich mich veranlaßt sah, auch Ihnen den Gebrauch dieses Programmpakets wärmstens ans Herz zu legen. Ich habe einige herbe Kritik dafür einstecken müssen, eine solche "kostenlose Werbung" getrieben zu haben.

Nun kommt der Autor uns in der Form entgegen, daß wir eine Club-Ausgabe von Macro-Basic zum halben Preis vertreiben dürfen. Wer dieses Angebot für so kommerziell hält, daß es hier nicht weiter behandelt werden sollte, möge einfach weiterblättern. Da ich zum allgemeinen Nutzen an guten Programmen für unsere Bibliothek interessiert bin, halte ich es für angemessen, an dieser Stelle ein wenig auf Macro-Basic einzugehen. Eine umfassende Würdigung der über 200 Befehle würde Bände füllen. So beschränke ich mich hier darauf, exemplarisch die Befehle aufzuzählen, die den Programmablauf festlegen. Besonders diese Befehle haben nämlich unmittelbaren Einfluß auf die erzielbare Programmqualität.

Es können Labels vergeben werden. Das sind Markierungen bzw. Namen von Sprungzielen oder Datazeilen. Diese Labels können dann statt Zeilennummern angegeben werden:

```
goto ("label") gosub ("label") und restore ("label")
```

Auf diese Weise wird nicht nur die Lesbarkeit und Übersichtlichkeit erhöht, auch die Programmerstellung wird wesentlich erleichtert.

Darüber hinaus ist es möglich, Sprungziele durch Variablen festzulegen oder zu errechnen z.B.:

```
goto oder gosub (a/b+10)
```

Für "if - then - else"-Strukturen stehen im Fall der Erfüllung einer Bedingung beliebig viele Programmzeilen zur Verfügung. Im normal-Basic ist man auf den Rest der Zeile beschränkt, und wenn dieser nicht ausreicht, ist hier ein Sprungbefehl notwendig. Das Ende des Zweiges "Bedingung erfüllt" wird durch "endif" markiert.

Außer den üblichen "for - next"-Strukturen bietet Macro-Basic drei weitere Schleifenarten:

"do" eröffnet eine Schleife, die nur durch "enddo" verlassen werden kann. "do until" markiert den Anfang einer Schleife (bis "enddo"), die solange durchlaufen wird, bis eine Bedingung erfüllt ist. "do while" - "enddo" wird solange durchlaufen wie eine Bedingung erfüllt ist.

Eleganter und Übersichtlicher als "if-then-else"-Strukturen ist oftmals die Macro-Basic-Struktur "select-case-other-

ends", die in recht knapper Schreibweise die Überprüfung mehrerer Bedingungen mit einem Befehl zuläßt.

Unter allen hier aufzuführenden Beispielen dürfte das folgende Sprachelement das mächtigste sein. Es ist möglich, bestehende Unterprogramme ohne Zeilennummern- oder Variablennamen-Anpassung in ein Programm einzufügen oder daran anzuhängen. Diese Unterprogramme können vom Hauptprogramm nur durch den Befehl "call(label)" aufgerufen werden. Variablen und Tabellen, die von Haupt- und Unterprogramm gleichermaßen gebraucht werden, müssen dem Unterprogramm explizit übergeben werden. Vom Unterprogramm selbst angelegte Variablen gehen beim Rücksprung ins Hauptprogramm mittels "returns" oder "endp" verloren. Es sind also wie in Fortran oder Pascal echte lokale Variablen möglich. Alle, die es mir am Telefon nicht glauben wollten, haben es jetzt endlich schwarz auf weiß.

Nicht zuletzt gibt es einige Möglichkeiten, die auch so schön gewonnenen Programmstrukturen zu zerdeppern. Die keiner Erklärung bedürftigen Befehle heißen: "exit select", "exit if", "exit for", "exit do", "exit next" und "exit return". Hier wird beim vorzeitigen Verlassen der in den Befehlen benannten Strukturen der Stapelzeiger bereinigt.

Programme nachladen kann man mit den Befehlen "dlink", womit ein Programm an ein bestehendes angehängt wird, und zum Mischen von Programmen nach Zeilennummern mit "dmerge". Wenn ein Programm mit "overlay" geladen wird, werden sämtliche bis dahin bestehenden Variablen gelöscht. Vermittels "submit", "next pgm", "end pgm" und "end sub" kann man den Computer veranlassen, mehrere voneinander unabhängige Programme automatisch nacheinander abzuarbeiten. Variable werden hierbei nicht übernommen.

Nichts für Puristen in Sachen strukturierter Programmierung ist die Möglichkeit, sich selbst verändernde Programme zu schreiben. Allerdings erscheint es sinnvoll, Programme, die etliche Verzweigungen entsprechend den diversen Basics und Kernals benötigen, gleich beim Programmstart nur ein mal automatisch anzupassen, statt immer wieder die selben Abfragen vorzunehmen.

Diese nicht ganz vollständige Kurzübersicht über einige thematisch ausgewählte Macro-Basic-Befehle möge genügen, Ihnen eine Vorstellung von Macro-Basics Leistungsfähigkeit zu vermitteln. Möge sie angetan sein, bei meinen Kritikern Verständnis zu wecken!

Man kann sich aus Macro-Basic individuelle Befehlserweiterungen zusammenstellen. Wenn eine solche Zusammenstellung genau die Befehle enthält, die ein Programm benötigt, darf ein solches Modul auch weitergegeben werden, ohne daß man mit dem Urheberrecht in Konflikt gerät. Vorläufig aber ist nur eine komplette Befehlserweiterung über den Verein zu beziehen. Das Programm zur Erstellung individueller Erweiterungen wird dann in kürze unaufgefordert und kostenlos nachgeliefert.



Nachdem ich im Frühjahr vergangenen Jahres mich schon Übermäßig (drei Monate) mit der Korrektur des Betriebssystems m98c und der Erstellung eines ausführlichen Handbuches herumgeschlagen habe, glaubte ich eigentlich von dem Chaos, das Commodore mit dem Druckerbetriebssystem produziert hatte, die Schnautze voll zu haben. Aber das Anliegen vieler Benutzer, den Drucker auch mit Centronics betrieben zu wollen, veranlaßte mich, noch einmal in die Niederungen des selben hinabzusteigen.

Mit folgenden Utensilien habe ich meine Arbeit begonnen:

1. Das von mir mäßig dokumentierte Disassembler-Listing des Betriebssystems m98c;
2. ein lang ersehntes Centronics-Betriebssystem m19a, das mir ein freundlicher Benutzer nebst Schaltungsunterlagen zur Verfügung gestellt hat;
3. der Schaltplan des Druckers;
4. ein Reassembler (um aus dem Programmcode ein Source-File für den Assembler herzustellen);
5. viele Ideen und Vorschläge, was der Drucker alles können sollte.

Eigentlich hatte ich gehofft, innerhalb von vier Wochen alle neuen Befehle (siehe nächste Seite) zu implementieren, aber es hat leider vier Monate gedauert, wohlgerne täglich mehrstündiger Arbeit. Als größtes Handikap hat sich bei meiner Arbeit der Stil erwiesen, in dem das Betriebssystem ursprünglich geschrieben wurde. Ein so schlechtes Programm habe ich wirklich noch nie gesehen. Nicht nur, daß der Programmierer im Fünfeck gedacht hat, es war ihm anscheinend auch nicht möglich, seinen eigenen Programmwust noch korrekt zu beherrschen. Ein Ergebnis davon waren die vielen Fehler, die ich beseitigen wollte. Was aber viel schlimmer war: Wenn ich irgendeine Veränderung programmiert hatte lief erstmal überhaupt nichts mehr.

Dem genannte Umstand ist es zu verdanken, daß ich manche Routinen, nachdem ich endlich verstanden hatte, was sie sollten, komplett neu geschrieben habe. Ich glaube, daß ich im Endeffekt schneller gewesen wäre, wenn ich von vornherein das meiste neu geschrieben hätte. Aber hinterher ist man ja immer klüger.

Zu guter Letzt litt meine Arbeit unter chronischem Platzmangel und ich mußte mir mehrfach Platz durch das Verkürzen von Routinen machen, was eine sehr fehleranfällige Prozedur ist.

Daß ich nicht aufgegeben habe lag einzig und alleine daran, daß ich auf das bißchen Geld, das ich mit dem neuen m101 zu verdienen hoffte, dringend angewiesen war und bin.

Nun aber zu dem angenehmeren Teil meiner Arbeit, dem Resultat:

Es gibt jetzt einen einheitlichen Source-Text, in dem über Parameter einstellbar ist, wie groß der Speicher ist, und ob die IEEE oder Centronics-Schnittstelle gebraucht wird.

Folgende Fehlerkorrekturen und Ergänzungen sind im neuen m101 zu finden:

- Der Walzenvorschub bei Terminatoren (return, line-/form-feed...) wird direkt ausgeführt, nicht erst beim Druckbeginn der nächsten Zeile. Besonders für den formfeed-Befehl ist das ungemein wichtig.
- Alle 1-byte-Befehle werden auch beim Rückwärtsdruck richtig bearbeitet.
- Die linefeed-Taste hat jetzt eine Dauerfunktion.
- Der Überlauf bei Walzenrückbewegungen ist verkleinert, jetzt sind auch in der obersten Zeile die Halbzeilen-Befehle zu gebrauchen.
- Der Zeichenbuffer kann nicht mehr überlaufen.
- Beliebige Benutzung der Zeichenabstandsbeefehle (Umschalten von 12 auf 15 Zeichen/Zoll ist jetzt direkt möglich).

Folgende neue Befehle machen den Drucker zu einem ernstzunehmenden Gerät:

- Seitenrückschub (geht an den Anfang einer Seite).
- Zeilenvor-/rückschub um 1/144 Zoll.
- Auf Blatteinzug warten (ein großer Druckerpuffer ist jetzt auch bei Einzelblattverarbeitung voll nutzbar).
- Papier-Kontrolle ein-/ausschalten.
- Zeile immer nach rechts drucken.
- Zeilenbuffer ausdrucken.
- Randverschiebung (z.B. für Lochrand bei Endlospapier).
- Fettschrift ein-/ausschalten und dessen Stärke einstellen.
- Einfaches Nachladen von Zeichensatz und Proportional-schrifttabelle.
- Farbbandvorschub ein-/ausschalten.
- Nullen durchstreichen ( $\emptyset$ ), macht Listings besser lesbar.
- Leerzeichenmodus (zur vereinfachten Benutzung mehrerer Typenräder auf einer Seite).
- Rollen abheben (zum Bedrucken und Verschieben der obersten Zeilen eines Blattes).
- Druckverlangsamung (zum noch präziseren Drucken).
- IEEE-Geräteadresse einstellen.
- Alle Befehle sind als Escape-Sequenzen realisiert.

Mit am interessantesten ist aber wohl die Implementation der Centronics-Schnittstelle. Mit einer kleinen Hardware-Erweiterung (ein Platinchen mit einem Treiberbaustein und einem Schalter) und der Centronics-Version vom m101 läßt sich der 8028 an Computer mit Centronics-Schnittstelle anschließen. Der Befehlssatz ist umschaltbar zwischen Commodore- und EPSON-Modus, letzteres wohl inzwischen ein Standard unter den Druckertreibern.

Das ganze wird durch ein neu überarbeitetes Handbuch (ca. 150 Seiten) abgerundet. Nun, ich hoffe, daß das neue Betriebssystem ein bißchen Anklang findet. Es wird über den Verein vertrieben (siehe Bestellschein).

## Flohmarkt

Mit dieser Rubrik gibt es das Problem, daß hier kaum eine ausreichende Aktualität herzustellen ist. Ingo Lerch (siehe Adressen, Seite 7) wird aktuelle Angebote und Anfragen sammeln.

Nachzutragen ist noch, daß Wolfgang Abends, Föhrenweg 8, 5068 Overath 3, den Austausch von Einzelteilen zu bewerkstelligen sucht, die eventuell auf Verdacht angeschafft worden sind.

Wir suchen beispielsweise dringend Analogplatinen 8250 und 8250 LP sowie 2 Digitalplatinen 8050/8250 HP.

Zwei Angebote, von denen ich weiß, daß sie noch aktuell sind:

1541, neu bei Hauke Lengsfeld, Parkstr. 3, 2250 Husum

Komplette 600er-Anlage bei Ferdinand Beutel, Hauptstr. 64, 5014 Kerpen 1

#### Vorschau auf Heft 4

Wir wollen nicht zu viel versprechen, aber die Chancen für ein baldiges Erscheinen des nächsten Heftes stehen gut. Hier liegt schon einiges an Artikeln herum, das nicht mehr in dieses Heft hineinpaßt. Wir bitten die Autoren um Verständnis. Die bisherige Desorganisation neigt sich erfreulich dem Ende zu, und allein die Vorstellung neu aufzunehmender Software von unseren amerikanischen Freunden füllt mindestens ein halbes Heft. Wer hätte gedacht, daß wir uns einmal vor Programmen kaum retten können würden!

Weitere Themen:

Centronics, gleich zwei Lösungen  
CBM-Floppies, neue Serie  
Hardwarebeschreibung 600/700, Schaltplan Seiten 1 und 2  
Was wir von CBM-Deutschland zu erwarten haben  
Projektberichte  
Rezensionen

#### Impressum

Dieses ist das letzte von vier Heften, die für DM 20.- Selbstkostenpreis zu haben sind. Folgende Hefte werden im Rahmen der CBM-Selbsthilfe weitergegeben, und sind im Mitgliedsbeitrag enthalten.

Redaktion:  
Gerd Schumacher, Andreas Stankewitz.

Artikel, die nicht namentlich gekennzeichnet sind, stammen von der Redaktion. Verantwortlich i.S.d.P. für die Artikel sind die jeweiligen Autoren.

Herstellung und Vertrieb:  
CBM-Selbsthilfe e.V. (i.G.),  
Lindenstr. 15 Hinterhaus,  
5000 Köln 1.



wächst und gedeiht. Wie bereits im Vorwort erwähnt, habe ich (Gerd Schumacher) hier eine ganze Menge Energie rein gesteckt. Ein sinnvoller Ansatz zur Arbeitsteilung hat sich hier schon bewährt, indem die von mir grob sortierten Disketten von anderen weiterbearbeitet wurden. Diese Leute sind auf den so zustande gekommenen Disketten benannt und stehen als Ansprechpartner für Kritik und Anregungen bezüglich dieser jeweils einen Diskette zur Verfügung. Unmöglich können sämtliche Programme von einer Diskette unter allen möglichen Bedingungen voll ausgetestet werden (DIN, ASCII, 128k, 256k). Oft sind mehrere Versionen von einem Programm zu finden, wo mit vertretbarem Aufwand eine genaue Überprüfung der Vor- und Nachteile der einzelnen Versionen nicht festzustellen waren. Alle Mitbenutzer sind aufgerufen, sich an der Endkontrolle der Sammelurien zu beteiligen, damit gelegentlich überarbeitete Versionen erstellt werden können.

Nach bestem Wissen sind die als public domain gekennzeichneten Disketten frei von urheberrechtlich geschützten Programmen. Sollte uns hier wider Erwarten ein Fehler unterlaufen sein, sollten Sie nichts Eiligeres zu tun haben, als uns Bescheid zu geben, damit der Fehler korrigiert und Schaden von den Autoren und dem Verein abgewendet werden kann.

In der Vergangenheit haben wir viele randvolle Disketten zur Auswertung bekommen, wo kaum eine Silbe Erläuterungen zu finden war. Es ist grausam, sich nachher mit so etwas herumzuschlagen. Jörg Linnig hat ein Programm erstellt, "Dir.Doc!", das aus einem Directory ein Basic-Programm macht, das es gestattet, Kommentare an die Directory-Einträge anzufügen. Läßt man das Programm laufen, wird das kommentierte Directory entweder auf Bildschirm oder Drucker ausgegeben oder in eine sequentielle Datei umgewandelt. Für künftige umfangreichere Einsendungen bitte ich, davon regen Gebrauch zu machen. Zweierlei Mitarbeit tut noch Not: 1. Wir brauchen Programme, möglichst auf Ihrem eigenen Mist gewachsene. 2. Die zu erwartenden Programme müssen zu einigermaßen geordneten Disketten zusammengestellt werden.

### Urheberrechtlich geschützte Software

Den unter diese Rubrik fallenden Disketten sind in diesem Heft schon einige Worte gewidmet. Hier sind zwei erfreuliche Zugänge zu verzeichnen, nämlich das ehemals arg teure Protext und das einfache Texteingabeprogramm Schreibmaschine von Rolf Hustedt. Die Preise sind so kalkuliert, daß für den Verein etwa so viel hängenbleibt wie bei den Disketten mit freier Software. Für die Aufnahme solcher Programme in unsere Bibliothek achten wir strikt auf ein Preis-Leistungs-Verhältnis, das unseren Ausverkaufs-Computern angemessen ist. Wollen wir weiter beim Erwerb der Vertriebsrechte an kommerziellen Programmen Erfolg haben, ist es unbedingt erforderlich, daß die CBM-Selbsthilfe über jeden Verdacht erhaben ist, daß solche Software auf ungesetzliche Art weiterverbreitet wird.

## CBUG-Software

Unsere amerikanischen Mitbenutzer haben mittlerweile eine Bibliothek von insgesamt etwa 100 Disketten, größten Teils freie Software. Die abgabepflichtigen Programme stehen uns zu den selben Bedingungen zur Verfügung wie CBUG. Im Moment gehen wir von einem Dollarkurs von DM 1,80 aus, und von einem Bißchen Auf und Ab lassen wir uns nicht beirren.

Derzeit ist erst eine begrenzte Anzahl von CBUG-Disketten zu haben. Gewiß wird sich dieser Zustand sehr bald ändern. Ein Teil der Disketten kann nur bezogen werden, wenn zusätzliche Erklärungen abgegeben werden. Ich bitte um Ihr Verständnis und um die unbedingte Einhaltung eventueller Nutzungs- und Weitergabebeschränkungen. Mißbrauch der Materialien kann arge Unannehmlichkeiten nach sich ziehen.

Mögen sich Mitglieder finden, die wichtige Teile der amerikanischen Software auch unseren der englischen Sprache weniger mächtigen Mitgliedern zugänglich machen, zunächst wären deutsch dokumentierte Directories ein wichtiger kleiner Schritt.

## Elektro (-nik) und Funk

Diese unsere erste Clubdiskette dürfte auch eine der nützlichsten sein. Wir warten seit ungefähr einem halben Jahr auf eine überarbeitete und erweiterte Version.

Bedauerlich ist, daß die zahlreichen Funker unter uns unsäglich schreibfaul sind. Dabei hätten sie gewiß auch den anderen Mitbenutzern einiges zu vermitteln.

|    |                       |     |      |                    |     |
|----|-----------------------|-----|------|--------------------|-----|
| 0  | "Elektro & Funk 1" ii | 2c  | 44   | "filter berechn"   | prg |
| 2  | "menue shift run"     | prg | 28   | "netzteil 1"       | prg |
| 18 | "aktivfilter"         | prg | 5    | "koax verk.fakt"   | prg |
| 8  | "vierpolfilter"       | prg | 5    | "dx checker"       | prg |
| 5  | "e96"                 | seq | 24   | "grayline"         | prg |
| 34 | "transistor"          | prg | 11   | "ant.-match"       | prg |
| 23 | "netzwerkanalys"      | prg | 19   | "elektronik"       | prg |
| 32 | "rc netzwerke"        | prg | 11   | "morsetrainer"     | prg |
| 9  | "daempfungsglied"     | prg | 8    | "range-bearing"    | prg |
| 20 | "lautsprecher"        | prg | 13   | "cw programm"      | prg |
| 13 | "multiv.ne-555"       | prg | 7    | "morsen 610"       | prg |
| 23 | "transformator1"      | prg | 6    | "qth-kenner"       | prg |
| 36 | "Filterberechn"       | prg | 22   | "morsen mit cbm"   | prg |
| 19 | "elektro ber."        | prg | 17   | "ralaisliste"      | prg |
| 13 | "bandpass-cauer"      | prg | 2    | "ind.widerstand"   | prg |
| 9  | "tongenerator"        | prg | 2    | "kap.widerstand"   | prg |
| 9  | "aktiv-filter"        | prg | 8    | "scheiwiderst."    | prg |
| 6  | "helixfilter"         | prg | 9    | "scheinleiw."      | prg |
| 19 | "aktivfilter741"      | prg | 4    | "resonanz"         | prg |
| 5  | "yagiantenne"         | prg | 7    | "luftspulen"       | prg |
| 9  | "ne 555 timer"        | prg | 6    | "eisendrossel"     | prg |
| 6  | "koax.-kreise"        | prg | 29   | "transformator"    | prg |
| 4  | "filterberechn."      | prg | 18   | "filter"           | prg |
| 5  | "schwingkr.ber."      | prg | 7    | "winkefunkt."      | prg |
| 22 | "transistorstuf"      | prg | 26   | "emittersch.6"     | prg |
| 17 | "netzteilber."        | prg | 31   | "emittersch.7"     | prg |
| 26 | "lautspr.ber"         | prg | 26   | "lautsprecherber." | prg |
| 12 | "daempfungsgl."       | prg | 7    | "locator"          | prg |
| 11 | "durchlasskurve"      | prg | 3316 | blocks free.       |     |

## Basic-Utilities

und andere Kostbarkeiten, fast alle auf dem Mist unserer Mitglieder gewachsen, machen zusammen schon was her. Dennoch, die Diskette ist durchaus noch ausbaufähig. Wer macht mit?

|    |               |                                          |
|----|---------------|------------------------------------------|
| 4  | shift run     | .prg.:die Disk stellt sich vor           |
| 8  | logo          | .scr.prg.:stark verbesserungsbedürftig   |
| 20 | Directory     | .dok :.prg.:dieses hier                  |
| 1  | dokumentatio  | .usr.:wunderhübsche Überschrift          |
| 21 | Dir.Dok !     | .bas .prg.:zur Erzeugung dieser datei    |
| 2  | logodesign    | .bas.prg.:Hilfe zum Bildchen-malen       |
| 2  | ! lesen !.bas | .prg.:Erklär. zu den prg. von r. hustedt |



|    |                  |                                                             |
|----|------------------|-------------------------------------------------------------|
| 5  | Briefaus.bas     | .prg.:Leseprogramm für das folgende prg.                    |
| 38 | Briefein.bas     | .prg.:ganz einfaches Texteingabeprg.<br>profitextkompatibel |
| 8  | Sub02            | .prg.:                                                      |
| 3  | Sub0f            | .prg.:                                                      |
| 3  | Sub15            | .prg.:                                                      |
| 24 | Suble            | .prg.:                                                      |
| 3  | Sublf            | .prg.:                                                      |
| 1  | untersuchung     | .usr.:                                                      |
| 8  | Listing 1b.bas   | .seq.:                                                      |
| 9  | Listing 2b.bas   | .seq.:                                                      |
| 4  | basiclist1       | .prg.:                                                      |
| 5  | basiclist2       | .prg.:                                                      |
| 12 | cross-ref.bas    | .prg.:prg zur Erstellung von variablen-                     |
| 7  | variablenliste   | .prg.:listen und Sprungadressenverzei.                      |
| 7  | xref             | .prg.:                                                      |
| 1  | startadresse     | .prg.:von Programmen auf diskette                           |
| 8  | Di St-Adr.bas    | .prg.:                                                      |
| 2  | zeile/speicher   | .prg.:wo steht eine basiczeile                              |
| 2  | search           | .prg.:suchen                                                |
| 2  | find .bas        | .prg.:und finden                                            |
| 1  | pgmerstellun     | .usr.:Programmerstellung                                    |
| 4  | ren+find-dok     | .prg.:                                                      |
| 5  | ren+find-610     | .prg.:                                                      |
| 6  | renum. Basic 256 | .prg.:umnummerieren                                         |
| 5  | renum            | .prg.:                                                      |
| 6  | renumber         | .prg.:                                                      |
| 4  | renumber.1024    | .prg.:                                                      |
| 1  | mask merge/doc.  | .prg.:Programme verknüpfen                                  |
| 1  | mask merge .doc  | .prg.:                                                      |
| 8  | mask merge v.4.6 | .prg.:                                                      |
| 15 | merge v.4.6      | .prg.:                                                      |
| 11 | basic append     | .prg.:Programme anhängen                                    |
| 3  | auto/zeile-load  | .prg.:automatische Numerierung                              |
| 2  | autonumber.bas   | .prg.:bei der Eingabe                                       |
| 1  | auto 1           | .prg.:                                                      |
| 2  | auto             | .prg.:                                                      |
| 6  | menuegestaltung  | .prg.:siehe auch hsg?! nr.3                                 |
| 5  | hauptpgm         | .prg.:                                                      |
| 2  | unterpgm         | .prg.:                                                      |
| 2  | basic-comm       | .prg.:                                                      |
| 1  | B-listschutz     | .prg.:hoffnungsloser versuch                                |
| 4  | relocate         | .prg.:                                                      |
| 6  | var-n-disk       | .prg.:schnelles variablensichern                            |
| 1  | hex              | .usr.:                                                      |
| 6  | datawandler      | .prg.:macht aus Maschinenprg. datazeilen                    |
| 6  | data wandler     | .prg.:                                                      |
| 5  | U-hex-dump       | .prg.:Speicherlisting in hex                                |
| 5  | hex-dump         | .prg.:                                                      |
| 7  | U-hex-eingabe    | .prg.:                                                      |
| 7  | hex-eingabe.bas  | .prg.:                                                      |
| 11 | hex-eing./quers  | .prg.:                                                      |
| 1  | DATUM            | .usr.:wenn man in prg. das datum braucht                    |
| 3  | pgmdat           | .prg.:                                                      |
| 9  | predat           | .prg.:                                                      |
| 4  | chedat           | .prg.:                                                      |
| 1  | datum            | .seq.:                                                      |
| 1  | kompression      | .usr.:Basicprogramme werden verkürzt                        |

```

10 compacteur .prg.:und damit schneller
10 compacteur/a .prg.:
21 compactor .prg.:
24 compactor basic .prg.:
8 compactorml.4 .prg.:
20 Kompressor .prg.:
7 decompacteur .prg.:Basicprogramme werden wieder
7 decompacteur/a .prg.:auseinandergezogen
7 decompacteur .prg.:
1 videocontrol .usr.:Hilfen zur Feineinstellung des
14 videotool 3.0 .prg.:Videocontrollers zur Anpassung
1 usertable usr.seq.:an verschiedene Monitore
1 usertable ntt.seq.:
1 usertable ntg.seq.:
1 usertable pal.seq.:
1 usertable ml2.seq.:
15 crtc 6545 info .prg.:
1 dramtest .usr.:
3 dram test.bas .prg.:testet das D-RAM bis bank 14
4 dram test.ml .prg.:
2 $0800-$1fff test.prg.:
1 uhren .usr.:
6 irq-uhr .sou.seq.:
1 irq$0400 .com.prg.:
1 irq$6000 .com.prg.:
3 irq-uhr .bas .prg.:
4 uhr 2 .prg.:
11 wecker .bas.prg.:
9 weckuhr .prg.:
1 drucken .usr.:
9 sonderz.3022 .prg.:
2 hardcopy .bas .prg.:
2 hardcopy 3022 .prg.:
2 hardcopy .prg.:
2 hard-copy .prg.:
19 visi-list .prg.:Bas.-list mit ausgerückten Zeilennr.
15 profitext->ascii.prg.:ueber rs232
1 diverses .usr.:
1 MB-Menueteil .prg.:
1 ptu floppy .prg.:Floppyumschaltung bei Profitext
23 filetrans8050bes.prg.:verändert basic-startadr. f. 8032
11 filetrans8050 .prg.:
6 basicode .prg.:computer-esperanto v. radio hilversum
2 pointers .prg.:Zeiger ermitteln
3 rel pu .prg.:
8 zeichengen.700 .prg.:ermittelt hex-Werte
1 IEEE / V24 .usr.:
17 IEEE-Status/80Z.prg.:ermittelt, was angeschlossen ist
8 V.24-Transfer .prg.:
13 V.24-Initialisi. .prg.:
0 1 selbsttest .usr.:testet CBM 600/700,
1 .prg.:(alles RAM und i/os, so weit möglich)
1 setup .prg.:
20 cbm-ii.bin .prg.:Maschinencode dazu
1 Markus K .usr.:einige Programme von Markus Kammrau
85 menu.bas .prg.:programmauswahl, erlaubt
Erläuterungen zum directory
69 menu.dat .rel.:ein datensatz dazu fuer folgende prg

```

```

176 star.bas      .prg.: einige Spiele, recht übersichtlich
22 einsiedler.bas .prg.: programmiert
22 bio.bas       .prg.:
19 super.bas     .prg.:
50 dame.bas      .prg.:
8 kopf.bas       .prg.:
7 dart2.bas      .prg.:
10 tabelle.bas   .prg.:

```

### ROM-Disk

Auf dieser Diskette sind die Inhalte von EPROMs zu finden, die etwas mit den 600-700ern zu tun haben. Das sind zunächst einmal die Betriebssysteme dieser Rechner. Desweiteren sind aber auch die verschiedenen Betriebssysteme des Druckers CBM-8028 vorhanden, der wohl meistens mit einem 600-700er zusammen benutzt wird. Was noch aussteht sind die ROM-Inhalte der verschiedenen Floppys, die bei den 600/700ern gebräuchlich sind. Da einige von uns zur Zeit daran sind, sich die Floppys vorzunehmen (auch und gerade in Hinblick auf billige und schnelle Einbaulösungen), wollen wir erst deren Erfahrungen abwarten, bevor wir diverse, vielleicht überaltete Version der ROMs verbreiten.

```

1 dummy-start    .prg.:
1 dummy-doc      .prg.:
1 ---- 600/700 ---.prg.:
1 ---- kernal ----.prg.:
33 600/700 ker asc .prg.: 901244-04 Standard-Betriebssystem
33 600 ker din     .prg.: 324928-01 org. DIN-Kernal
33 700 ker din     .prg.: 324866-03 org. DIN-Kernal
65 600 ker din/asc .prg.: Clubversion umschaltbar
65 700 ker din/asc .prg.: Clunversion umschaltbar
1 ---- basic ----.prg.:
33 x20 basic high .prg.: 901240-04a Basic 256k high
33 x20 basic low  .prg.: 901241-04a      low
33 x10 basic low  .prg.: 901243-03b Basic 128k low
33 x10 basic high .prg.: 901242-03a      high
65 xy0 basic high .prg.: Bas. 128k/256k umschaltbar (club)
65 xy0 basic low  .prg.:
1 --- character --.prg.:
33 600 char asc    .prg.: 901237-01 Standard-Zeichensatz
33 600 char din/asc .prg.: 324929-01 org. DIN-Zeichensatz
33 700 char asc    .prg.: 901232-01 Standard-Zeichensatz
33 700 char din/asc .prg.: 324867-03 org. DIN-Zeichensatz
33 700 char d/a +1 .prg.: Schönere Version von
                          700 char din/asc (club)
33 700 char d/a +2 .prg.: wie drüber, mit geschweiften
                          Klammern und div. Sonderzeichen
1 ----- 500 -----.prg.:
33 510 kernal      .prg.: Betriebssystem des 500ers
33 510 bas.hi      .prg.: Basic des 500ers
33 510 bas.low     .prg.:
1 -- geschichte --.prg.: Alte Versionen für besonders
33 600 ker asc.01  .prg.:      interessierte
33 x10 basic lo.01 .prg.:

```

```

33 x10 basic hi.01 .prg.:
33 x10 basic hi.02b.prg.:
33 x10 basic lo.02b.prg.:
33 700 ker asc.03 .prg.:
1 - drucker 8028 -.prg.:
33 8028 ker m07a .prg.: Standard-Betriebssystem 8028
33 8028 ker m07f .prg.: RS-232 - Betriebssystem 8028
33 8028 ker m19b .prg.: Centronics-Betriebssystem 8028
33 8028 ker m98c .prg.: Neuere Betriebssystemversion 8028
33 8229 ker m98d .prg.: Neuere Betriebssystemversion 8229
33 8229 ker m98f .prg.: Neuste Commodore-Version,
                                auch für 8028
9 8028 car 07 .prg.: Wagensteuerung 8028/8229 für m98
9 8028 dsy 06 .prg.: Typenradsteuerung 8028/8229
9 8028 car 06 .prg.: Wagensteuerung für m07/ml9
1 -- floppy 8280 -.prg.:
9 8280 contr.neu .prg.: Fehlerfreier Diskcontroller,
                                unbedingt notwendig!

```

### Disk-Utilities I v2.0

Diese Diskette ist ein sehr würdiger Nachfolger der Version 1.0, unter der mancher hat leiden müssen. Sie bietet zwar nicht alles, was zur Diskettenbearbeitung gebraucht werden kann, ist aber auf dem besten Wege dazu.

```

21 Dir.dok ! .bas :.prg.:erstellt das gerade laufende Prg.
33 Directory .dok :.prg.:was Sie gerade lesen
29 610-dirsort .prg.:zum sortieren des Directorys
56 8/filemaster .prg.:vielleicht eine Art Knueller
19 disk handling .prg.:umfassende Hilfsprogramme
54 disk helper .prg.:
31 disk hand .prg.:
42 8/1388diskmaster.prg.:
16 file wandlung .prg.:gemeint ist der Filetyp
3 name + id change.prg.:wie der Name sagt
33 disk log .prg.:
12 disk manager .prg.:ermittelt die Startadressen
9 complete dir .prg.:dito zzgl. rel-file-info
14 8/disk log. .prg.:ermittelt Spur/Sektor/Start-Endadr.
1 startadresse .prg.:Startadresse eines Programms
9 dir-print .prg.:dito, aber fuer alle Dateien
1 - :Kopierhilfe.usr.:
18 610-filecopy 2.0.prg.:Einzellaufwerk-Kopierprogramm,ver-
9 c2.q .prg.: aendert aber die
1 c2.o .prg.: Startadressen !!
19 copy unit-unit .prg.:wohl das universellste Kopierprg.
13 bas.unitcopy .prg.:nur seq pgm usr kleiner 120 Blöcke
11 disk copy.bas .prg.:dito, aber ohne Programmauswahl
10 8/copy all .prg.:von Jim Butterfield
9 kopierer bank1 .prg.:
9 kopierer bank4 .prg.:
7 file copy .prg.:
10 mb filecopy .prg.:benoetigt Macro-Basic
25 konvertcopy .prg.:Basic in 8032-Modus konvertieren
19 MS-copy v4.2 .prg.:

```

```

6 backup sfd-sfd .prg.:danach habe ich gelehzt (schnell !)
5 blockkopie .prg.:
3 mb blockcopy .prg.:benoetigt Macro-Basic
6 bam-copy : .prg.:
1 - :loesch u.ae.usr.:recht interessant, wie verschieden
6 loeschhilfe .prg.: und verschieden effizient
11 610 filekill .prg.: man ein Problem loesen kann
12 disk file-retter.prg.:fuer gescratchte dateien
27 unscratch .prg.:dito
30 file regenerate .prg.:dito
17 file protector .prg.:bringt Schreibschutz an
7 loeschschutz .prg.:Das sollte man oefter mal anwenden !
1 - :umschalten .usr.:
4 change address .prg.:aendert Floppy - Geraeteanschrift
1 ds->ss - disk .prg.:bringt SFD oder 8250 ins 8050-Format
1 turbo dos 2.7 .prg.:zum schreiben mit Vorsicht genießen!
1 - :Monitore +a.usr.:Diese Abteilung hat es echt in sich !
9 disk mon. celle .prg.:Diskmonitor
11 disk mon. künne .prg.: dito vom Autor des 8432
18 MS-flomo vl.2 .prg.: dito fuer 610/710
15 MS-flomo620 vl.0.prg.: dito fuer 620/720
1 MS-flomo620/15 .prg.:Hilfsprogramm
5 MS-flomo620/3.1.prg.: dito
19 MS-floV1.3 druck.prg.:
8 diskmon8 .prg.:wie disk mon. kuenne, fuer 8432
26 8/dm-vl.0 ! .prg.:
23 8/diskmon/comp .prg.:Kompilat von diskmon8, fuer 8432
5 disk memory dump.prg.:Anzeige der Speicher beider Prozze
18 disk sektor620.1.prg.:Angesichts der Diskmonitore
8 sektor.pgm .prg.: sind die
1 sektor.ctl .prg.: Sektor-Anzeige-
15 t&s 1001 .prg.: Programme
35 8/display t & s .prg.: wohl nur
18 display t&s .prg.: von geringem
27 8/t&s .prg.: praktischem Nutzen
1 - :Verzeichnis.usr.:Was soll man da noch kommentieren ?
58 disk verz. 4040 .prg.:
25 8/disk-verzeichn.prg.:
32 disk verz.single.prg.:
17 8/master directo.prg.:
9 8/master index .prg.:
1 - :4040 .usr.:hierfuer ein bisschen was Spezielles
4 4040 bam view .prg.:
4040 belegung .prg.:3040 wird 4040-kompatibel
1 4040 repair18sl.prg.:
6 4040 revive disk.prg.:
1 4040 turbo .prg.:
12 4040 t&s .prg.:
9 4040 copy .prg.:
7 - :8280 .usr.:Spezialitaeten fuer 8 Zoll
6 il ibm read asc .prg.: Insbesondere das neue
11 i2 ibm read ebc .prg.: Diskcontroller-
30 i3 ibm rw ebcDic.prg.: Programm
11 i6 print ibm ebc.prg.: ist fuer
11 print ibm ebc .prg.: ein frohes
30 ibm r/w .prg.: Schaffen
5 ibmcopy .prg.: unbedingt
2 uti-copy .prg.: erforderlich,

```

```

11 ibm-dump      .prg.:      nicht nur zur
3  block r/w    .prg.:      Bearbeitung von Dateien
1  util. 8280    .prg.:      im IBM-Format.
9  contr.8280neu .prg.:neues Betriebssystem, fehlerfrei !!!
1  - ;Diagnostik .usr.:
5  check disk   .prg.:ueberprueft Lesbarkeit gesamter Disk
9  8050 rotation .prg.:      "      Drehzahl
8  8050 diagnost .prg.:und so weiter
17 8/logic diagnost.prg.: (8050- Tests sind teilweise auch auf
8  logic diagnost .prg.:      SFD und 8250 anwendbar.)
35 8/help 8050   .prg.:ueberfluessig
15 8/8050 pll-adj .prg.:
17 8/8050 burn-in .prg.:Dauertest
8  sfd 1001 test .prg.:
1  - ;Beispiele .usr.:zum Beispiel.....
9  8/rel access 4. .prg.:
8  relative file .prg.:
16 expand relative .prg.:
9  8/seq basic 4. .prg.:
12 sequential file .prg.:
13 random file .prg.:

```

### Gless-Wrede-Sampler

Diese beiden Programmautoren waren so fleißig, daß wir ihr Werk nicht in Stücke reißen wollten. Hier findet bestimmt jede(r) etwas brauchbares. Neben verschiedenen Utilities findet sich einiges Unterhaltsame, beispielsweise ein beispielloser Musikeditor. Musik kann auch im Hintergrund ablaufen. Wie gefällt beispielsweise die Fehlermeldung 'Du bist verrückt mein Kind' oder 'Ich weiß nicht, was soll es bedeuten'? Eine ganze Menge fertig programmierte Musik ist gleich mit dabei. Wo nötig, gibt es auch anständige Dokumentationen. Allerdings scheint es bei einigen Programmen Basic- oder Kernalkompatibilitaeten zu geben.

```

0 "Gless-Wrede-Samp" ii 2c
18 "610-filecopy 2.0" prg      26 "saebeltanz"          prg
9  "c2.q"              prg      12 "baba yaga"          prg
1  "c2.o"              prg      21 "biscaya"           prg
15 "610-PCR"           prg      10 "der kleine neger"   prg
2  "snoogle"           prg      9  "digiboogie"        prg
94 "610-pcr 2.1"       prg      26 "easy winners"      prg
42 "610-pcr 2.2"       prg      7  "from me to you"     prg
46 "screentext"        seq      5  "in dixieland"       prg
3  "PCR-Manual"        prg      3  "in moskau"          prg
48 "printtext"         seq      10 "j.d.'s boogie"      prg
12 "610-filekill"      prg      13 "kueken-ballett"     prg
1  "-+-+-+---+---+---+" prg      16 "manhattan look"    prg
2  "snoogle/"          prg      18 "rag of rags"        prg
36 "snoogle 4.0"       prg      4  "ragtime dance"      prg
4  "snoogle.o7"        prg      7  "thema in blues"     prg
15 "610-PCR/"          prg      3  "when the saints"    prg
119 "snoogle.mus"      prg      4  "when i'm 64"        prg
9  "lesmich.doch"      prg      39 "mozart kv 381"       prg
4  "-----"          seq      10 "praeludium ii"      prg

```

|    |                    |     |      |                                    |     |
|----|--------------------|-----|------|------------------------------------|-----|
| 23 | "snoogle 610 erg"  | seq | 5    | "praeludium vi"                    | prg |
| 23 | "snoogle 610 alt"  | seq | 21   | "major tom"                        | prg |
| 4  | "-----"            | seq | 7    | "rising sun"                       | prg |
| 9  | "titel"            | prg | 18   | "soir de paris"                    | prg |
| 9  | "level meikel"     | prg | 21   | "alla turka"                       | prg |
| 8  | "level 1"          | prg | 8    | "when i'm 64.q"                    | prg |
| 8  | "level 2"          | prg | 25   | "leicht synkop."                   | prg |
| 9  | "level 3"          | prg | 4    | "-----"                            | seq |
| 9  | "level 4"          | prg | 3    | "610-Disk'othek"                   | prg |
| 9  | "level 5"          | prg | 1    | "XXXXXXXXXXXXXXXXXX"               | prg |
| 9  | "level 6"          | prg | 11   | "adressverwaltung"                 | prg |
| 9  | "level 7"          | prg | 35   | "lp-archiv"                        | prg |
| 9  | "level 8"          | prg | 2    | "\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$" | prg |
| 9  | "level 10"         | prg | 15   | "610-target pong"                  | prg |
| 9  | "level 11"         | prg | 58   | "610-disk 5.4"                     | prg |
| 4  | "-----"            | seq | 24   | "610-chip's 3.8"                   | prg |
| 11 | "leveleditor 1.6"  | prg | 26   | "chiplist"                         | seq |
| 1  | " " pcr-musik      | prg | 26   | "610-seriell.q"                    | prg |
| 16 | "afrikaan beat"    | prg | 3    | "610-seriell.o"                    | prg |
| 7  | "air"              | prg | 10   | "mneul.q"                          | prg |
| 7  | "carmen"           | prg | 1    | "mneul.o"                          | prg |
| 22 | "axel f"           | prg | 40   | "6509-simulator"                   | prg |
| 9  | "durch die waelde" | prg | 1    | "rec-name 610"                     | seq |
| 28 | "haendel op.4/4 i" | prg | 1    | "rec-data 610"                     | seq |
| 34 | "kl. nachtmusik"   | prg | 1    | "#####"                            | prg |
| 6  | "leb wohl"         | prg | 34   | "610-disassembler"                 | prg |
| 23 | "moldau"           | prg | 14   | "610-labels 2"                     | seq |
| 10 | "moment musical"   | prg | 25   | "masch untersuch"                  | prg |
| 22 | "cat's jump"       | prg | 14   | "masch-beschreib"                  | prg |
| 4  | "triumphmarsch"    | prg | 2592 | blocks free.                       |     |

### Mathematisch-naturwissenschaftliche Anwendungen

Die Programmtitel dürften selbsterklärend sein. Ein sehr großer Teil der Programme läuft nur mit Basic 256k.

|    |                       |     |    |                    |     |
|----|-----------------------|-----|----|--------------------|-----|
| 0  | "Mathe+Naturw. 1 " ii | 2c  | 18 | "pH-Berechnung"    | prg |
| 3  | "shift run"           | prg | 64 | "Chemie Uebung I"  | prg |
| 8  | "BILD"                | prg | 93 | "Chemie Uebung II" | prg |
| 56 | "Differenzieren"      | prg | 1  | "----- 10 -----"   | prg |
| 76 | "Mathematik"          | prg | 41 | "Statistik 01"     | prg |
| 35 | "Taschenrechner"      | prg | 15 | "Chi-Quad-Konting" | prg |
| 73 | "Kurvendiskussion"    | prg | 1  | "----- 11 -----"   | prg |
| 1  | "----- 00 -----"      | prg | 5  | "test: bartlett"   | prg |
| 8  | "Nystroem Integr."    | prg | 5  | "test: david"      | prg |
| 6  | "Simpson Integral"    | prg | 6  | "test:stud-new-ke" | prg |
| 5  | "Simpson Integ II"    | prg | 8  | "test: duncan"     | prg |
| 1  | "----- 01 -----"      | prg | 6  | "test: tukey"      | prg |
| 3  | "Gleich.loesen"       | prg | 5  | "test: cochran"    | prg |
| 2  | "Lin 2x2 Gleichgn"    | prg | 8  | "test: multiple-t" | prg |
| 14 | "Transzent.Gleich"    | prg | 5  | "test:link-wallac" | prg |
| 5  | "Num.Lsg.Diff.Gl"     | prg | 6  | "test: levene"     | prg |
| 1  | "Quadratwurzel"       | prg | 7  | "test: dunnett"    | prg |
| 2  | "Quadratwurzel II"    | prg | 14 | "test: multi-tl"   | prg |
| 1  | "----- 02 -----"      | prg | 11 | "var.ana:zw.mwwow" | prg |
| 5  | "Fourierberechn"      | prg | 9  | "var.ana:zw.mwwmw" | prg |

|    |                     |     |      |                    |     |
|----|---------------------|-----|------|--------------------|-----|
| 14 | "Fourieranalyse"    | prg | 7    | "var.ana:einfach"  | prg |
| 1  | "----- 03 -----"    | prg | 9    | "linear Kontraste" | prg |
| 6  | "Statistik"         | prg | 6    | "Daten eingeben"   | prg |
| 6  | "Histogramm"        | prg | 7    | "Daten ausgeben"   | prg |
| 2  | "Elementzahl"       | prg | 1    | "----- 12 -----"   | prg |
| 4  | "Basis raten"       | prg | 50   | "Konfidenz"        | prg |
| 24 | "Wettrechnen"       | prg | 1    | "----- 13 -----"   | prg |
| 1  | "Zahlen runden"     | prg | 9    | "Lin. Regression"  | prg |
| 4  | "MW Standardabw"    | prg | 8    | "Klassenhaeufigk." | prg |
| 3  | " -Berechnung"      | prg | 9    | "Lin.Reg.Druck"    | prg |
| 5  | " -Berechnen"       | prg | 8    | "St.Kennwert."     | prg |
| 6  | " -Stellen berech"  | prg | 7    | "St.Kennwert.2"    | prg |
| 1  | "----- 04 -----"    | prg | 4    | "Geom.Mittel"      | prg |
| 3  | "Primfaktoren"      | prg | 4    | "Harmon.Mittel"    | prg |
| 1  | "Primfaktoren II"   | prg | 8    | "Modus/Median"     | prg |
| 1  | "Primzahl"          | prg | 4    | "Median klass."    | prg |
| 1  | "Primz. von a-b"    | prg | 9    | "Exp. Regr."       | prg |
| 1  | "Primzahlzwillig"   | prg | 8    | "Rangkorrelation"  | prg |
| 10 | "Primfakt.Zerleg."  | prg | 9    | "Geometr.Regr."    | prg |
| 1  | "----- 05 -----"    | prg | 9    | "Log.Regression"   | prg |
| 15 | "Berechnungen"      | prg | 11   | "Polynom-Reg."     | prg |
| 1  | "Teilbarkeit"       | prg | 11   | "Glockenkurve"     | prg |
| 1  | "Teilbarkeit II"    | prg | 4    | "Vier-Felder-K."   | prg |
| 1  | "Teiler"            | prg | 12   | "Multiple Lin."    | prg |
| 1  | "Quersumme"         | prg | 5    | "Randomisierung"   | prg |
| 2  | "Quersummenregel"   | prg | 6    | "Normal. Zufall."  | prg |
| 1  | "Bruch kuerzen"     | prg | 5    | "Binom. Zufall."   | prg |
| 2  | "Dezimalbruch"      | prg | 7    | "Vertrauensint."   | prg |
| 1  | "Division m. Rest"  | prg | 6    | "Vertr. Haeuf."    | prg |
| 2  | "Umw in Dezimalz"   | prg | 5    | "Stich-Umfang"     | prg |
| 1  | "Umw Dezimalbruch"  | prg | 7    | "Stich. Haeufig"   | prg |
| 2  | "G-adische Darst."  | prg | 6    | "Binomialvert."    | prg |
| 1  | "gr.gemein.Teiler"  | prg | 8    | "Hypergeo.Vert."   | prg |
| 1  | "kl.gemein.Vielf"   | prg | 6    | "Poisson-Vert."    | prg |
| 4  | "Roem. Zahlen"      | prg | 5    | "Dichte Normal"    | prg |
| 3  | "In Worten"         | prg | 6    | "Integral Norm."   | prg |
| 1  | "----- 06 -----"    | prg | 5    | "Schranken Norm."  | prg |
| 2  | "Kub. Gleichung"    | prg | 7    | "Integr. t-Vert."  | prg |
| 3  | "Polynom"           | prg | 7    | "Schranke t-Vert." | prg |
| 1  | "Mitternachtsform"  | prg | 5    | "Integr. F-Vert."  | prg |
| 17 | "Dreieckberechnngn" | prg | 5    | "Integr.Chi-Quad." | prg |
| 4  | "Koord. umwandeln"  | prg | 1    | "----- 14 -----"   | prg |
| 6  | "Regula falsi"      | prg | 24   | "statistik"        | prg |
| 1  | "----- 07 -----"    | prg | 12   | "arithmet. reihe"  | prg |
| 14 | "hex-dez-bin"       | prg | 2    | "techn.notation"   | prg |
| 3  | "dez-hex-dual"      | prg | 2    | "primfaktoran"     | prg |
| 1  | "----- 08 -----"    | prg | 2    | "prim. sieb"       | prg |
| 10 | "Mathe-Uebungen"    | prg | 3    | "fakultaet"        | prg |
| 11 | " Naehierungen"     | prg | 4    | "roem. zahlen"     | prg |
| 13 | "Radioakt.Zerfall"  | prg | 8    | "nullstellen"      | prg |
| 1  | "----- 09 -----"    | prg | 10   | "multiplikation"   | prg |
| 61 | "Molgewicht"        | prg | 13   | "kurvendiskuss."   | prg |
| 62 | "Chem. Berechnung"  | prg | 23   | "Basicalc"         | prg |
| 17 | "Formelmasse"       | prg | 2718 | blocks free.       |     |



### fig-forth 78

Zu dieser freien Forth-Implementation von Dr. Wolfgang Kessel habe ich weiter vorne schon ein paar Worte verloren. Hier können Sie sich vom Umfang der von ihm geleisteten Arbeit überzeugen.

|    |                    |    |     |      |                    |     |
|----|--------------------|----|-----|------|--------------------|-----|
| 0  | "fig-forth78/orig" | 12 | 2c  | 28   | "forth80/2.text"   | seq |
| 6  | "start8432.21"     |    | prg | 38   | "forth80/3.text"   | seq |
| 81 | "8432.21"          |    | prg | 33   | "forth80/4.text"   | seq |
| 4  | "bank15.20/1024"   |    | prg | 34   | "forth80/5.text"   | seq |
| 2  | "t1"               |    | prg | 21   | "forth80/6.text"   | seq |
| 2  | "td"               |    | prg | 33   | "forth80/7.text"   | seq |
| 2  | "t2"               |    | prg | 35   | "forth80/8.text"   | seq |
| 1  | "-- forth-sys ---" |    | seq | 27   | "forth80/9.text"   | seq |
| 11 | "forth.readme"     |    | prg | 18   | "forth80/10.text"  | seq |
| 1  | "forth.com"        |    | prg | 41   | "forth80/11.text"  | seq |
| 26 | "forth78"          |    | prg | 46   | "forth80/12.text"  | seq |
| 3  | "opsys78"          |    | prg | 1    | "-- forth-fils --" | seq |
| 1  | "-- forth-text --" |    | seq | 466  | "fil0"             | rel |
| 1  | "edit"             |    | prg | 539  | "fill"             | rel |
| 17 | "editor700.7"      |    | prg | 1    | "-- forth-ende --" | seq |
| 27 | "forth80/1.text"   |    | seq | 2213 | blocks free        |     |

### comal 0.14 (8432)

Sehen Sie selbst, wie eifrig unser Kollege Dr. Meingast war. Wie früher schon bemerkt, bietet Comal ausgezeichnete Möglichkeiten und ist in dieser freien Version ganz besonders zu empfehlen, um programmieren zu lernen, ohne sich wie mit Basic den Programmierstil zu versauen. Die auf der Diskette enthaltene Version für den C-64 unterstützt auch dessen Grafikmöglichkeiten sehr gut. Die Comal-Einführung ist auch zu DM 2,- auf Papier erhältlich.

|     |                    |    |     |   |                     |     |
|-----|--------------------|----|-----|---|---------------------|-----|
| 0   | "comal 0.14 8432 " | ii | 2c  | 2 | "dreierwurf.f"      | seq |
| 11  | "startCOMAL0.14"   |    | prg | 1 | "dritte'wurzel.f"   | seq |
| 6   | "cbmcomalerrors"   |    | seq | 1 | "empty'buffer.p"    | seq |
| 104 | "comal 80 / 0.14"  |    | prg | 2 | "fenster(x;y).p"    | seq |
| 1   | "-----"            |    | seq | 4 | "fetch.proc.379.L"  | seq |
| 9   | "diskhelp48.L"     |    | seq | 2 | "fib(n).f"          | seq |
| 1   | "-----"            |    | seq | 1 | "file'exists381.p"  | seq |
| 11  | "start C-64 .14"   |    | prg | 6 | "floppy.p.f"        | seq |
| 6   | "boot comal"       |    | prg | 1 | "floppyfehler.f"    | seq |
| 131 | "comal 0.14.ev"    |    | prg | 3 | "lese'dir.p.2.01"   | seq |
| 6   | "comalerrors"      |    | seq | 1 | "func'dez.L2000"    | seq |
| 1   | "-----"            |    | seq | 2 | "ganzzahl.f"        | seq |
| 3   | "enter '.....'"    |    | seq | 1 | "gcd(a#;b#).f"      | seq |
| 16  | "Einf.lesen.L"     |    | seq | 1 | "gerundet.f"        | seq |
| 16  | "Einf.lesen2.00.L" |    | seq | 1 | "get .f"            | seq |
| 14  | "Einf.l.C64 .14.L" |    | seq | 2 | "get'char.pr.384."  | seq |
| 16  | "Einf.lesen2.01.L" |    | seq | 1 | "get'record\$.2.00" | seq |
| 11  | "Stichwort"        |    | seq | 1 | "get'valid.p386.L"  | seq |
| 11  | "Inhaltsverz"      |    | seq | 1 | "hex\$(n).f"        | seq |
| 3   | "Inhaltsverz.1"    |    | seq | 2 | "hex(n;refh\$.p)"   | seq |
| 38  | "Instruct.0.14"    |    | seq | 3 | "hex(zahl).p"       | seq |

|    |                    |     |    |                    |     |
|----|--------------------|-----|----|--------------------|-----|
| 3  | "Inhalt-schr2.01"  | seq | 9  | "inhalt"           | seq |
| 11 | "Stichwort C-64"   | seq | 2  | "ja'nein.f"        | seq |
| 11 | "COMAL-Einf00leat" | seq | 7  | "kommazahl.f"      | seq |
|    | :                  |     | 2  | "kuerze.p"         | seq |
| 5  | "COMAL-Einf037fKt" | seq | 2  | "lies(namen\$.p)"  | seq |
| 4  | "-8-Damen"         | seq | 2  | "lower'to'upper.p" | seq |
| 12 | "adresse.870222a"  | seq | 4  | "n-te Wurzel.L"    | seq |
| 6  | "beisp.disk'get'"  | seq | 1  | "perm(x).f"        | seq |
| 2  | "beispielcase.L"   | seq | 1  | "position.391.p"   | seq |
| 2  | "beispielwhile.L"  | seq | 2  | "pot(z;n).f"       | seq |
| 4  | "bounce.L"         | seq | 3  | "proc.mahnung"     | seq |
| 9  | "bruchrechnen"     | seq | 1  | "put'record.392.p" | seq |
| 2  | "change unit addr" | seq | 2  | "quadratwurzel.L"  | seq |
| 2  | "data-Beisp."      | seq | 6  | "quicksort(a\$.p)" | seq |
| 26 | "evaluator.1"      | seq | 11 | "quicksort.L"      | seq |
| 28 | "formatter.L"      | seq | 1  | "rad(x).f"         | seq |
| 10 | "hannibal.L"       | seq | 4  | "rahmen(lo;ru).p"  | seq |
| 12 | "hanoi.L"          | seq | 1  | "randomize.394.p"  | seq |
| 6  | "kalender.L"       | seq | 1  | "rekufib.L"        | seq |
| 3  | "kickshaw.L"       | seq | 4  | "remove //.L"      | seq |
| 4  | "kuerzen1.ref.L"   | seq | 1  | "round.func.395.L" | seq |
| 3  | "kuerzen1.L"       | seq | 1  | "satzlaenge().f"   | seq |
| 7  | "mag.quadrat.1"    | seq | 2  | "scankey.pr.396.L" | seq |
| 26 | "othello.L"        | seq | 1  | "schaltjahr.f"     | seq |
| 7  | "pfeffernuss"      | seq | 3  | "schreib(nam\$.p)" | seq |
| 2  | "piberechnung.L"   | seq | 1  | "screen'pos.398.f" | seq |
| 6  | "praemie.2"        | seq | 2  | "screen80.f"       | seq |
| 13 | "pythagoras-1"     | seq | 2  | "screenload.p"     | seq |
| 5  | "rechner.L"        | seq | 2  | "screensaven.p"    | seq |
| 6  | "rel.Datei.1"      | seq | 8  | "scrn"             | seq |
| 10 | "rel.Datei.L.1"    | seq | 1  | "set'window.p"     | seq |
| 3  | "schaltjahr"       | seq | 1  | "shift.func.399.L" | seq |
| 3  | "schaltjahr.tex"   | seq | 1  | "sqrt(x).f"        | seq |
| 6  | "sortieren.8702"   | seq | 2  | "stringbeispiel"   | seq |
| 2  | "suchprg.0.14.L"   | seq | 5  | "strstr5000.1"     | seq |
| 7  | "textanalyse.L"    | seq | 2  | "take'in.p.400.L"  | seq |
| 2  | "textbaustein.1.L" | seq | 8  | "take'in.Beisp.L"  | seq |
| 1  | "textzunahme"      | seq | 2  | "uhr.f.stellen.p"  | seq |
| 2  | "verdoppeln"       | seq | 2  | "uhr1"             | seq |
| 1  | "vorwaerts"        | seq | 2  | "value.402.f"      | seq |
| 3  | "wahrh.tab.L"      | seq | 1  | "value.f....402.L" | seq |
| 7  | "wochentag.L"      | seq | 1  | "von8250zu8050.p"  | seq |
| 4  | "zahlraten"        | seq | 1  | "wait(refa\$.p)"   | seq |
| 3  | "5.Wurzel.L"       | seq | 2  | "window.p"         | seq |
| 1  | "arccos(x).f"      | seq | 1  | "yes'no.p"         | seq |
| 1  | "arcsin(x).f"      | seq | 2  | "zahl50.f"         | seq |
| 1  | "auswahl.f"        | seq | 1  | "ziffer.f"         | seq |
| 2  | "auswahl2.f"       | seq | 2  | "dice.e"           | prg |
| 1  | "beisp.raahmen"    | seq | 3  | "S019.rem.....L"   | seq |
| 3  | "binsearch.p"      | seq | 1  | "S021.abs.....L"   | seq |
| 3  | "blockreadw.p"     | seq | 2  | "S023.and then.L"  | seq |
| 6  | "blockreadw0.14."  | seq | 1  | "S049.chr\$.....L" | seq |
| 3  | "blockreadw2.00."  | seq | 2  | "S051.close....L"  | seq |
| 1  | "clear.p"          | seq | 1  | "S053.closed...L"  | seq |
| 2  | "create(.)p"       | seq | 2  | "S063.data....L"   | seq |
| 2  | "create.prc.369.L" | seq | 2  | "S081.div.....L"   | seq |
| 1  | "current'col.f"    | seq | 1  | "S096.endfor...L"  | seq |
| 1  | "current'row.f"    | seq | 2  | "S098.endfunc..L"  | seq |

|   |                     |     |    |                    |     |
|---|---------------------|-----|----|--------------------|-----|
| 1 | "cursor(r;c).p"     | seq | 2  | "S104.func.fak.L"  | seq |
| 1 | "cursor(z.s).p"     | seq | 2  | "S108.endwhile.L"  | seq |
| 2 | "cursor.prc.370.L"  | seq | 3  | "S156.import...L"  | seq |
| 1 | "dec(hex\$).f"      | seq | 1  | "Seite 342.1"      | seq |
| 1 | "deg(x).f"          | seq | 11 | "Start COMAL 0.14" | prg |
| 1 | "delay.p"           | seq | 3  | "boot 8432.24"     | prg |
| 1 | "dez .f"            | seq | 81 | "8432.24"          | prg |
| 1 | "dez.f"             | seq | 5  | "tabler"           | prg |
| 1 | "disk'get'skip.L"   | seq | 3  | "t1"               | prg |
| 1 | "disk'get'string."  | seq | 3  | "t2"               | prg |
| 4 | "disk'get.fun372L"  | seq | 3  | "td"               | prg |
| 1 | "disk(command\$).p" | seq | 3  | "tw"               | prg |
| 3 | "displ.disp"        | seq | 5  | "bank F.23.14"     | prg |
|   |                     |     |    | 2764 blocks free.  |     |

### Assembler I v2.0

Diese Diskette aus der Reihe der urheberrechtlich geschützten gibt den Assemblerprogrammierern eine Menge "lebensnotwendiger" Werkzeuge an die Hand. Der ema ist an anderer Stelle ausführlich beschrieben. Die beiden Monitore (S-Mon und Monitor 6000) ergänzen sich gegenseitig. Der 8432-Emulator wird zwar von keinem der Programme gebraucht, ist aber der Vollständigkeit halber mit aufgenommen. Das Bank-0x-Kernal soll als Basis für größere Assemblerprogramme dienen. Es ermöglicht, Programme in den Banken 0-14 fast ohne Geschwindigkeitsverlust (6,7%) bei Betriebssystemzugriffen zu betreiben. Das b0x-Basic läßt Eingriffe in den Basic-Interpreter zu, ohne daß die EPROMs geändert werden müssen. Im Gegensatz zum Bank-0x-Kernal läuft das b0x-Basic auch in Bank 15.

Der ema ist nur für die private Nutzung innerhalb des Vereins freigegeben. Die Diskette wird mit Handbuch geliefert, in dem alle Programm ausführlich beschrieben sind.

|    |                   |     |    |                   |     |
|----|-------------------|-----|----|-------------------|-----|
| 0  | "Assembler v2.0 " | 22  | 2c |                   |     |
| 1  | " ema 600/700     | prg | 1  | " s-mon           | prg |
| 2  | "ema-b0x-start"   | prg | 33 | "cod.smon\$2x20"  | prg |
| 34 | "ema-b0x"         | prg | 33 | "cod.smon\$4x20"  | prg |
| 36 | "ema-600/700-org" | prg | 13 | "bas.unitcopy"    | prg |
| 1  | " b0x-kernal      | prg | 11 | "ass.unitcopy"    | seq |
| 2  | "b0x-ker-start"   | prg | 9  | "ass.codewand"    | seq |
| 33 | "b0x-ker-v2.0-a7" | prg | 1  | " 8432            | prg |
| 33 | "b0x-ker-v2.0-a6" | prg | 3  | "boot 8432.24"    | prg |
| 33 | "b0x-ker-v2.0-d6" | prg | 81 | "8432.24"         | prg |
| 33 | "b0x-ker-v2.0-d7" | prg | 5  | "tabler"          | prg |
| 65 | "b0x-Basic 256k"  | prg | 3  | "t1"              | prg |
| 65 | "b0x-Basic 128k"  | prg | 3  | "t2"              | prg |
| 1  | " monitor 6000    | prg | 3  | "td"              | prg |
| 1  | "/"               | seq | 3  | "tw"              | prg |
| 19 | "Monitor 6000"    | prg | 5  | "bank F.23.14"    | prg |
| 19 | "Monitor 6000-a"  | prg |    | 3548 blocks free. |     |

### Protex

Hier handelt es sich nicht im eigentlichen Sinne um eine Clubdiskette, sondern wir erwerben die (kopiergeschützten!) Disketten von einem offiziellen Lizenznehmer. Lediglich die Handbücher stammen aus eigener Produktion sowie eine Sonderleistung des Vereins, eine extra Diskette mit Druckertreibern,\* die auch einzeln erhältlich ist.

Dies ist übrigens ein typischer Trennungsfehler.

### Macro-Basic

Hierzu ist in diesem Heft an anderer Stelle schon reichlich lesen. Derzeit sind zwei Disketten zu haben, nämlich die mit einer kompletten Befehlserweiterung und ausführlichem Handbuch und die mit den Beispielprogrammen aus dem Handbuch. Wer eine komplette Erweiterung erwirbt, dem wird eine Systemdiskette zur Erstellung individueller Erweiterungen zu gegebener Zeit nachgeliefert.

In Kürze wird Macro-Basic auch für andere CBM-Rechner hier zu haben sein. Die jeweils erste Version (mit Handbuch) kostet DM 75,-, jede weitere DM 40,-. Da bei anderen Rechnern aus Mangel an Speicherplatz komplette Erweiterungen nicht möglich sind, müssen wir noch auf das Entdornen dieser Versionen warten.

All w. files are ASCII format, they can be read with Superscript or the Keytrix 'text command. All +files are machine code or data used by other programs. All other programs can be loaded and run unless a note says to just list. All programs by Jim Butterfield have JB initials shown in comments. ML stands for machine language or machine code. The .bl28 addition is not consistent in my files. All programs on this disk run in Bl28 (except those in the 8432 section), some also run in B256 and some run in all CBM machines (most of JB programs, and most of my non-utility programs).

```

??"w.contents"          s-what you're now reading.
6 "pre-superscript"    p-(v3)run before loading SS. Fixes bounce.
15"bl28 irq rate"      p-  explains via disassembly: load, list.
1  "+ta irq.400"       p   object code of a similar version
7  "spool bl28"        p-disk to printer s files. load, run, sys.
10"w.bl28 spool"       s   explains more.
1  "<----- a ----->" p
21"dcl bl28.pal"       p-for ML people -B definitions
2  "+btxfer.fe48"      p-transfer suences, must be used with all ML.
1  "+xcall.fe00"       p-allows calling non-kernal rom routines.
6  "xcall"            p   source code, load and list.
80"w.bl28 xcalls"      s   long winded explanation, updated.
1  "<----- h ----->" p
5  "supermonb v7"      p-ML better monitor for bl28/256. Load,run, it
16"+smb v7.0400"       p   loads one of the +files. If you have mem.
16"+smb v7.e000"       p   in the cartridge one of +file+ or +file++
16"+smb v7.0400+"      p   is loaded.
16"+smb v7.e000+"      p
16"+smb v7.0400++"     p
16"+smb v7.e000++"     p
46"supermonb v7.ins"  p   supermon instructions. load,run.
1  "<----- b ----->" p
4  "keytrix v7"        p-Basic utilities. Load, run, it loads needed
15"+bl kxb128.v7"      p   +files. If you have mem. in the cartridge,
15"+bl kxb256.v7"      p   one of the +file+ or +file++ loads.
2  "+bl5kxb128.v7"     p
2  "+bl5kxb256.v7"     p
2  "+bl5kxb128.v7+"    p
2  "+bl5kxb256.v7+"    p
2  "+bl5kxb128.v7++"   p
2  "+bl5kxb256.v7++"   p
47"keytrix v7.ins"    p   keytrix instructions: load,run.
1  "<----- c ----->" p
4  "change dev#"       p-JB disk device soft change
23"copy-all bl28.4"   p-JB copies files across drives or devices.
2  "+copy-a.2lc3c"     p   Allows rearranging. Load/run. +file loads.
8  "disk view.bl28"    p-JB disk utilities. Load, run.
11"disk logger.bl28"  p   view shows track/sector contents, log logs
1  "+disklog.400"      p   directory + header address + length, mod
7  "disk mod.bl28"     p   allows rewriting using screen input.
4  "big rel file"      p-switches 8050 to hold more than 700 records
14"reader/writer.3"   p-puts out contents of CABS files to screen
1  "<----- d ----->" p   or disk or printer
22"bl28<-->amiga.2"    p-RS232 file transfers for B-people fooling
2  "+bl28<>am.2"       p   with Amigas! (subject to change if Amiga's

```

```

1 "<---- g ---->"      p protocols change).
4 "picdesign"           p-very optionalp- draw inp mode.
7 "4023 scr dump"      p-Exact screen dump to 4023 printer load,run,
8 "p.dump ins"         p this gets loaded as further instructions.
4 "p.cats"             p so do 4 cats
1 "<---- f ---->"      p
1 "configure bl28"     p-if you don't use Supermon/Keytrix, use this.
2 "mykeys"             p-my key definitions. load/list/edit/run.
2 "datamaker"          p-makes DATA lines from ML code, primitive.
4 "verifierbl28"       p-might end up at Transactor.
5 "stringthing bl28"   p-JB original, LINE INPUT simulation.
4 "qplot.bl28"         p-building blocks for 160x50 plots
3 "primes"             p-a neat way to work prime numbers
11"squares"           p-two player game, no frills
16"rotate"            p-a puzzle
5 "tab"               p-tabs data into intervals - demo
4 "sm sort"           p-pointer sort demo
3 "waitline"          p-waiting in line - simulation, JB.
8 "words match"       p-template teachers can use
27"w.exp15"           s-how to expand Bl28 in the cartridge
1 "<---- e ---->"      p-----
22"list all/jim b"     p-JB LISTS programs from all computers
4 "cross.bl28-1.1"     p-JB crossreferences basic programs FAST.
5 "+cross.bl28"        p this gets loaded by cross*
12"twin bagels"        p-JB two computer vs human 'mastermind'
23"grungy towers"     p-JB you gotta solve a murder story
2 "impossible"         p-JB this is a weird program
16"stepwise"          p-JB three statistics programs
3 "step illustr"       p- demo
6 "smoothing"          p-JB triple exponential smoothing, no less.
15"diary"             p-JB builds, uses, remembers a calendar
2 "events"            s this file is used by diaryp.
1 " <--8432 utl-->"    p-----
3 "Start8432.21"      p-THE 8432 system is an 80-col Pet emulator,
2 "t1"                p but it is better than the PET in that it's
2 "td.alt"            p faster! Many 80-col Pet programs even
2 "t2"                p including some machine code programs do
2 "td"                p run on the B machine. Load and run"Start"
81"8432.21"           p after loading and running the instructions
4 "bank F.20 '1024"    p file it will load the other files.
5 "instructions8432"   p explained
9 "supermon4.rel"     p-just load and run. X exits the monitor.
5 "aid4.6b00"         p-Bas.utilities: load, sys6*4096+11*256, new.
5 "aid4.7b00"         p- " : " sys7*4096+11*256, new.
                        use $6b00 version if you loaded supermon.
6 "cross"             p-cross-references basic programs
1 "<---- i ---->"      p-----
6 "peeker 80 col"     p-Gerry Neufeld's disk utilities - including
5 "errors 8050"        p- backup for some protected programs. to
10"backup 8050"        p- backup - use a newly formatted disk.
5 "seek id's 8050"    p-
9 "vir dir 8050"      p- great directory reader
8 "confirm 8050"      p-
8 "8050 errors/2"     p- error maker
21"clone 22 & 23"      p- (allps: load, run)
1 "<---- j ---->"      p-----
??"w.bug2/86"         s-article for The CBUG Escape.
1 "<---- k ---->"      p

```

## CBUG -Kernaghan's Utilities

### UTILITIES

'Basic crs cmpld' & 'bl5' prints a list of variables used and lines using them, also line numbers called by other lines.  
'Change 8050' changes the device number from 8 to 9 through 15.  
'Chg addrv2.gb' & 'bl5' modify the load address of a program.  
'Compactor' will compact a program for faster running and less space.  
'Compactor.fast' does the same, faster. Uses 8432 Pet emulator.  
'Conv c64-bl28' & 'conv patchbm-bl28' are charts of memory addresses.  
'Copy' copies files from drive 0 to drive 1.  
'Copy 8/9 chg add' copies from device 8 to 9, while changing address  
'Copy all mult' formats a disk and copies all programs.  
'Copy alpha v2' can format, copy all or selected files.  
'Copy-sort' copies all files from 0 to 1 with optional alpha sorting.  
'Data writer' reads code in memory, then outputs it for data statem..  
'Dir 2 col ptr v2' prints a 2-column directory.  
'Dir printout' is an elegant little directory printer.  
'Disassem pet sym' disassembles from memory, to screen or printer.  
'Disassembler p' is a Butterfield disassembler for screen/printer.  
'Disassembler.d' works with the files on disk, for screen/printer.  
'Disinterpreter' converts code in memory to BASIC on screen.  
'Disk check 2.0' tests disks in device 8, drive 0 or 1.  
'Disk commands' allows loading, scratching, renaming, or copying.  
'Disk doctor.b' sector of a specified track, allowing changes.  
'Disk logger v2' provides a deluxe directory, with T/S and loading addresses.  
'Disk maintenance' offers backup, copy, dir, new, rename, scratch..  
'Disk manager' has formatting, single program copy, renaming, erasing.  
'Disk revival' is an unscratcher, to restore scratched files.  
'Disk T/S' displays any track/sector on screen/printer.  
'Disk tidier' helps delete unwanted files, using the directory.  
'Disk unscratch' is an excellent file restorer for scratched files  
'Diskview' not only views, it can change what it shows you.  
'Dsk hdr/id chg' will change disk header and/or id characters.  
'Hex dump dos' is a hex dump of disk drive processor or IEEE proc.  
'Hex dumper 80' is a hex dumper from disk.  
'Load address is' displays the load address of any program file.  
'MSD purge.b' selectively scratches program files on an MSD drive.  
'Peek print' displays the results of a keypress at any location.  
'Performance 1.0' is a disk and drive checker.  
'Pgm data.disk & pgm search.disk locate programs in your disk library  
'Prog adds.gb' & 'bl5' display both loading and ending program addr.  
'Prog print' allows easy printing of programs.  
'Prog.printout.v2' prints program listings.  
'Seq list cbm/asc' reads seq files such as those on this disk.  
'Seq lister' quickly displays a sequential file to screen.  
'Seq prg-bas prg' tokenizes listings of basic programs.  
'Seq read/print' is a sequential file reader and printer.  
'Typist' uses computer and printer as a correctable typewriter.  
'Uncompactor' stretches those compacted or regular programs.  
'WP prg printout' is a good program for printing most wordpro files.  
'WP read/print' is good for displaying wp files, but prints slowly.

## CALENDARS

'Calendar printer' outputs any month to screen or printer.  
'Calendars screen' displays any month from 1904-2099.  
'Day of week' outputs the day of the week from day, month, and year.  
'Memory calendar' is an appointment record. (Uses 'March').  
'Per calendar 2' will display any year and/or any month's calendar.  
'Perp calendar' will display a calendar accurate from 1582 on.

## EDUCATIONAL

'Body weight' calculates more than you can believe about your body.  
'Brain balance' helps determine which side of your brain is most used  
'Life expectancy' will estimate your total span of years.  
'Long division' drills students in progressively difficult problems.  
'Yellow light' is a reaction time evaluation.

## ENTERTAINMENT

'Bioprinter' furnishes screen or hardcopy for biorythm plots.  
'Biorythm 2.5' displays the plots on screen only.

## GAMES

'Afo/balloon' is a graphics game wherein the player tries to hit AFO  
'Blackjack' is the old standby, with graphics.  
'Blackjack-jb' takes place at the Butterfield Soc. & Recreat. Club  
'Bomber' is a graphics game, with the player trying to down the bomber

'Checkers80.gb' & 'bl5' A wide screen checker game.  
'Dominoes.c' is an easy to play graphics game.  
'Hamurabi' Try your hand at governing Sumeria for a 10 year term.  
'Kingdom' is an improved version similar to hamurabi.  
'Labyrinth.v2' lets you specify a 3-D lab., then try to get through.  
'Lunar lander' is learning to balance velocity, fuel and time.  
'Super lander' is an improved version of lunar lander.

## GRAPHICS

'75 names' can be a grid filled with one character, or 80 columns repeating a name, phrase, or sentence.  
'ASCII screen' pokes the screen RAM with any ASCII code typed in.  
'Banner' outputs large characters or signs on the printer.  
'Bar chart' is a demonstration of graphics display.  
'Bar graph subrtm' is a demo intended to be interfaced with a program  
'Bar grf' is a horizontal bar graph presentation to screen/printer.  
'Bargraph' is a comprehensive vertical bar auto-scaling printer.  
'Big letter ads' displays, on screen, moving messages up to 20 lines.  
'Big letters' displays a brief message in 3-D characters.  
'Big time' displays a large running digital clock.  
'Pedigree charts' prints blank pedigree charts for geneologists.  
'Squiggle' displays entertaining designs.

## SOUND

'Yesterday' from the same song. Just 'run'.

## TECHNICAL



'555 timer' helps design a timer circuit, for electronics design.  
 'Anglo-metric' is a conversion program, easy to use.  
 'Lo-pass filter' helps design a low-pass active filter circuit.  
 'Metric conv' is similar to 'anglo-metric'. Take your pick.  
 'Reg ps design' helps design a complete DC power supply.  
 'TV satellites.v2' calculates satellite positions.

### CBUG-Diagnostik

Dies ist eine Diskette voller Testprogramme, neben dem 600/700-Selbsttest die Testprogramme für die Diskettenstationen, die für den 8032 geschrieben sind, und auch nicht alle unter 8432-Emulation laufen. Tony Goceliak hat einiges davon an unsere Rechner angepaßt und zu dieser Diskette zusammengestellt.

|     |                       |     |     |                     |     |
|-----|-----------------------|-----|-----|---------------------|-----|
| 0   | "b/i3e diagnostic" aj | 2c  | 1   | "---disk utils---"  | seq |
| 1   | "----computer----     | seq | 5   | "change device #"   | prg |
| 1   | "b-128/256 test"      | prg | 4   | "change device 2"   | prg |
| 20  | "load 1"              | prg | 1   | "drive 0 format"    | prg |
| 1   | "load 2"              | prg | 1   | "drive 1 format"    | prg |
| 1   | "---monitor ----"     | seq | 3   | "drive identifier"  | prg |
| 1   | "screen 80"           | prg | 1   | "--print tests--"   | seq |
| 1   | "---disk drive---     | seq | 27  | "printer demo"      | prg |
| 1   | "---basic xfers--"    | seq | 9   | "printer test"      | prg |
| 49  | "prod soft 2.5"       | prg | 1   | "---extra test---   | seq |
| 30  | "prod soft dos2.7"    | prg | 39  | "new rom analog"    | prg |
| 6   | "caution"             | prg | 1   | "----demo of ----"  | seq |
| 18  | "8050 memory test"    | prg | 1   | "--fast m/l xfer--" | seq |
| 17  | "4040 memory test"    | prg | 30  | "prod soft 2.7+"    | prg |
| 13  | "2031 memory test"    | prg | 1   | "disk m/l xfer+"    | prg |
| 37  | "digital 2.7 test"    | prg | 1   | "--- notice ---"    | seq |
| 31  | "2031 board test"     | prg | 1   | "---the above---    | seq |
| 36  | "system test dual"    | prg | 1   | "--programs have--" | seq |
| 32  | "sys test"            | prg | 1   | "--been patched--"  | seq |
| 32  | "2031 sys test"       | prg | 1   | "by a j goceliak"   | seq |
| 35  | "analog test"         | prg | 1   | "--to run on the--" | seq |
| 36  | "2040/4040 test"      | prg | 1   | "--b 128/256 but--" | seq |
| 550 | "drive perf test"     | prg | 1   | "---only those---   | seq |
| 19  | "pll test"            | prg | 1   | "--dealing with--"  | seq |
| 1   | "---alignment ---"    | seq | 1   | "--mpi dos 2.7 --"  | seq |
| 1   | "----of drive----     | seq | 1   | "---have been---    | seq |
| 1   | "---unit tests---     | seq | 513 | "tested by being"   | seq |
| 64  | "mpi test"            | prg | 1   | "--actually run--"  | seq |
| 60  | "sfd-1001 test"       | prg | 1   | "--caveat emptor--" | seq |
| 63  | "microp test"         | prg | 1   | "-----"             | seq |
| 60  | "tandon test"         | prg | 1   | "--happy testing--" | seq |
| 55  | "shugart test"        | prg | 1   | "++++++++++++++"    | seq |
| 1   | "--max run time--"    | seq | 1   | "++++freebie +++"   | seq |
| 72  | "10 pass softerr"     | prg | 1   | "++++++++++++++"    | seq |
| 72  | "100 pass softerr"    | prg | 7   | "disk concatenate"  | prg |
| 81  | "disk final test"     | prg | 2   | "screen gong test"  | prg |
|     |                       |     | 5   | "uk test loader"    | prg |
|     |                       |     | 13  | "uk diagnostic"     | prg |
|     |                       |     |     | 1140 blocks free.   |     |

## CBUG - Goceliak's patch

Tony Goceliak ist einer der fleißigsten und wohl auch fähigsten Autoren des CBUG. Diese seine erste Clubdiskette ist in erster Linie für 8050-Betreiber interessant, aber die meisten Programme für DOS 2.7 laufen auf allen entsprechenden Geräten.

Es gibt aber noch eine Menge anderer interessanter Programme hier. Die meisten Titel sind wohl aussagefähig genug. Wie bei den meisten CBUG-Disketten gibt es ausführliche Erklärungen zu den Programmen auf der Diskette.

|     |                          |    |                         |
|-----|--------------------------|----|-------------------------|
| 0   | "goceliak's patch" x4 2c |    |                         |
| 2   | "shift/run" prg          | 33 | "create a disk" prg<    |
| 1   | "-8050 utilities-" usr   | 1  | "--expense calc--" usr  |
| 12  | "disk rx" prg            | 35 | "auto vl" prg           |
| 2   | "8050 copy 0 to 1" prg   | 85 | "rattletrap #1" rel     |
| 7   | "copy disk job q" prg    | 1  | "rattletrap #1" seq     |
| 10  | "copy id0<>idl" prg      | 1  | "----utility----" usr   |
| 15  | "fdc zpage displa" prg   | 1  | "----don't run---" rg?  |
| 5   | "move block 8050" prg    | 1  | "--these alone---" el?  |
| 6   | "change bytes t/s" prg   | 3  | "print screen" prg      |
| 6   | "change bytes hex" prg   | 1  | "m/l print screen" prg  |
| 1   | "seek" prg               | 7  | "gotcha" prg            |
| 4   | "blinky #3" prg          | 1  | "track1-sector2" usr    |
| 4   | "test" prg               | 1  | "track1-sector3" usr    |
| 3   | "b-e 1-2" prg            | 1  | "--more utility--" usr  |
| 1   | "manual copier" prg      | 1  | "---ok to run----" eq?  |
| 3   | "disk t&s dump" prg      | 1  | "basic print scrn" prg  |
| 4   | "20 error" prg           | 3  | "read error test" prg   |
| 3   | "24 error" prg           | 7  | "basic time demo" prg   |
| 1   | "8050 rapid off" prg     | 1  | "-----" usr             |
| 3   | "move head" prg          | 1  | "--del file demo-" usr  |
| 2   | "ds\$ experiment" prg    | 1  | "-----" usr             |
| 513 | "--discourse on--" usr<  | 4  | "duplicate disk" del    |
| 1   | "----the 8050----" usr<  | 1  | "-----" usr             |
| 43  | "investigations#1" seq   | 1  | "---duplicate ---" usr< |
| 19  | "investigations2a" seq   | 1  | "-file name demo-" usr< |
| 24  | "mpi8050 drv swap" seq   | 1  | "-----" usr             |
| 35  | "85 track 8050" seq      | 1  | "goceliak" seq<         |
| 1   | "--90track test--" usr   | 1  | "goceliak" seq<         |
| 6   | "superseek test90" prg   | 1  | "-----" usr             |
| 1   | "---real stuff---" usr<  | 1  | "----sound &----" usr   |
| 1   | "85 track mode" prg      | 8  | "modem tones" prg       |
| 1   | "--8050 charts---" usr   | 8  | "tone generator" prg    |
| 12  | "8050 direct tabl" prg   | 14 | "morse tutor" prg       |
| 24  | "8050 page \$10XX" seq   | 28 | "happy birthday &" prg  |
| 9   | "fdc ram page 0" prg     | 1  | "--patched games-" usr  |
| 9   | "8050 ip page \$00" prg  | 1  | "20 bumps" prg          |
| 4   | "8050 riot address" prg  | 54 | "labyrinth" prg         |
| 1   | "--disassembly---" usr   | 27 | "lunar lander" prg      |
| 25  | "fdc \$05 start up" seq  | 21 | "super lander" prg      |
| 122 | "fdc assy file+" seq     | 15 | "beethoven 1.4" prg     |
| 85  | "fdc assy file" seq      | 18 | "dominoes" del          |
| 1   | "----dumps to----" usr   | 1  | "----wordpro3----" usr  |

|     |                    |      |     |                    |     |
|-----|--------------------|------|-----|--------------------|-----|
| 1   | "--bank 15 cart--" | usr  | 22  | "wordpro3"         | prg |
| 5   | "fdc rom(-32768)+" | prg  | 1   | "--instructions--" | usr |
| 97  | "8050rom(-32768)+" | prg  | 130 | "instructions"     | seq |
| 1   | "---customized---  | usr  | 58  | "instructions #2"  | seq |
| 1   | "-disassemblers--" | usr  | 1   | "instructions #3"  | seq |
| 19  | "8050 rom dis"     | prg  | 1   | "---seq reader---  | usr |
| 19  | "fdc rom dis"      | prg  | 12  | "seq read/print"   | prg |
| 19  | "my dis no zpage"  | prg  | 1   | "----patched ----" | usr |
| 3   | "8050 rom dumper"  | prg  | 1   | "---disk pgms---   | usr |
| 1   | "----8050 ip ----" | usr  | 16  | "disk doctor"      | prg |
| 13  | "quicklook 8050ip" | seq  | 16  | "disk doctor dr 1" | prg |
| 1   | "--secure files--" | usr  | 14  | "superdirectory"   | prg |
| 11  | "security lock"    | prg< | 6   | "sfd view bam"     | prg |
| 10  | "file type change" | prg< | 24  | "rename disk"      | prg |
| 11  | "drink of lethe"   | prg< | 14  | "display t&s"      | prg |
| 1   | "lock disk"        | prg< | 13  | "l drv copy files" | prg |
| 5   | "disksave protect" | prg< | 13  | "disk logger"      | prg |
| 5   | "disksave unlock"  | prg< | 11  | "basic cross ref"  | prg |
| 1   | "---encrypted ---" | usr  | 1   | "-----tests-----"  | usr |
| 513 | "-----files -----" | usr  | 11  | "sfd perf chk"     | prg |
| 7   | "decrypt"          | prg< | 8   | "sfd also chk"     | prg |
| 5   | "encrypted msg100" | seq< | 27  | "printer demo"     | prg |
| 6   | "no key crypt"     | seq< | 9   | "printer test"     | prg |
| 6   | "encrypt"          | prg< | 7   | "sfd auto test"    | prg |
| 1   | "---8050 disks---  | usr  | 4   | "search disk"      | prg |
| 1   | "---on the sfd---  | usr  | 1   | "--demo cartoon--" | usr |
| 18  | "convert disk.ins" | seq  | 11  | "disk demo"        | prg |
| 1   | "---b-128 forms--- | usr  | 6   | "disk display"     | prg |
| 6   | "esc key printout" | prg  | 3   | "disk comm1"       | prg |
| 15  | "superchart"       | prg  | 2   | "disk comm2"       | prg |
| 12  | "zero page blank"  | prg  | 3   | "disk comm3"       | prg |
| 1   | "-----"            | usr  | 4   | "disk write"       | prg |
| 1   | "hide frm ms.deal" | usr  | 4   | "disk read"        | prg |
| 1   | "-----"            | usr  | 2   | "disk overlays"    | prg |
| 3   | "function keys+"   | prg  | 4   | "disk dir"         | prg |
| 1   | "--printer cmds--" | usr  | 6   | "data file"        | seq |
| 4   | "8023 sa"          | prg  | 1   | "test file"        | seq |
| 2   | "4023 sa"          | prg  | 1   | "--descriptions--" | usr |
| 1   | "++++++++++++++++" | usr< | 31  | "descriptions"     | seq |
| 1   | "----freebie ----" | usr< | 27  | "!!--NOTICE--!!"   | seq |
| 1   | "++++++++++++++++" | usr< | 0   | blocks free.       |     |

### CBUG - Norman's utilities vl.2

Diese ist eine der ersten CBUG-Disketten, zusammengestellt vom Vorsitzenden Norman Deltzke. Die Titel erklären sich weitgehend selbst. Viel Programme sind für den 8032 geschrieben, laufen aber mit dem 8432-Emulator.

|    |                    |    |     |    |                    |     |
|----|--------------------|----|-----|----|--------------------|-----|
| 0  | "NORMs UTIL vl.2 " | 86 | 2c  | 6  | "modem connect"    | prg |
| 3  | "Start8432.21"     |    | prg | 2  | "terminal"         | prg |
| 2  | "t1"               |    | prg | 9  | "wat comm"         | prg |
| 2  | "td.alt"           |    | prg | 5  | "pet logger"       | prg |
| 2  | "t2"               |    | prg | 23 | "pterm"            | prg |
| 2  | "td"               |    | prg | 47 | "8010 modem drivr" | prg |
| 81 | "8432.21"          |    | prg | 11 | "terminal.ieee"    | prg |

|    |                    |     |    |                    |     |
|----|--------------------|-----|----|--------------------|-----|
| 4  | "bank F.20 '1024"  | prg | 11 | "terminal.rs232"   | prg |
| 4  | "instructions8432" | prg | 6  | "old term.ieee"    | prg |
| 1  | "bml"              | prg | 6  | "old term.rs232"   | prg |
| 1  | "bm2"              | prg | 27 | "term.vl3"         | prg |
| 1  | "bm3"              | prg | 17 | "term.rs232"       | prg |
| 1  | "bm4"              | prg | 7  | "term.ieee"        | prg |
| 1  | "bm5"              | prg | 11 | "64 term"          | prg |
| 1  | "bm6"              | prg | 17 | "term 8032 v7"     | prg |
| 1  | "bm7"              | prg | 28 | "term inst.wp 1"   | prg |
| 1  | "bm8"              | prg | 20 | "term inst.wp 2"   | prg |
| 1  | "-mon extension-   | prg | 71 | "rita"             | prg |
| 3  | "extra mn instr"   | prg | 1  | " "                | seq |
| 4  | "extra monitor"    | prg | 1  | " "                | seq |
| 35 | "extra monitor.s"  | prg | 3  | "READ MEMO"        | seq |
| 7  | "list-me.bu"       | prg | 25 | "bl28 xterm orig"  | prg |
| 1  | "sample boot.b"    | prg | 26 | "cis dload xterm"  | prg |
| 2  | "txram obj"        | prg | 1  | " "                | seq |
| 15 | "sample obj"       | prg | 1  | " "                | seq |
| 4  | "test.basic"       | prg | 31 | "MOVERS & SHAKERS" | seq |
| 3  | "hex dumper.z"     | prg | 1  | " "                | seq |
| 1  | "flash.bas.b"      | prg | 1  | " "                | seq |
| 1  | "flash.obj"        | prg | 1  | "JIM WHITE UTILIT" | seq |
| 7  | "flash.8.1"        | prg | 8  | "labels.p"         | seq |
| 11 | "chg ld addr vl.2" | prg | 7  | "hlabels"          | seq |
| 2  | "mem dump.b"       | prg | 14 | "shipping labels"  | prg |
| 1  | "err\$ list.b"     | prg | 7  | "3 sorts"          | prg |
| 1  | " "                | seq | 9  | "text-bl28"        | prg |
| 1  | " "                | seq | 19 | "name sort"        | prg |
| 13 | "IMPORTANT NOTE"   | seq | 1  | "bl28 loader"      | prg |
| 4  | "8050 DEMO DISC"   | seq | 9  | "disk directory"   | prg |
| 38 | "performance"      | prg | 15 | "b256 labels"      | prg |
| 5  | "dos support"      | prg | 7  | "music 1.4"        | prg |
| 10 | "copy/all"         | prg | 14 | "beethoven 1.3"    | prg |
| 7  | "change unit add"  | prg | 7  | "happy birthday"   | prg |
| 22 | "printer demo"     | prg | 47 | "b index-sup/scpt" | seq |
| 38 | "display trk&sec"  | prg | 2  | "CREDITS"          | seq |
| 10 | "basic cross ref"  | prg | 1  | "---071585-----"   | seq |
| 16 | "8250 expand rel"  | prg | 1  | "WHITE UPDATE"     | seq |
| 1  | " "                | seq | 16 | "rev ship labels"  | prg |
| 1  | " "                | seq | 1  | "---speed---       | seq |
| 2  | "TPUG C1 COMMUNIC" | seq | 27 | "format speed err" | seq |
| 1  | "signon"           | prg | 1  | "-----cp-----"     | seq |
| 13 | "basic aid"        | prg | 28 | "256K upgrade"     | seq |
| 2  | "exec hello"       | prg | 29 | "bl28 expansion"   | prg |
| 1  | "hello"            | prg | 1  | "-----mg-----"     | seq |
| 5  | "universal wedge"  | prg | 4  | "line plot"        | prg |
| 11 | "tnw answr demo"   | prg | 4  | "plot"             | prg |
| 14 | "morse tutor"      | prg | 11 | "double column"    | seq |
| 4  | "morse-bttrfld"    | prg | 2  | "load run instr"   | prg |
| 11 | "tnw488/103a"      | prg | 2  | "loading instruct" | seq |
| 2  | "tele dialer"      | prg | 1  | "-----rl-----"     | seq |
| 23 | "pc.net.modem"     | prg | 29 | "datafile"         | prg |
| 6  | "cbm 8010"         | prg | 29 | "dfmail"           | prg |
| 4  | "ascii-wpro"       | prg | 29 | "dfreport"         | prg |
| 1  | "pet to pet"       | prg | 9  | "bargraph.rl"      | prg |
| 3  | "modem pet-ascii"  | prg | 13 | "invoicer.c"       | prg |
| 5  | "logger"           | prg | 50 | "lost gold adv"    | prg |
| 5  | "modem menu"       | prg | 2  | "df example"       | seq |

|    |                    |     |      |                    |     |
|----|--------------------|-----|------|--------------------|-----|
| 2  | "log to ascii prt" | prg | 1    | "<-instructions->" | prg |
| 1  | "rcv#4"            | prg | 3    | "Dbargraph.rl"     | seq |
| 4  | "pet clist"        | prg | 3    | "Dinvoicer.c"      | seq |
| 4  | "build hello"      | prg | 3    | "Dlost gold adv"   | seq |
| 23 | "src.ex hello.asm" | prg | 7    | "Ddatafile"        | seq |
| 38 | "expansion pres"   | prg | 3    | "Ddfmail"          | seq |
| 32 | "comm primer"      | prg | 3    | "Ddfreport"        | seq |
| 25 | "freq generator!"  | prg | 27   | "!!--NOTICE--!!!"  | seq |
| 5  | "modem log"        | prg | 18   | "NORMs UTIL vl.2"  | seq |
| 2  | "bulletin"         | prg | 1917 | blocks free.       |     |

### CBUG - Pre-Releases 1

Die Titel sind größtenteils einigermaßen aussagefähig. Die Programme dieser Vorveröffentlichung sind nicht vollständig ausgetestet. Eine Fundgrube!

|    |                    |         |    |                    |     |
|----|--------------------|---------|----|--------------------|-----|
| 0  | "pre-releasel      | " pl 2c | 41 | "B-Address File"   | prg |
| 3  | "preamble"         | seq     | 22 | "Medical Survey"   | prg |
| 3  | "Liz Deal's 2"     | seq     | 37 | "B-Labyrinth"      | prg |
| 1  | "<---tools--->"    | prg     | 10 | "B-PayCheck Anal"  | prg |
| 11 | "disk logger.bl28" | prg     | 23 | "G-Solitaire"      | prg |
| 1  | "disklog.400"      | prg     | 20 | "G-Black Sunday"   | prg |
| 8  | "disk view.bl28"   | prg     | 17 | "G-Concentration"  | prg |
| 1  | "<-----p----->"    | prg     | 13 | "G-Deflection"     | prg |
| 11 | "reader/writer"    | prg     | 69 | "Account Pac"      | prg |
| 1  | "<-----6----->"    | prg     | 2  | " directory "      | seq |
| 2  | "d.a/p gif"        | seq     | 14 | "instr word machi" | seq |
| 9  | "d.vendor file"    | seq     | 7  | "instr name machi" | seq |
| 4  | "d.check file"     | seq     | 2  | "instr b-x modem " | seq |
| 12 | "d.a/p inv file"   | seq     | 1  | "B-memory peeker"  | seq |
| 1  | "d.a/p inv ctr"    | seq     | 2  | "Name List"        | seq |
| 1  | "<-----i----->"    | prg     | 3  | "b function key"   | seq |
| 1  | "-----ws-----"     | seq     | 1  | "-----gv-----"     | seq |
| 6  | "ws info"          | seq     | 2  | "gv info"          | seq |
| 1  | "+++GAMES+++"      | prg     | 8  | "amaz grace bl28"  | prg |
| 47 | " star trek 1"     | prg     | 22 | "bl28 address"     | prg |
| 52 | " star trek 2"     | prg     | 6  | "bl28 harmonica"   | prg |
| 4  | "^star trek 1"     | rel     | 26 | "bl28 terminal"    | prg |
| 4  | "^star trek 2"     | rel     | 13 | "calendar bl28"    | prg |
| 1  | "+++UTILITIES+++"  | prg     | 6  | "pirate adventure" | prg |
| 24 | "!copy:"           | prg     | 14 | "disklook bl28"    | prg |
| 3  | "!transfer"        | prg     | 19 | "diskmgr bl28"     | prg |
| 11 | "!file copy:"      | prg     | 34 | "file control 128" | prg |
| 9  | "!clean:"          | prg     | 5  | "loan bl28"        | prg |
| 11 | "!rescue:"         | prg     | 25 | "mlborne bl28"     | prg |
| 1  | "+++TERMINAL+++"   | prg     | 38 | "nuke bl28"        | prg |
| 26 | " terminal"        | prg     | 23 | "ruler bl28"       | prg |
| 5  | "!term"            | prg     | 25 | "therapy bl28"     | prg |
| 1  | "+++SOME DOCS+++"  | prg     | 12 | "yahtzee bl28"     | prg |
| 14 | " tokens"          | seq     | 1  | "-----"            | seq |
| 19 | " cursor config."  | seq     | 1  | " 8432 EMULATOR "  | seq |
| 29 | " program lines"   | seq     | 1  | " PRGMS. FOLLOW "  | seq |
| 21 | "backu!l"          | prg     | 1  | " "                | seq |
| 1  | "-----tc-----"     | seq     | 86 | "ohare adventure"  | prg |
| 3  | "tc info"          | seq     | 71 | "dog.star.adven"   | prg |

|    |                    |     |      |                    |     |
|----|--------------------|-----|------|--------------------|-----|
| 8  | " Menu"            | prg | 60   | "wizard's.castle"  | prg |
| 8  | " Menu"            | prg | 60   | "king.tut"         | prg |
| 8  | " Menu"            | prg | 58   | "trip to atlantis" | prg |
| 35 | "B-Name Machine"   | prg | 12   | "hammurabi"        | prg |
| 39 | "B-Word Machine"   | prg | 20   | "fishery"          | prg |
| 23 | "B-Copy Calc"      | prg | 15   | "cassette labeler" | prg |
| 23 | "B-Invoice Prog"   | prg | 30   | "disk master v2"   | prg |
| 22 | "C-Mail List"      | prg | 28   | "yahtzee"          | prg |
| 29 | "B-SoftBall Stats" | prg | 3    | "un-dload easier"  | prg |
| 11 | "U-Load Address"   | prg | 3    | "dload easier"     | prg |
| 1  | "reset.1"          | prg | 15   | "backup 2031"      | prg |
| 1  | "reset.5"          | prg | 18   | "towers of hanoi"  | prg |
| 12 | "B-Function Key"   | prg | 20   | "3d tic tac toe"   | prg |
| 1  | "test.1"           | prg | 1    | "-----lc-----"     | seq |
| 1  | "test.5"           | prg | 3    | "lc rs232"         | seq |
| 25 | "B-X Term Program" | prg | 26   | "cooper RS232"     | seq |
| 8  | "B-Memory Peeker"  | prg | 1781 | blocks free.       |     |

### CBUG - TPUG I

Dies ist eine Clubdiskette der Toronto-Pet-User's-Group", der größten Commodore-Geräte-Benutzergruppe überhaupt. Sicher benutzt jeder von uns viele Programme, die von dort kommen, insbesondere von ihrem wohl bekanntesten Mitglied Jim Butterfield, auf dessen Arbeit sogar ein großer Teil unserer Clubprogramme basiert.

Die Programme sind original für den 8032 geschrieben.

|    |                    |    |     |    |                    |     |
|----|--------------------|----|-----|----|--------------------|-----|
| 0  | "CBUG TPUG #1 v1." | 91 | 2c  |    |                    |     |
| 2  | "----OX6----       |    | seq | 3  | "Deasy edit/mx-80" | seq |
| 15 | "easy edit/tape"   |    | prg | 3  | "Deasy.edit.mx-82" | seq |
| 16 | "easy edit/disk"   |    | prg | 4  | "Dword pro th2058" | seq |
| 14 | "easy edit/c"      |    | prg | 3  | "Dread wp2040"     | seq |
| 18 | "easy edit/mx-80"  |    | prg | 2  | "Dwp tape2058"     | seq |
| 17 | "easy.edit.mx-82"  |    | prg | 2  | "Dread wp tape204" | seq |
| 9  | "page1 eze edit"   |    | seq | 3  | "D8050 all"        | seq |
| 9  | "page2 eze edit"   |    | seq | 3  | "D4040 all"        | seq |
| 9  | "page3 eze edit"   |    | seq | 2  | "Dwp.all"          | seq |
| 10 | "page4 eze edit"   |    | seq | 2  | "De.g set-up 80co" | seq |
| 10 | "page5 eze edit"   |    | seq | 2  | "Dv all"           | seq |
| 7  | "page6 eze edit"   |    | seq | 1  | "----PXb----       | seq |
| 27 | "word pro th2058"  |    | prg | 2  | "screen print"     | prg |
| 27 | "read wp2040"      |    | prg | 53 | "file maker"       | prg |
| 6  | "hi memory106"     |    | prg | 2  | "A"                | seq |
| 13 | "instructions#7"   |    | prg | 2  | "B"                | seq |
| 10 | "word p demo"      |    | prg | :  |                    |     |
| 27 | "wp tape2058"      |    | prg | :  |                    |     |
| 27 | "read wp tape2040" |    | prg | 2  | "Z"                | seq |
| 24 | "tax 81 ont v1.0"  |    | prg | 8  | "screen"           | prg |
| 12 | "graph.printer"    |    | prg | 2  | "home row"         | seq |
| 17 | "graphix instr"    |    | prg | 2  | "all rows"         | seq |
| 21 | "graphics-loader"  |    | prg | 2  | "G"                | seq |
| 22 | "graphics-demo"    |    | prg | 2  | "flash"            | seq |
| 23 | "mail list 4040"   |    | prg | 8  | "file maker ii"    | prg |
| 41 | "4040 update"      |    | prg | 2  | "high wrist"       | seq |

|    |                    |     |      |                    |     |
|----|--------------------|-----|------|--------------------|-----|
| 5  | "4040 format"      | prg | 2    | "low wrist"        | seq |
| 25 | "4040 print"       | prg | 2    | "shift lock"       | seq |
| 3  | "4040 menu"        | prg | 2    | "# sign"           | seq |
| 23 | "mail list 8050"   | prg | 2    | "\$ sign"          | seq |
| 6  | "8050 format"      | prg | 2    | "% sign"           | seq |
| 42 | "8050 update"      | prg | 6    | "Numbers 1-5"      | seq |
| 3  | "8050 menu"        | prg | 6    | "Numbers 6-0"      | seq |
| 25 | "8050 print"       | prg | 3    | "Each Numeral"     | seq |
| 5  | "v40.2.b"          | prg | 4    | "Alpha Progress"   | seq |
| 5  | "v40.4.b"          | prg | 66   | "dum 3.4"          | prg |
| 5  | "v80.4.b"          | prg | 53   | "keyboarding"      | prg |
| 12 | "e.g set-up 80col" | prg | 3    | "cover"            | seq |
| 41 | "wp3/4 inst1"      | prg | 2    | "list-me pxb.1"    | prg |
| 15 | "wp3/4 inst2"      | prg | 1    | "----docs----      | prg |
| 1  | "<-instructions->" | prg | 4    | "Dfile maker ii"   | seq |
| 3  | "Dgraph.all"       | seq | 4    | "Dscreen print"    | seq |
| 3  | "Dgraph.printer"   | seq | 4    | "Dfile maker"      | seq |
| 3  | "Dtax 81 ont vl.0" | seq | 5    | "Dkeyboarding"     | seq |
| 2  | "Deasy edit/tape"  | seq | 2    | "Dlist-me pxb.1"   | seq |
| 3  | "Deasy edit/disk"  | seq | 2    | "Ddum 3.4"         | seq |
| 3  | "Deasy edit/c"     | seq | 27   | "!!!--NOTICE--!!!" | seq |
| 3  | "Dpage.all"        | seq | 3228 | blocks free.       |     |

### CBUG - Swam's Utility #1

Dies ist eine Programmsammlung, die jedem etwas bietet. Dateien, mit dem Präfix "&" sind sequentielle Textdateien. So umfangreiche Dokumentationen wären für unsere Disketten auch wünschenswert.

Einige der wichtigsten Programme auf dieser Diskette sind original für unsere Geräte geschrieben und nicht von älteren Maschinen angepaßt.

|     |                    |         |    |                    |     |
|-----|--------------------|---------|----|--------------------|-----|
| 0   | "SWANS UTIL #1     | " 88 2c | 1  | "<- demos/math ->" | prg |
| 29  | " start up":       | prg     | 29 | " lifetime":       | prg |
| 1   | "<-- articles -->" | prg     | 10 | " calculator":     | prg |
| 34  | "& file prefixes"  | seq     | 2  | " factors":        | prg |
| 36  | "&&file prefixes"  | seq     | 13 | " regression":     | prg |
| 18  | "& cursor config." | seq     | 3  | " squirm":         | prg |
| 18  | "&&cursor config." | seq     | 1  | " 4 fill 6"        | prg |
| 27  | "& program lines"  | seq     | :  |                    |     |
| 27  | "&&program lines"  | seq     | 1  | " 4 fill 20"       | prg |
| 1   | " 1 fill 7"        | prg     | 1  | "<- utilities -->" | prg |
| 1   | " 1 fill 8"        | prg     | 45 | "!futility":       | prg |
| :   |                    |         | 1  | " /count"          | prg |
| 1   | " 1 fill 18"       | prg     | 34 | "!copy":           | prg |
| 1   | "<-instructions->" | prg     | 3  | " /transfer"       | prg |
| 97  | "& star trek"      | seq     | 5  | " rename disk":    | prg |
| 106 | "&&star trek"      | seq     | 34 | " archive 1":      | prg |
| 20  | "& eliza"          | seq     | 1  | " /count & copy"   | prg |
| 22  | "&&eliza"          | seq     | 12 | ".get filenames"   | prg |
| 29  | "& calculator"     | seq     | 1  | ".get mach lang"   | prg |
| 31  | "&&calculator"     | seq     | 1  | " 5 fill 10"       | prg |
| 19  | "& regression"     | seq     | :  |                    |     |
| 22  | "&&regression"     | seq     | 1  | " 5 fill 20"       | prg |
| 26  | "& archive 1"      | seq     | 1  | "<-- graphics -->" | prg |

|    |                    |     |     |                    |     |
|----|--------------------|-----|-----|--------------------|-----|
| 28 | "&&archive 1"      | seq | 20  | " large letter":   | prg |
| 26 | "& get filenames"  | seq | 14  | " calendar":       | prg |
| 31 | "&&get filenames"  | seq | 8   | " s-enterprise":   | prg |
| 4  | "& get mach lang"  | seq | 4   | " p-enterprise":   | prg |
| 5  | "&&get mach lang"  | seq | 8   | " bouncing ball":  | prg |
| 17 | "& large letter"   | seq | 1   | " 6 fill 6"        | prg |
| 19 | "&&large letter"   | seq |     | :                  |     |
| 58 | "& terminal"       | seq | 1   | " 6 fill 20"       | prg |
| 62 | "&&terminal"       | seq | 1   | "<communication->" | prg |
| 1  | " 2 fill 19"       | prg | 26  | " terminal":       | prg |
| 1  | " 2 fill 20"       | prg | 5   | " /term"           | prg |
| 1  | "<--- games ---->" | prg | 11  | " obj-to-prg":     | prg |
| 52 | " star trek 2":    | prg | 7   | " prg-to-obj":     | prg |
| 47 | " star trek 1":    | prg | 2   | "-----df-----"     | seq |
| 4  | "^star trek 2"     | rel | 59  | "backup2"          | prg |
| 4  | "^star trek 1"     | rel | 7   | "file pursuit"     | prg |
| 8  | " eliza":          | prg | 22  | "backup1"          | prg |
| 17 | "&eliza data"      | seq | 1   | "-----docs-----"   | prg |
| 17 | " eliza data prgm" | prg | 8   | "backup1.doc"      | seq |
| 23 | " iq test":        | prg | 9   | "backup2.doc"      | seq |
| 1  | "&iqtest data"     | seq | 4   | "file pursuit.doc" | seq |
| 8  | " craps":          | prg | 8   | "why backup"       | seq |
| 1  | " 3 fill 11"       | prg | 8   | "backup strategy"  | seq |
|    | :                  |     | 27  | "!!!--NOTICE--!!!" | seq |
| 1  | " 3 fill 20"       | prg | 537 | blocks free.       |     |

### CBUG - financial analysis disk 1

Eine so umfangreiche Programmsammlung zu einem einzigen Thema hat gewiß Seltenheitswert.  
Vielleicht übersetzt jemand wenigstens diese Diskettendokumentation mal ins Deutsche...

|    |            |                                                      |
|----|------------|------------------------------------------------------|
| 15 | "bondval2" | ratio of change of bond value.                       |
| 20 | "deprdb"   | declining balance depreciation schedule.             |
| 14 | "discbal"  | true interest on discounted loan.                    |
| 17 | "dilanal"  | analysis of merging companies stock values.          |
| 14 | "markup"   | gross margins on costs.                              |
| 13 | "loanaffd" | finds loan amount available interest and incomebased |
| 18 | "valprinf" | probability based on nature using user input.        |
| 12 | "fgeq"     | maximum quantity orders based on one price.          |
| 13 | "rentprch" | value of investment in real estate.                  |
| 17 | "expval"   | expected future value of an investment.              |
| 21 | "deprsf"   | depreciation using sinking fund schedule.            |
| 18 | "compbal"  | compensating loan interest and balance.              |
| 15 | "rule78"   | interest change on loan using rule of 78.            |
| 21 | "sinkfund" | sinking fund and tables.                             |
| 10 | "merganal" | value of acquired company.                           |
| 11 | "prindpa"  | paasche price index.                                 |
| 13 | "npv"      | net present value of project.                        |
| 20 | "optloss"  | opportunity loss table.                              |
| 54 | "acctrec"  | accounts receivable with no instructions.            |
| 21 | "utility"  | computes decision on uncertain actions.              |
| 16 | "envelope" | prints envelopes.                                    |
| 26 | "annul"    | annuity computation.                                 |
| 20 | "bondval"  | evaluation formula for bonds.                        |



22 "timeclock" calculates employee hrs and overtime.  
 23 "check2" nebs check print.  
 44 "checkbkl" checkbook maintenance.  
 15 "upszone" outputs ups zone.  
 29 "trans" determines method of least expensive shipping.  
 17 "sharpe1" helps determine good stock selection.  
 18 "condprof" conditional profit table.  
 30 "mailpac" mailing program.  
 19 "deprddb" depreciation using double declining balance.  
 44 "autoexp" expense records for auto.  
 46 "arbcamp" arbitrage computations.  
 16 "leaseint" lease interest rate.  
 27 "saleleas" sale leaseback analysis.  
 7 "invoice" set up invoices.  
 32 "finrat" financial ratios for business.  
 40 "insfile" file for insurance information.  
 46 "invent2" inventory program.  
 18 "deprsy" depreciation schedule for sum of digits.  
 16 "optwrite" development system for option writing.  
 32 "letwrt" simple word processor.  
 18 "deprsl" straight line depreciation schedule.  
 19 "mortgage/a" prints mortgage repayment table.  
 21 "timusan" summary of time use for a period.  
 54 "acctpay" accounts payable no instructions.  
 45 "busbud" bookkeeping and budget analysis.  
 26 "teldir" telephone directory.  
 44 "portval9" stock portfolio program.  
 60 "payroll2" payroll program.  
 10 "fval" future value of an investment.  
 10 "pval" present value of an investment.  
 10 "date" number of days between 2 dates.  
 11 "blacksh" black sholes stock valuation formula.  
 13 "annudf" present value of a deferred annuity.  
 18 "assign" figures cost of assignment.  
 12 "capl" capital asset pricing model of analysis.  
 15 "betaalph" computes alpha and beta values of stock with ret.  
 10 "eq" economic order quantity analysis.  
 10 "breakevn" computes breakeven on sales.  
 12 "dateval" maturity of an obligation that is a dated value.  
 11 "epest" estimate of expected earnings per share.  
 11 "fgeowsh" invent. theory fixed quantity eq with shortages.  
 9 "loanpay" calculates amount of loan payment.  
 13 "fuprinf" gives future price with current inflation.  
 9 "paynet" gross pay needed for a given net.  
 11 "prindlas" price comparison base year to current.  
 12 "profind" profitability index of a product.  
 16 "queuecb" waiting line cost analysis.  
 12 "rrconvbd" convertible bond rate of return.  
 32 "rtval" value of a stock right.  
 15 "bayes" probability of occurrence in state of nature.  
 11 "warval" computes value of a warrant.  
 13 "wacc" weighted average cost of capital.  
 13 "valadinf" computes bounds on sampling variance.  
 19 "simplex" linear analysis of optimum use of resources.  
 10 "labell" shipping label printing routine.  
 10 "rrconst" rate of return on investment with constant inflow  
 18 "rrvarin" rate of return on investment with variable inflow  
 8 "salvage" salvage value of an item.

|    |             |                                                      |
|----|-------------|------------------------------------------------------|
| 10 | "stockvall" | computes value of a share of stock.                  |
| 13 | "termisny"  | comperison of methods for payment of purchases.      |
| 14 | "timenov"   | simple moving average smoothing program.             |
| 14 | "timetr"    | computes a linear trenl for a time series.           |
| 12 | "taxdep"    | computes net effect on income from depreciation.     |
| 8  | "effect"    | calculates an effective interest rate on loan.       |
| 14 | "ncfanal"   | net cash flow analysis utilising depreciation.       |
| 15 | "queuel"    | waiting line cost analysis #2.                       |
| 16 | "multmon"   | computes compounding of interest.                    |
| 16 | "seasind"   | seasonal index for a series of sales periods.        |
| 9  | "sellpr"    | figures selling price given total sale & tax rate.   |
| 10 | "regwith"   | amount available for withdrawal over a fixed period. |
| 17 | "fgeogpb"   | invent. theory fixed eoq with quantity price breaks. |
| 14 | "simpdisk"  | computes discount, list price or sale price.         |

189 blocks free.

### CBUG - Entertainment

Dies ist eine Werbediskette des CBUG. Hier finden sich ein paar ganz hübsche Sachen, u. a. aus dem 700-Sampler.

|    |                |   |       |    |                   |     |
|----|----------------|---|-------|----|-------------------|-----|
| 0  | "Entertainment | " | GS 2c | 28 | "F-KEY template"  | prg |
| 5  | "loader"       |   | prg   | 7  | "happy birthday"  | prg |
| 8  | "boot"         |   | prg   | 45 | "disk utilities"  | prg |
| 8  | "boot2"        |   | prg   | 1  | " directory "     | seq |
| 48 | "loader2"      |   | prg   | 6  | "banner inst"     | prg |
| 5  | "sys0400"      |   | prg   | 8  | "music3"          | prg |
| 8  | "boot3"        |   | prg   | 48 | "loader2.1"       | prg |
| 8  | "boot4"        |   | prg   | 67 | "banner"          | prg |
| 8  | "boot5"        |   | prg   | 57 | "solitaire"       | prg |
| 8  | "boot6"        |   | prg   | 63 | "checkers"        | prg |
| 2  | "loader3"      |   | prg   | 52 | "z16"             | prg |
| 8  | "boot7"        |   | prg   | 5  | "Directory .dok": | prg |
| 6  | " menu"        |   | prg   | 3  | "directory .dok": | seq |
| 8  | "boot8"        |   | prg   |    | 3613 blocks free. |     |

### CBUG - 700 sampler (demos)

Diese Diskette ist ziemlich nutzlos. Es handelt sich hier um eine Werbediskette von Commodore in England aus der Zeit, da man dort noch mit einer großen Zukunft der 600er und 700er gerechnet hat.

Demos der Qualität, wie sie hier sehen sind, sucht man auf unseren Computern sonst vergeblich.

|    |              |   |       |    |                 |     |
|----|--------------|---|-------|----|-----------------|-----|
| 0  | "700 sampler | " | aa 2c |    |                 |     |
| 1  | " loader"    |   | prg   | 8  | "pet ii screen" | prg |
| 8  | " boot2"     |   | prg   | 7  | "network"       | prg |
| 8  | " bootscrn"  |   | prg   | 9  | "logo"          | prg |
| 9  | " copy"      |   | prg   | 29 | "ast.demo"      | prg |
| 8  | " instscrn"  |   | prg   | 22 | "and.demo"      | prg |
| 6  | " menu"      |   | prg   | 7  | "arr.demo"      | prg |
| 49 | "demo"       |   | prg   | 53 | "lab.demo"      | prg |
| 5  | "menu"       |   | prg   | 27 | "tit.demo"      | prg |

Beeline ist ein universelles Terminal-Programmpaket, das bei den Amerikanern den Rang eines Standards erreicht zu haben scheint. Ich hoffe, daß wir in der nächsten Ausgabe eine deutsche Rezension veröffentlichen können, und daß in absehbarer Zeit auch eine deutsche Anleitung zu haben sein wird. Ich selbst habe mit DFÜ überhaupt nichts am Hut und kann nichts weiter dazu sagen, als daß das Programm ziemlich vollständig erscheint, und daß ihm eine Menüführung, wie ich sie an anderer Stelle in diesem Heft vorgestellt habe, zum Vorteil gereichen würde. Der Preis von DM 61,- scheint gerechtfertigt.

#### CBUG-Commodore Entwicklungsunterlagen

Auf drei Disketten finden sich sämtliche Source-Files, aus denen sich das BASIC und das Betriebssystem unsere 600/700-Rechner assemblieren lassen. Es handelt sich um sequentielle ASCII-Files, der benutzte Assembler ist der von Commodore und befindet sich ebenfalls auf der Diskette. Es ist aber nicht überprüft, ob er mit 8432 auf dem 700 läuft; der Editor tut das jedenfalls nicht.

Zu den Source-Files ist noch zu sagen, daß sie so gut dokumentiert sind, daß eine bessere Dokumentation für die ROMs unserer Rechner nicht zu finden ist. Es muß halt nur ausgedruckt werden.

Das Basic läßt sich so assemblieren, daß sich verschieden viele Bänke nutzen lassen (von 64k - 256k). Auch das 500-Basic (Start in Bank 0) läßt sich erzeugen.

Die Sources für das DIN-Betriebssystem fehlen uns noch, aber wir hoffen auf die Früchte der Zusammenarbeit mit Commodore Deutschland.

Die Directorys der Disketten sind nicht sehr erquicklich und deswegen hier weggelassen.

Unsere Rechner sind mit einiger Sicherheit die einzigen, für die seitens des Herstellers die Entwicklungsunterlagen den Anwendern überlassen worden sind. Der Bezug aber ist an die Mitgliedschaft in unserem Verein und an die Abgabe einer besonderen Erklärung gebunden. Die Weitergabe an Dritte ist ebenso strikt verboten wie die wie auch immer geartete gewerbliche Verwendung.

### CBUG - Swan's Basic Tutotorial

Warren D. Swan, einer der tüchtigsten Autoren von Clubdisketten (siehe auch Swan's Utilities), hat diese umfangreiche Einführung in das Basic des 600/700 verfaßt. Auf der Diskette liegt der ca. 1000 Blöcke umfassende Text einmal als sequentielle Datei und zusätzlich als formatierte Superscript-Datei vor. Wer einigermaßen Englisch, aber noch nicht Basic spricht, dem kann man diese Diskette wärmstens empfehlen.

## Erklärung zum Bezug von Commodore-Unterlagen

Ich erkläre mich mit den Bedingungen für den Erwerb von Original-Commodore-Unterlagen der Chicago B-User-Group, (bezogen über die CBM-Selbsthilfe e.V.) einverstanden und gebe folgende Erklärung ohne Einschränkung oder Zusätze ab:

1) Ich bin Mitglied der CBM-Selbsthilfe e.V. und dadurch auch des CBUG.

2) Ich gedenke, die erwünschten Materialien ausschließlich zu benutzen, um für mich und/oder andere Mitglieder den Nutzen der Computer der Serie CBM 600/700 und deren Peripherie zu mehren.

3) Ich werde den Inhalt der erhaltenen Unterlagen keinen Nicht-Mitgliedern des CBUG bzw. der CBM-Selbsthilfe e.V. überlassen, unter ihnen verbreiten oder für sie kopieren, es sei denn, jemand benötigt sie, um mir in den unter 2) genannten Angelegenheiten weiterzuhelfen.

4) Ich werde im Falle einer gemäß obiger Erlaubnis erfolgten Weitergabe oder Verbreitung dieser Unterlagen dafür sorgen, daß eine unterschriebene Kopie dieser Erklärung an den CBUG oder die CBM-Selbsthilfe e.V. gelangt.

5) Ich bin in Kenntnis gesetzt worden, daß für viele der hier angesprochenen Unterlagen das Urheberrecht bzw. die Eigentumsrechte bei Commodore Business Machines und/oder deren Zulieferern liegt, daß der einzige Zweck der Überlassung dieser Informationen durch CBM an die CBUG-Mitglieder die Erhöhung des Nutzens der CBM-Geräte ist, die die Mitglieder sich angeschafft haben, und daß keine Garantie, Zusicherung der Richtigkeit, Vollständigkeit oder anderer Eigenschaften durch CBM, CBUG oder jemand anderen gegeben wird. Jegliche andere Weitergabe ist ausdrücklich verboten, und ich werde kein solche verbotene Verbreitung vornehmen.

6) Ich stimme zu, daß mit "Unterlagen", wie es hier benutzt ist, alle Formen gemeint sind, in denen die Informationen gespeichert sein können, einschließlich Disketten, gedruckt, kopiert, übertragen oder auf jegliche andere Weise. Hierunter fallen alle Unterlagen, die intern oder extern einen entsprechenden Urheberrechts- oder Eigentumsvermerk tragen.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Vorname

\_\_\_\_\_  
Straße, Nr.

\_\_\_\_\_  
PLZ.

\_\_\_\_\_  
Ort

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Unterschrift

Erklärung zum Bezug von Programmen des Anthony J. Goceliak

Hiermit verpflichte ich mich, über die CBM-Selbsthilfe erhältliche Programme des Mitgliedes der Chicago B-Users Group Anthony J. Goceliak nicht weiterzugeben, insbesondere nicht in den Einflußbereich des Warschauer Vertrages gelangen zu lassen.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Vorname

\_\_\_\_\_  
Straße, Nr.

\_\_\_\_\_  
PLZ.

\_\_\_\_\_  
Ort

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Unterschrift

Info-Formular

Ich bitte um nähere Informationen über den Bezug von ...

- ☐ INFO 1 Typenräder 8028
- ☐ INFO 2 Typenräder 6400
- ☐ INFO 3 EPROM-Programmiersgerät 600/700
- ☐ INFO 4 DIN-Tastenkappen CBM 600/700
- ☐ INFO 5 Computer CBM 510
- ☐ INFO 6 Computer CBM 710
- ☐ INFO 7 Floppy SFD 1001
- ☐ INFO 8 ml01 neues Druckerbetriebssystem für 8028
- ☐ INFO 9 Satzung CBM-Selbsthilfe

(Gewünschtes bitte anzukreuzen)

\_\_\_\_\_  
Name

\_\_\_\_\_  
Vorname

\_\_\_\_\_  
Straße, Nr.

\_\_\_\_\_  
PLZ.

\_\_\_\_\_  
Ort

\_\_\_\_\_  
Telefon

## Ansprechpartner und Mitarbeit

Im nächsten Heft soll eine Liste von Mitgliedern erscheinen, die als Ansprechpartner für bestimmte Fragen zur Verfügung stehen. Dieses Formular soll die Erstellung der Liste erleichtern. Wenn Sie also als Ansprechpartner zur Verfügung stehen wollen, dann füllen Sie bitte das untenstehende Formular aus. Keine Bange, durch die Arbeitsteilung wird die Anfragerei keine große Arbeitsbelastung für den einzelnen bedeuten. Außerdem können Sie angeben, ob Sie sich an konkreten Arbeiten für Clubdisketten beteiligen wollen.

\_\_\_\_\_

Name

\_\_\_\_\_

Vorname

\_\_\_\_\_

Straße, Nr.

\_\_\_\_\_

PLZ.

\_\_\_\_\_

Ort

\_\_\_\_\_

Telefon

Beschränkungen bei telefonischen Anfragen:

\_\_\_\_\_

Wochentag(e)

\_\_\_\_\_

Uhrzeit von - bis

Für folgende Themen stehe ich zur Verfügung:

- |                                                                 |                                                   |                                        |
|-----------------------------------------------------------------|---------------------------------------------------|----------------------------------------|
| <input type="checkbox"/> Basic                                  | <input type="checkbox"/> Assembler                | <input type="checkbox"/> EMA           |
| <input type="checkbox"/> Macro-Basic                            | <input type="checkbox"/> Basic-Firmware           | <input type="checkbox"/> DFÜ           |
| <input type="checkbox"/> Kernal- Editor-Firmware                | <input type="checkbox"/> Übertragung CBM <--> IBM |                                        |
| <input type="checkbox"/> 8032                                   | <input type="checkbox"/> 8096/8296                | <input type="checkbox"/> 8432-Emulator |
| <input type="checkbox"/> Hardware Floppies                      | <input type="checkbox"/> Floppies soft            |                                        |
| <input type="checkbox"/> 600/700 Hardware                       | <input type="checkbox"/> 8032 Hardware            | <input type="checkbox"/> Protext       |
| <input type="checkbox"/> Stelle 8000-Software bereit            | <input type="checkbox"/> Profitext                |                                        |
| <input type="checkbox"/> Übernehme CBUG-Clubdisk-"Patenschaft": |                                                   |                                        |

-----

(Gewünschte Diskette oder Gebiet)



# Beitrittserklärung

\_\_\_\_\_  
Name

\_\_\_\_\_  
Vorname

\_\_\_\_\_  
Straße, Nr.

\_\_\_\_\_  
PLZ.

\_\_\_\_\_  
Ort

\_\_\_\_\_  
Telefon

Hiermit bitte ich um Aufnahme in den Verein CBM-Selbsthilfe e.V.

Den Mitgliedsbeitrag für das laufende Jahr von DM 50.- habe ich:

☐ Überwiesen,                    ☐ einen Scheck beigelegt.

(Zutreffendes bitte ankreuzen)

Die folgenden jährlichen Mitgliedsbeiträge (ebenfalls DM 50.-) werde ich bis zum 31. Dez. des Vorjahres dem Verein zukommen lassen, es sei denn, ich kündige meine Mitgliedschaft vor dem 31. Nov. des Vorjahres.

Ich verpflichte mich hiermit, mich an die Satzung und die Beschlüsse der Mitgliederversammlungen und des Vorstandes zu halten (Satzung und Protokolle von Versammlungen können angefordert werden).

Als Bestätigung meiner Mitgliedschaft erhalte ich eine Quittung (Steuerbegünstigt gemäß §10b Einkommensteuergesetz).

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Unterschrift

## Bestellschein

Disketten:

Public-Domain:

| Stk: | BstNr. | Titel                           | Preis | Summe |
|------|--------|---------------------------------|-------|-------|
|      | PD 0   | gibt's noch nicht               |       |       |
|      | PD 1   | Elektronik und Funk             | 15.-  |       |
|      | PD 2   | Disk-Utilities v.2              | 15.-  |       |
|      | PD 3   | Comal Programmiersprache        | 15.-  |       |
|      | PD 4   | 8432 Emulator und mehr für 8432 | 15.-  |       |
|      | PD 5   | ROMs CBM 500-700 und 8028 *2    | 15.-  |       |
|      | PD 6   | Basic-Hilfen                    | 15.-  |       |
|      | PD 7   | Ernsthafte Anwendungen          | 15.-  |       |
|      | PD 8   | Spiele                          | 15.-  |       |
|      | PD 9   | Mathematik                      | 15.-  |       |
|      | PD 10  | Gless-Wrede-Sampler             | 15.-  |       |
|      | PD 11  | Protext-Druckertreiber u.a.     | 15.-  |       |
|      | PD 12  | Macro-Basic Beispielprogramme   | 15.-  |       |
|      | PD 13  | DOS 3.0 Listing 8280 (engl.)    | 15.-  |       |

Geschützte Software:

| Stk: | BstNr. | Titel                     | Preis | Summe |
|------|--------|---------------------------|-------|-------|
|      | GS 100 | Assembler I (v2.0)        | 55.-  |       |
|      | GS 101 | Macro-Basic               | 75.-  |       |
|      | GS 102 | Protext (nur 8250-Format) | 70.-  |       |
|      | GS 103 | Schreibmaschine           | 25.-  |       |
|      |        |                           |       |       |
|      |        | Zwischensumme             |       |       |

## Übertrag

### CBUG - Public-Domain:

| Stk: | BstNr.  | Titel                  |    | Preis | Summe |
|------|---------|------------------------|----|-------|-------|
|      | CBUG 5  | Financial              |    | 15.-  |       |
|      | CBUG 6  | CBUG - TPUG I          |    | 15.-  |       |
|      | CBUG 27 | Goceliak's Goldmine    | *4 | 15.-  |       |
|      | CBUG 36 | 700 Sampler (Demos)    |    | 15.-  |       |
|      | CBUG 48 | Diagnostics            | *4 | 15.-  |       |
|      | CBUG 67 | Entwicklungsunterlagen | *1 | 30.-  |       |
|      | CBUG x1 | Entertainment          |    | 15.-  |       |
|      | RR 1    | Norman's Utility v1.2  |    | 15.-  |       |
|      | PR 1    | Pre-Release I          |    | 15.-  |       |

### CBUG - geschützte Software:

| Stk: | BstNr.   | Titel                     |  | Preis | Summe |
|------|----------|---------------------------|--|-------|-------|
|      | CBUG 3   | Swan's Utility            |  | 24.-  |       |
|      | CBUG 16  | Swan's Basic-Course       |  | 33.-  |       |
|      | CBUG 17a | Liz's Utility v1.2a       |  | 27.-  |       |
|      | CBUG 32  | Kernaghan's Utilities v.3 |  | 17.-  |       |
|      | CBUG x2  | Beeline                   |  | 61.-  |       |

### Literatur:

| Stk: | BstNr.        | Titel                       |  | Preis | Summe |
|------|---------------|-----------------------------|--|-------|-------|
|      | Lit 1         | HSG ?! -Sammelband 0-3      |  | 15.-  |       |
|      | Lit 2         | CBUG-Escape Sammelband 1-7  |  | 38.-  |       |
|      | Lit 3         | CBUG-Escape Sammelband 8-10 |  | 15.-  |       |
|      | Lit 4         | Einführung in Comal         |  | 2.-   |       |
|      | Zwischensumme |                             |  |       |       |

Übertrag

Hardware:

| Stk: | BstNr. | Artikel                           | Preis | Summe |
|------|--------|-----------------------------------|-------|-------|
|      | HW 1   | Bank 15 - RAM-Platine bestückt    | 40.-  |       |
|      | HW 2   | Bank 15 - RAM-Leerplatine 5.-     | 5.-   |       |
|      | HW 3   | Zwischen-Sockel ROM-EPROM         | 7.-   |       |
|      | HW 4   | Basic, umschaltbar 128/256        | 35.-  |       |
|      | HW 5a  | Kernal, umschaltbar DIN/ASCII 600 | 19.-  |       |
|      | HW 5b  | Kernal, umschaltbar DIN/ASCII 700 | 19.-  |       |
|      | HW 6a  | Zeichensatz DIN/ASCII 600         | 15.-  |       |
|      | HW 6b  | Zeichensatz DIN/ASCII 700         | 15.-  |       |
|      | HW 7   | EPROM Typ .....*2                 | 15.-  |       |
|      | HW 7   | EPROM Typ .....*2                 | 15.-  |       |
|      | HW 8   | Farbbänder 8028                   | 5.-   |       |
|      | HW 9   | Typenrad 8028 Nr. ....*3          | 30.-  |       |
|      | HW 9   | Typenrad 8028 Nr. ....*3          | 30.-  |       |
|      | HW 9   | Typenrad 8028 Nr. ....*3          | 30.-  |       |
|      | HW 10a | m101 für 8028 normal IEEE         | 80.-  |       |
|      | HW 10b | m101 für 8028 32 kbyte IEEE       | 80.-  |       |
|      | HW 10c | m101 für 8229 (16 kbyte) IEEE     | 70.-  |       |
|      | HW 11a | m101 für 8028 normal Centronics   | 80.-  |       |
|      | HW 11b | m101 für 8028 32 kbyte Centr.     | 80.-  |       |
|      | HW 11c | m101 für 8229 (16kbyte) Centr.    | 70.-  |       |
|      | HW 12a | m101 Centr.-Hardware-Erweiterung  | 40.-  |       |
|      | HW 12b | dito. Leerplatine                 | 10.-  |       |

freiwillige Spende

Versandkostenpauschale 5,-DM

Summe

\* Anmerkungen

1) Drei Disketten, die nur abgegeben werden, wenn die Erklärung für Commodore unterschrieben der Bestellung beigelegt wird.

2) EPROM-Dienst

Die auf der ROM-Disk zu findenden ROM/EPROM-Inhalte sind auch als EPROM zu haben (bitte genauen Titel gemäß Directory-Eintrag angeben!!). Für Basic und Kernal empfehlen wir die umschaltbaren Club-Versionen mit Schalter (einmal um) und Widerstand (3.3 kOhm) und Verbindungsdraht (ca. 40 cm/0.5 qmm).

3) Typenräder

Bitte unbedingt getrennten Verrechnungsscheck beigeben, bei dem die Auszahlungssumme offengelassen ist (mit Zahlungsbezeichnung).

4) Software von Tony Goceliak ist nur bei Abgabe einer besonderen Erklärung gestattet.

Den Betrag über ..... (Summe der Bestellung)

O habe ich auf das Konto der CBM-Selbsthilfe überwiesen

O zahle ich mit beiliegendem Verrechnungsscheck.

Konto der CBM-Selbsthilfe e.V. (i.G.):

Gerd Schumacher,  
KNr. 3443 57-503,  
BLZ. 370 100 50.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Vorname

\_\_\_\_\_  
Straße, Nr.

\_\_\_\_\_  
PLZ.

\_\_\_\_\_  
Ort

\_\_\_\_\_  
Telefon

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Unterschrift