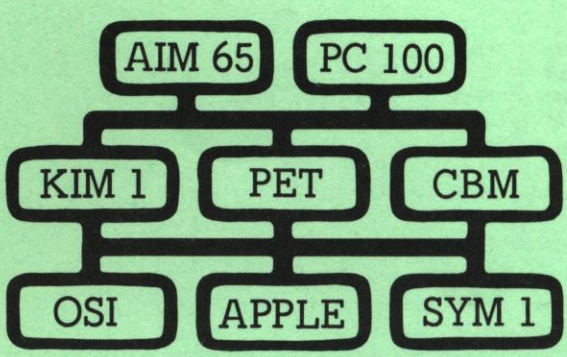
65xx MICRO MAG

COMPUTING · SOFTWARE · HOBBY

DM 9,50 Nr. 34 Dezember 1983



Inhaltsverzeichnis

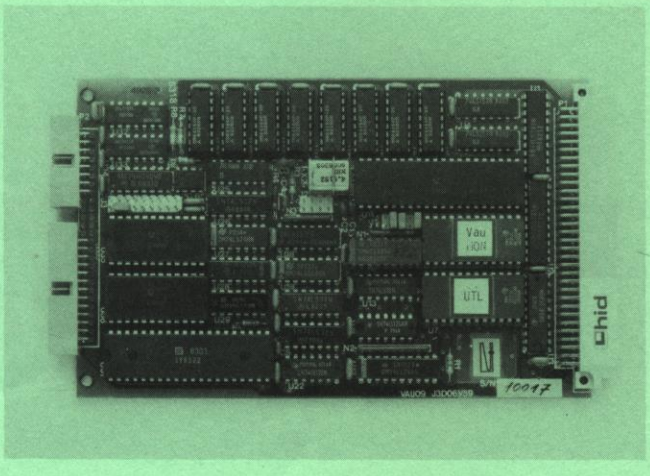
Mini Floppy Interface für AIM (2) 3
Supertape for CBM 19
EVAL für CBM 3001 25
ROFF (2) - Textformatierung für 6502-Mikroprozessoren 27
Einkommensteuerberechnung für 1983 49
Bücher 18, 51
Editorial 54
64K DRAM am AIM - Aufbau und Anwendungen 55
AIM Spezial (16) 57
Aus der Branche 58

Produkt-Info

VauSYS 09

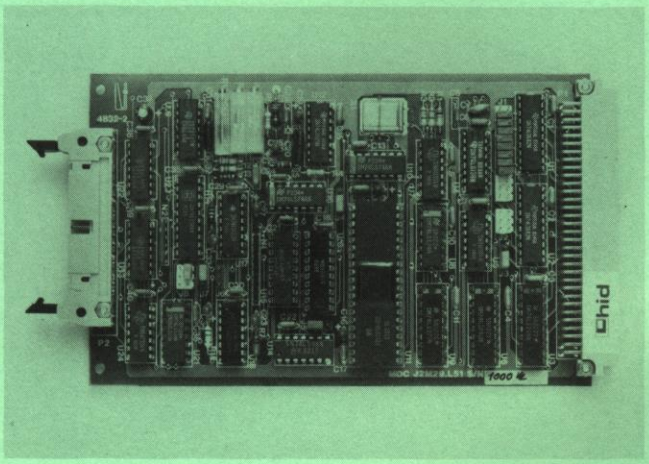
microcomputer - die bessere technik

Vau09
6809 Single Board Computer



DM 875,-
(DM 997,50 inkl. MwSt)

Floppy Disk Controller
(jetzt auch für AIM65 lieferbar)



DM 798,-
(DM 909,72 inkl. MwSt)

Betriebssysteme **FLEX** und **OS9** lieferbar.

HID · DIPL. ING. (FH) B. HECKL
ALTE WALLENSTEINSTR. 146
8500 NÜRNBERG 80
☎ 0911 / 65 17 47

ohid

Michael Zimmermann, 6102 Pfungstadt

Mini Floppy Interface für AIM (2)

4.1 Software für das Floppy-Interface

Nach der Beschreibung der Hardware soll im Folgenden die erforderliche Software dargestellt werden, die für den Anschluß einer Floppy-Disk an den AIM erforderlich ist. Wie bei der Hardware soll mehr Wert auf Einfachheit als auf Eleganz gelegt werden. Die Darstellung erfolgt dabei in mehreren Schichten. Am Anfang stehen Überlegungen zur Implementierung und eine Betrachtung des Controllers in Hinblick auf seine Programmierung. Daran schließt sich eine Beschreibung des physikalischen Routinen zum Steuern des Laufwerkes an. Diese werden dann zu logischen Routinen zusammengefaßt, die bereits eine Dateistruktur enthalten. Diese Dateien sind sowohl von Routinen des AIM als auch von Benutzerprogrammen aus in bestimmter Art und Weise ansprechbar. Auch dies wird dargestellt. Den Abschluß der Software bildet eine Reihe von Dienstprogrammen, die erforderlich sind, um z.B. Disketten zu initialisieren, Dateien anzulegen oder auch um Reparaturen an einzelnen Sektoren vorzunehmen. Nach diesem Ausblick auf die gesamte Palette der Software nun zuerst die allgemeinen Überlegungen zur Implementation.

4.2 Hinweise zur Implementation

Das das Interface steuernde Programm sollte in seiner Adressierung entweder im oberen Bereich des RAM liegen oder als Prom in einem vollkommen separaten Adreßbereich untergebracht werden. Der Controller selber belegt einen Adreßraum von 1 Page, zusammen mit der Motorsteuerung sind insgesamt 1 KB erforderlich. Zusätzlich zu Steuerprogramm und Controller sind nochmal 1 Page als Puffer und Kontrollspeicher sowie 6 Bytes in der Page Zero für Pointer erforderlich. Bezüglich der hier real verwendeten Adressen sei auf den kommentierten Programmausdruck verwiesen. Die Sektorgröße wurde zu 128 Bytes gewählt, der Sektorpuffer liegt dabei in der unteren Hälfte der Puffer-/Kontrollpage. Auf einen theoretisch möglichen größeren Sektor wurde bewußt verzichtet, um eine Adressierung des gesamten Sektor-Puffers über ein einziges Indexregister zu gewährleisten. Bei der nächst möglichen Sektorgröße von 256 Byte wäre dies schon nicht mehr zu bewerkstelligen.

Bei einfacher Aufzeichnungsdichte passen jeweils 16 Sektoren in eine Spur. Bei normalerweise 40 Spuren pro Diskette ergibt sich somit eine Gesamtkapazität von 80 K. Weitere Überlegungen zur Implementation, insbesondere soweit sie spezielle Probleme berühren, werden im Zusammenhang mit der Programmbeschreibung erläutert.

4.3 Programmierung des Controllers SYP 1793-002

Als digitaler Schaltkreis besitzt dieser Controller die Fähigkeit, auf bestimmte von außen zugeführte Befehle zu reagieren. Für deren Ausführung und Kontrolle der Ausführung stehen insgesamt 5 Register zur Verfügung, die in folgender Weise adressiert werden:

A1	A0	READ	WRITE
0	0	Status	Command
0	1	Track	Track
1	0	Sector	Sector
1	1	Data	Data

Grundsätzlich wird ein Befehl dadurch angestoßen, daß ein bestimmtes Befehlswort in das Command-Register eingeschrieben wird, die Befehlsausführung wird danach über das Status-Register kontrolliert. Track-, Sector- und Data-Register dienen dabei als weitere Parameter bei der Befehlsdefinition bzw. -ausführung. Eine Darstellung der möglichen Befehle gibt folgende Übersicht aus dem Datenblatt des WD 1793 (13):

Table 2. COMMAND SUMMARY

TYPE	COMMAND	BITS 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	u	h	V	r_1	r_0
I	Step In	0	1	0	u	h	V	r_1	r_0
I	Step Out	0	1	1	u	h	V	r_1	r_0
II	Read Sector	1	0	0	m	S	E	C	0
II	Write Sector	1	0	1	m	S	E	C	a_0
III	Read Address	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrrupt	1	1	0	1	I_3	I_2	I_1	I_0

Note: Bits shown in TRUE form.

Table 3. FLAG SUMMARY

TYPE I COMMANDS

h = Head Load Flag (Bit 3)
- h = 1, Load head at beginning
- h = 0, Unload head at beginning

V = Verify flag (Bit 2)
- V = 1, Verify on destination track
- V = 0, No verify

r_1r_0 = Stepping motor rate (Bits 1-0)

Refer to Table 1 for rate summary

u = Update flag (Bit 4)
- u = 1, Update Track register
- u = 0, No update

Table 4. FLAG SUMMARY

TYPE II & III COMMANDS

m = Multiple Record flag (Bit 4)
- m = 0, Single Record
- m = 1, Multiple Records

a_0 = Data Address Mark (Bit 0)
- a_0 = 0, FB (Data Mark)
- a_0 = 1, F8 (Deleted Data Mark)

E = 15 ms Delay (2MHz)
- E = 1, 15 ms delay
- E = 0, no 15 ms delay

S = Side Select Flag
- S = 0, Compare for Side 0
- S = 1, Compare for Side 1

C = Side Compare Flag
- C = 0, disable side select compare
- C = 1, enable side select compare

Table 5. FLAG SUMMARY

TYPE IV COMMAND

Ii = Interrupt Condition flags (Bits 3-0)
- I0 = 1, Not-Ready to Ready Transition
- I1 = 1, Ready to Not-Ready Transition
- I2 = 1, Index Pulse
- I3 = 1, Immediate Interrupt
- I_3 -I_0 = 0, Terminate with no Interrupt

Wie zu ersehen, sind Befehle in mehrere Typen eingeteilt. Mit den Befehlen des Typ 1 wird der Schreib-Lesekopf über die Spuren bewegt. Der Typ 2 dient dem Lesen von ganzen Sektoren und Adreßfeldern sowie dem Schreiben von Sektoren. Mit dem Typ 3 werden ganze Spuren ohne Rücksicht auf Sektorgrenzen bearbeitet. Diese Befehle werden z.B. für die Initialisierung benutzt. Der Typ 4 stellt einen Sonderfall dar, der Force-Interrupt-Befehl dient zum Abbrechen einer laufenden Operation. Er wird hier nicht verwendet. - In welcher Weise reagiert nun der Controller auf die Befehle? Zum einem mit Aussenden und Bearbeiten von Signalen zum Floppy-Drive, zum anderen mit Setzen bestimmter Ergebnis-Flags im Status-Register. Eine Übersicht über diese Flags in Abhängigkeit von den unterschiedlichen Befehlen sei wieder (13) entnommen:

(Abbildung auf der folgenden Seite)

Mit diesem Rüstzeug ist eine Besprechung der ersten grundlegenden Routinen für das Floppy-Interface möglich.

4.4 Physikalische Routinen steuernder Art

Die innerste Schale von Routinen zur Ansteuerung des Floppy-Laufwerkes bilden die rein physisch arbeitenden Routinen, die noch keine Rücksicht auf irgendwelche Dateistrukturen nehmen. Da der Controller selber keine Hilfsmittel zur Ansteuerung mehrerer Laufwerke und zur Motorkontrolle bietet, wird dies durch benutzereigene Schaltkreise bewirkt. Zum einen sind dies die gleichen Steuerungen wie für die Cassettenlaufwerke, zum anderen wird durch bestimmte Adressen Motor ON und OFF bewirkt.

Die Routine DISK ON schaltet hierbei das gewünschte Laufwerk ein, sie ist vergleichbar der Routine TIOSET im AIM-Monitor, bezieht nur ihre Information über das einzuschaltende Lauf-

Table 6. STATUS REGISTER SUMMARY

BIT	ALL TYPE I COMMANDS	READ ADDRESS	READ SECTOR	READ TRACK	WRITE SECTOR	WRITE TRACK
S7	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY
S6	WRITE PROTECT	0	0	0	WRITE PROTECT	WRITE PROTECT
S5	HEAD LOADED	0	RECORD TYPE	0	WRITE FAULT	WRITE FAULT
S4	SEEK ERROR	RNF	RNF	0	RNF	0
S3	CRC ERROR	CRC ERROR	CRC ERROR	0	CRC ERROR	0
S2	TRACK 0	LOST DATA	LOST DATA	LOST DATA	LOST DATA	LOST DATA
S1	INDEX	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

STATUS FOR TYPE I COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset it indicates that the drive is ready. This bit is an inverted copy of the Ready input and logically 'ored' with MR.
S6 PROTECTED	When set, indicates Write Protect is activated. This bit is an inverted copy of WRPT input.
S5 HEAD LOADED	When set, it indicates the head is loaded and engaged. This bit is a logical "and" of HLD and HLT signals.
S4 SEEK ERROR	When set, the desired track was not verified. This bit is reset to 0 when updated.
S3 CRC ERROR	CRC encountered in ID field.
S2 TRACK 00	When set, indicates Read/Write head is positioned to Track 0. This bit is an inverted copy of the $\overline{TROO}$ input.
S1 INDEX	When set, indicates index mark detected from drive. This bit is an inverted copy of the $\overline{IP}$ input.
S0 BUSY	When set command is in progress. When reset no command is in progress.

STATUS BITS FOR TYPE II AND III COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset, it indicates that the drive is ready. This bit is an inverted copy of the Ready input and 'ored' with MR. The Type II and III Commands will not execute unless the drive is ready.
S6 WRITE PROTECT	On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated.
S5 RECORD TYPE/ WRITE FAULT	On Read Record: It indicates the record-type code from data field address mark. 1 = Deleted Data Mark. 0 = Data Mark. On any Write: It indicates a Write Fault. This bit is reset when updated.
S4 RECORD NOT FOUND (RNF)	When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated.
S3 CRC ERROR	If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated.
S2 LOST DATA	When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when updated.
S1 DATA REQUEST	This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated.
S0 BUSY	When set, command is under execution. When reset, no command is under execution.

werke aus einer Tabelle. Das Gegenteil dieser Routine bewirkt DISK OFF. Dieser Programmteil setzt zuerst das vorher ausgewählte Laufwerk zurück, deaktiviert die Auswahl und schaltet anschließend den Laufwerksmotor durch Adressierung der entsprechenden Flip-Flop-Adresse ab.

Die Routine ACCEPT COMMAND erwartet ein Befehlsbyte im Akkumulator und speichert dieses in das Command-Register. Da der Controller eine gewisse Zeit zur Entgegennahme und Dekodierung eines Befehles benötigt, wird in der Schleife AC10 das Busy-Flag des Status-Registers solange abgeprüft, bis es gesetzt ist und der Controller damit signalisiert, daß er den Befehl entgegengenommen hat und in die Ausführung einsteigt. Diese Routine wird für die Ausführung aller Befehle benutzt, vom Typ 1 sind dies der Restore- und der Seek-Befehl.

Für den Restore-Befehl wird zunächst das Command-Byte in den Akku geladen, dies steht auf einer Step-Rate von 30 msec.. Dann wird die Befehlsausführung durch Aufruf von ACMD angestoßen. Bei Rückkehr aus dieser Routine befindet sich der Befehl in Ausführung. Die Restore-Routine muß jetzt nur noch das Busy-Flag des Status-Registers solange prüfen, wie dieses gesetzt ist (in der Schleife RS10). Hat der Schreibkopf des Laufwerkes die Spur Null erreicht, so wird das Busy-Flag gelöscht. Die Restore-Routine hat dann ihren Zweck erfüllt und kann die Kontrolle wieder an das aufrufende Programm zurückgeben.

Sehr ähnlich arbeitet die Seek-Routine. Der einzige Unterschied ist, daß beim Aufruf die Nummer der anzusteuernden Spur im Akku steht. Diese Nummer wird in das Daten-Register geladen und der Seek-Befehl aufgesetzt. Der Controller ermittelt jetzt durch Vergleich mit der im Track-Register stehenden Nummer der aktuellen Spur die Richtung der Bewegung und die Anzahl der Schritte, die ausgeführt werden. Hier zeigt sich exemplarisch, daß der Controller dem Anwender eine Reihe von Dienstleistungen abnimmt. Das Ende der Operation wird wieder durch Löschen des Busy-Flag angezeigt.

4.5 Physikalische Routinen mit Datentransfer

Während die vorstehenden Programmteile noch keinen Datentransfer bewirkten, werden bei den folgenden Routinen Read- bzw. Write-Sector Daten von der Diskette geholt oder auf dieser abgelegt. Beide Routinen erwarten beim Aufruf, daß der Lese-/Schreibkopf über der gewünschten Spur positioniert ist. Dies wird durch einen vorhergehenden Seek-Befehl bewirkt. Sie erwarten auch, daß im Akku die Nummer des zu bearbeitenden Sektors steht. Die erste Aktion ist daher das Versorgen des Sector-Registers. Da das Lesen und das Schreiben Vorgänge sind, die sehr engen zeitlichen Bedingungen unterliegen, wird von Vornherein für den Fehlerfall eine Befehlswiederholung vorgesehen. Die Anzahl der möglichen Versuche bestimmt die Konstante TRY, hier mit dem Wert 8, die in das Arbeitsfeld TCNT geladen wird.

In der folgenden Befehlsschleife RD05/WR05 wird zunächst das Indexregister mit Null initialisiert und der Akku mit dem entsprechenden Befehlsbyte geladen, die Ausführung wird mit ACMD angestoßen. In der Schleife RD20/WR20 wird das Statusregister geladen und ausgewertet. Ist das Busy-Flag gelöscht, so ist der Befehl noch in Ausführung. Ist das Datenflag ebenfalls gelöscht, so ist ein Datentransfer nicht erforderlich, es wird nach RD20/WR20 zu einer weiteren Prüfung zurückverzweigt. Wurde das Datenflag gesetzt, so wird der Datentransfer zwischen Sektor-Puffer im Speicher und Datenregister des Controllers durchgeführt und zur Bearbeitung des nächsten Bytes verzweigt.

Der Sektor-Puffer wird hierbei festen Adressen zugeordnet. Auf eine indirekt-indizierte Adressierung nur über einen Pointer in der Page Zero wurde wegen bereits angeführter enger Zeitbeschränkungen verzichtet. Die Zeitbeziehungen sind als Werte in der Routine als Kommentare verzeichnet. Zu beachten ist, daß nach dem Setzen des Daten-Flag der Controller beim Lesen in 55 und beim Schreiben in 47 Mikrosekunden bedient werden muß. Wurde der gesamte Sektor bearbeitet, so werden in RD90/WR90 der Status auf den Wert Null und der Index auf 128 (Anzahl der Bytes im Sektor) überprüft. Stimmen diese Werte, so wird der Status, hier Null, in den Index übertragen und zurückverzweigt. Der gesamte Sektor befindet sich jetzt im Pufferspeicher. Im Fehlerfall wird die Befehlsschleife nochmals durchlaufen, solange, bis der Maximalwert für die Operationsversuche ereeicht ist. Auch in diesem Falle wird mit dem Status, hier jedoch

ungleich Null, im X-Register zurückgesprungen. Das aufrufende Programm hat jetzt die Möglichkiet, abhängig vom Fehlertyp die ihm erforderlich scheinenden Maßnahmen einzuleiten.

5.1 Verarbeitung logischer Datenbestände

Nur in den seltensten Fällen werden sich die benötigten Daten, sei es für ein Programm zum Laden in den Kernspeicher oder als Textfile für den Editor oder als Datei für ein Benutzerprogramm, in einem Block unterbringen lassen. Der Regelfall wird sein, daß mehrere Sektoren für den gesamten Datenbestand erforderlich werden. Aus diesem Grunde sollen hier zuerst die Möglichkeiten aufgezeigt werden, in welchen Formen diese Zusammenfassung von mehreren physikalischen Einheiten zu logischen Zusammenhängen bewerkstelligt wird. Eine Übersicht über die herkömmlichen Verfahren liefert uns Fritzon (4), der in kurzer aber prägnanter Weise die hauptsächlichen Unterschiede herausarbeitet. Es sind zu unterscheiden:

- Sequentielle Zuordnung. Die logische Folge entspricht der pysikalischen Anordnung.
- Verkettete Zuordnung. Die logische Folge wird durch Zeiger bestimmt.
- Tabellarische Zuordnung. Die logische Folge ist einer Tabelle zu entnehmen.

Während die sequentielle Anordnung durch ihre Einfachheit besticht, bieten die anderen Verfahren einen erhöhten Komfort hinsichtlich des dynamischen Verhaltens von Dateigrößen. Da das hier angestrebte Ziel aber eine möglichst einfache Lösung ist, soll lediglich eine sequentielle Blockanordnung verwirklicht werden. Bezüglich einer tiefer gehenden Beschreibung, auch der anderen Verfahren und ihrer Charakteristika, sei auf o.a. Literatur verwiesen.

5.2 Der File-Control-Block

Unabhängig vom Verfahren der Blockanordnung in einer Datei ist für Zugriffe in jedem Fall eine Reihe von steuernden Informationen erforderlich, die in einem Speicherbereich zusammengefaßt und als File-Control-Block (FCB) bezeichnet werden sollen. In der hier beschriebenen Implementation hat der FCB eine Größe von 16 Bytes und belegt die oberen Adressen des Kontroll-RAM. Er enthält in diesem Fall der rein sequentiellen Speicherung folgende Informationen:

- den File-Namen
- das Laufwerk, auf dem sich die Datei befindet
- die Adresse des Sektors im Benutzerprogramm
- Spur und Sektor des Dateibeginns
- Spur und Sektor des aktuellen Dateizugriffes
- Spur und Sektor des Dateiendes.

Wie diese Daten in den FCB eingestellt werden und wie dieser in den Kontroll-RAM kommt, soll hier nicht weiter erläutert werden. Auf jeden Fall dürfte einsichtig sein, daß für alle Zugriffe die Spur- und Sektor-Informationen von überragender Bedeutung sind.

5.3 Routinen zum sequentiellen Lesen bzw. Schreiben von Sektoren

Die Routinen zum Lesen und Schreiben weisen auch hier eine große Ähnlichkeit im Aufbau auf, sie sollen deswegen zusammen beschrieben werden. Grundsätzlich gliedert sich die sequentielle Verarbeitung in drei Schritte:

- Einstellen des Laufwerkes auf den gewünschten Sektor,
- Ausführen der Verarbeitung,
- Vorbereitung der nächsten Verarbeitung durch Erhöhen des Sektorzählers.

Diese Struktur findet sich in den Routinen RNS (Read Next Sector) und WNS (Write Next Sector). Das jeweils anschließende Initialisieren des Puffer-Pointers ist eine Besonderheit, die sich aus den Gegebenheiten der Pufferverarbeitung ergibt. Wurde bei einem der Teile des sequentiellen Verarbeitens ein Fehler erkannt, so wird jeweils zum Ende der Routinen RNS bzw. WNS verzweigt, das Index-Register enthält in diesen Fällen einen Fehlercode ungleich Null, der zur Auswertung durch das aufrufende Programm zur Verfügung steht. Die eigentlichen verarbeitenden Routinen RDS und WRS wurden bereits bei den physikalischen Zugriffen besprochen. Neu sind hier nur EINS zum Einstellen auf den gewünschten Sektor und INCS zum Erhöhen des Sektorzählers.

In der Routine EINS wird zuerst durch Vergleich der im FCB stehenden Werte für Endsektor und Spur mit den Angaben des aktuellen Sektor und Spur überprüft, ob der gewünschte Sektor überhaupt noch in den Dateigrenzen ist. Im Fehlerfalle wird hier mit dem Code $FC die weitere Verarbeitung abgebrochen. Liegt keine Überschreitung der Dateigrenzen vor, so wird die aktuelle Spur aus dem FCB entnommen und mit dieser eine Suchoperation mit der bereits beschriebenen Routine SEEK aufgesetzt. Wurde diese erfolgreich abgeschlossen, so wird das Fehlerflag gelöscht und der aktuelle Sektor in den Akku geladen, als Vorbereitung für die folgende Verarbeitung in der Zugriffsoperation.

In der der Zugriffsoperation folgenden Routine INCS wird im Sinne einer sequentiellen Verarbeitung die Einstellung auf den logisch und physisch folgenden Sektor vorbereitet. Hierzu wird die Nummer des aktuellen Sektors aus dem FCB erhöht. Überschreitet diese die maximale Sektorzahl pro Spur, so wird der Spurzähler erhöht und der Sektorzähler auf die Anfangsnummer (01) zurückgesetzt. Eine Abfrage auf Dateigrenzen ist hierbei noch nicht möglich, da zu diesem Zeitpunkt noch nicht feststeht, ob der Zugriff überhaupt ausgeführt wird oder ob der gerade gelesene Sektor der letzte der Datei war.

Mit diesen Routinen sind wir in der Lage, Dateien zu verarbeiten, die eine Ausdehnung haben, die über die Kapazität eines Sektors hinausgeht. Für eine richtige Verarbeitung fehlen aber noch einige Voraussetzungen, die insbesondere das Füllen des FCB betreffen, eine Lücke, die jetzt zu schließen ist.

5.3 Das Disketten-Inhaltsverzeichnis und das Eröffnen von Dateien

Bei jeder Speicherung auf externen Medien ist es von Wichtigkeit, welche Dateien sich auf einem Speicher befinden. Dies gilt auch für Disketten. Die auf einer Diskette befindlichen Dateien werden hier in einem Inhaltsverzeichnis, auch als Volume Table of Contents (VTOC) bezeichnet, aufgeführt. Die Eintragungen in diesen VTOC sollen hier die gleiche Form haben, wie die FCB's. Eine Gesamtgröße des VTOC von 1 Sektor sollte ausreichen. Hiermit können insgesamt 8 Dateien auf einer Diskette Platz finden, von der Kapazität des VTOC her bestimmt. Als Stelle an der der VTOC auf der Diskette abgelegt ist, wurde der 1. Sektor der Spur 0 gewählt. Dies hat den Vorteil, daß dieser Sektor mit Hilfe des Restore-Befehles angewählt werden kann, einem Befehl, der sowieso für das Setzen definierter Anfangsbedingungen sinnvollerweise vor dem Zugriff auf eine Datei gegeben wird. Zum anderen weist die Spur 0 an der Außenseite der Diskette die am wenigsten kritischen Magnetisierungsbedingungen auf, sicher auch ein Kriterium für eine so eminent wichtige Information, wie es das Inhaltsverzeichnis einer Diskette darstellt.

Mit diesem Inhaltsverzeichnis ist es nun möglich, den gewünschten FCB aus dem VTOC in den entsprechenden Bereich des Kontroll-RAM zu übertragen und damit eine Datei auf einer im Laufwerk geladenen Diskette für die Verarbeitung bereit zu machen.

Bei dem im folgenden zu beschreibenden OPEN-Vorgang wird durch externe Eingabe die Auswahl der gewünschten Datei vorgenommen. Erste Aktion der OPEN-Routine ist ein Lesen des Statusregisters des Floppy-Controllers. Hierdurch wird das Motor-On-Signal für die angeschlossenen Laufwerke aktiviert. Diese haben Zeit, parallel mit den folgenden manuellen Eingaben hochzulaufen. Als weiteres wird der Name des gewünschten File vom Benutzer angefordert. Hierzu dient die Routine DNAM, die den File-Namen und Laufwerksnummer mit Hilfe der Monitorroutine FNAM anfordert. In DNAM werden anschließend Datei-Name und Laufwerksnummer in den aufzubauenden FCB im Kontroll-RAM kopiert.

Nachdem dieser Teil des FCB aufgebaut ist, ist es an der Zeit, die Informationen über die Lage der Datei auf der Diskette zu gewinnen. Hierzu wird zuerst das gewünschte Laufwerk selektiert und dann der 1. Sektor, der den VTOC der Diskette enthält, in den Sektorpuffer geladen. In diesem VTOC im Puffer werden in der Routine OP10 die File-Namen gegen den bereits im FCB aufgebauten Namen des gewünschten Files verglichen. Wurde bei einem Eintrag eine Ungleichheit erkannt, so daß es sich bei diesem nicht um das gewünschte File handelt, so wird nach OP30 verzweigt. - In dieser Routine wird auf den nächsten Eintrag eingestellt und mit diesem weiter

verglichen. Wurden alle Eintragungen verarbeitet und hatte der im FCB stehende Dateiname keine Entsprechung im VTOC, so wird zur Verarbeitung des Open-Fehlers verzweigt. Dieser wird durch den Fehlercode $FF angezeigt. - Wurde Gleichheit zwischen VTOC-Eintrag und Filename im FCB erkannt, so verzweigt die Verarbeitung zu OP40.

In diesem Routinenteil werden die Informationen über die Ausdehnung des File vom VTOC in den FCB übernommen. Hiermit ist das File endgültig eröffnet. Es steht damit zur Bearbeitung mit den bereits besprochenen Routinen zur Verfügung. Aber auch diese Eröffnung reicht für eine Verarbeitung durch die Monitor-Routinen des AIM nicht aus. Hierzu bedarf es weiterer Strukturen, die ein Bindeglied zwischen der zeichenorientierten Verarbeitung des AIM-Monitors und der blockorientierten Arbeitsweise der Floppy Disk bilden.

Literaturliste

(1) Allen, David: A Minifloppy Interface. BYTE 2/1978
(2) BASF: BASF 6106 Mini Disk Drive Interface Specification. Rev. 3, 1977
(3) Cuyper, J. de: Floppy Disk Interface für Junior. Elektor 11-12 1982
(4) Intel: 8272 Single/Double Density Floppy Disk Controller. 1. 80
(4) Fritzon, Richard: Data on Disk, Implementing File Systems. Kilobaud 1/1981
(5) Lesea, Austin, Zaks, Rodney: Mikroprozessor Interface Techniken. Markdorf 1979
(6) Motorola: M6800 Microprocessor Application Manual. 1975
(7) Motorola: A Floppy Disk Controller using the MC6852 SSDA and other MC6800 Microprocessor Family parts. Application Note 764 1976
(8) Motorola: Floppy Disk Controller MC6843. Advance Information 1978
(9) Nicholson, James, Camp, Roger: Build a Super Simple Floppy Disk Interface. BYTE 5-6/1981
(10) Pausch, Albrecht: Mini Floppy Interface. MC 10-11/1982
(11) Schnell, Dr. A.: AIM 65 mit 'fremder' Systemsoftware. 65xx MICRO MAG Nr. 17, 1981
(12) Welles, Dr. Kenneth B.: Build This Economy Floppy Disk Interface. Software for the Economy Floppy Disk. BYTE 2/1977 und BYTE 6/1977
(13) Western Digital Corporation: FD 179X-01 Floppy Disk Formatter/Controller Family. 1979

```
;---------------------------------------------------------------
;FLOPPY-FORMATER
;
;---------------------------------------------------------------
;06.11.83
;
;---------------------------------------------------------------
;FESTWERTE

0000            0003 TRY     =$0B            ;ANZAHL DER VERSUCHE
0000            0004 CR      =$0D            ;CARRIAGE RETURN-ZEICHEN
0000            0005 MSEC    =$10            ;MAXIMALER SECTOR
0000            0006 NULL    =$30            ;NULL
0000            0007 SEMI    =$3B            ;SEMIKOLON
;---------------------------------------------------------------
;MONITOR-ROUTINEN

0000            0009 START   =$E182          ;EINSPRUNG IN AIM-MONITOR
0000            0010 CKER00  =$E394          ;AUSGEBEN ERROR
0000            0011 KEP     =$E7AF          ;DRUCKEN MELDUNG
0000            0012 QM      =$E7D4          ;AUSGEBEN FRAGEZEICHEN
0000            0013 EQUAL   =$E7D8          ;AUSGEBEN GLEICH-ZEICHEN
0000            0014 BLANK2  =$E83B          ;AUSGEBEN 2 BLANKS
0000            0015 BLANK   =$E83E          ;AUSGEBEN 1 BLANK
0000            0016 FNAM    =$E8A2          ;ANFORDERN FILE-NAME
0000            0017 OUTLOW  =$E901           ;AUSGABE AUF TERMINAL LENKEN
0000            0018 RDRUB   =$E95F          ;LESEN ZEICHEN
0000            0019 OUTPUT  =$E97A          ;AUSGEBEN 1 ZEICHEN
0000            0020 CRLOW   =$EA13          ;AUSGEBEN CR UND LF
0000            0021 NUMA    =$EA46          ;1 BYTE ALS 2 ASCII-ZEICHEN AUSGEBEI
0000            0022 RD2     =$EA5D          ;2 ZEICHEN LESEN UND PACKEN
0000            0023 PHXY    =$EB9E          ;X UND Y AUF STACK ABLEGEN
```

```
CONTROLLER-ADRESSEN

0000            0025 CMR     =$B000          ;COMMAND-REGISTER
0000            0026 STR     =$B000          ;STATUS-REGISTER
0000            0027 TRR     =$B001          ;TRACK-REGISTER
0000            0028 SER     =$B002          ;SECTOR-REGISTER
0000            0029 DAR     =$B003          ;DATA-REGISTER
0000            0030 DRB     =$AB00          ;DRIVE-CONTROL-REGISTER

MONITOR-RAM-ADRESSEN

0000            0032 NAME    =$A42E          ;FILE-NAME
0000            0033 TAPIN   =$A434          ;IN-FLAG
0000            0034 TAPTR   =$A436          ;TAPE-BUFFER-POINTER
0000            0035 TAPTR2  =$A437          ;2. TAPE-BUFFER-POINTER
0000            0036 JUMP    =$A47D          ;SPEICHER FUER INDIREKTEN SPRUNG

PAGE 0 ADRESSEN

0000            0038 PPTR    =$FA            ;PARAMETER-POINTER
0000            0039 SPTR    =$FC            ;SEKTOR-POINTER
0000            0040 FPTR    =$FE            ;FCB-POINTER

PAGE 1 ADRESSEN

0000            0042         *=$108
0108 005C       0043         .WOR UREA       ;LINK ZUM USER-INPUT
010A 365C       0044         .WOR UWRT       ;LINK ZUM USER-OUTPUT
010C            0045 TABUF   =$5B00          ;TAPE-PUFFER
010C            0046 TCNT    =TABUF+$E0      ;ANZ. D. OPERATIONSVERSUCHE B. BEFEHL
010C            0047 SAVA    =TABUF+$E1      ;SAVE FUER AKKU
010C            0048 SAVX    =TABUF+$E2      ;SAVE FUER X
010C            0049 SAVY    =TABUF+$E3      ;SAVE FUER Y
010C            0050 T1      =TABUF+$E8      ;LETZTES AUSGABEZEICHEN
010C            0051 T2      =TABUF+$E9      ;VORLETZES AUSGABEZEICHEN
010C            0052 T3      =TABUF+$EA      ;DRITTLETZTES AUSGEBZEICHEN
010C            0053 ECNT    =TABUF+$EB      ;EOF-ZAEHLER
010C            0054 TTRK    =TABUF+$EC
010C            0055 TSEC    =TABUF+$ED
010C            0056 FCB     =TABUF+$F0      ;AKTUELLER FILE-CONTROL-BLOCK
010C            0057 FCBS    =FCB+7
010C            0058 SADR    =FCB+8          ;BENUTZER-PROGRAMM-ADRESSE
010C            0059 VTRK    =FCB+10         ;BEGINN-TRACK
010C            0060 VSEC    =FCB+11         ;BEGINN-SEKTOR
010C            0061 ATRK    =FCB+12         ;AKTUELLER TRACK
010C            0062 ASEC    =FCB+13         ;AKTUELLER SEKTOR
010C            0063 HTRK    =FCB+14         ;HOECHSTER TRACK
010C            0064 HSEC    =FCB+15         ;HOECHSTER SEKTOR

USER PROGRAMM

010C            0066         *=$200
0200 20F85F     0067         JSR JPCMD
0203 4F         0068         .BYT 'O'
0204 0003       0069         .WOR UFCB
0206 A900       0070         LDA #00
0208 8D0C03     0071         STA UFCB+12
020B A907       0072         LDA #07
020D 8D0D03     0073         STA UFCB+13
0210 20F85F     0074         JSR JPCMD
0213 53         0075         .BYT 'S'
0214 0003       0076         .WOR UFCB
0216 20F85F     0077         JSR JPCMD
0219 52         0078         .BYT 'R'
021A 0003       0079         .WOR UFCB
021C 4CBFE0     0080         JMP $E0BF
```

```
021F            0081          *=$300
0300 4630       0082 UFCB     .BYT 'F0
0308 8003       0083          .WOR USEC
030A 2020       0084          .BYT '
0312            0085          *=$380
0380            0086 USEC
0380            0087          *=$5C00
```

LOGISCHE USER-ZUGRRIFF-ROUTINEN

USER READ

```
5C00            0090 UREA
5C00 B00B       0091          BCS URC          ;WENN CARRY GESETZT EIN ZEICHEN LESEN
5C02 20CB5C     0092          JSR OPEN         ;DATEI EROEFFNEN
5C05 D02C       0093          BNE USER         ;FEHLER BEI OPEN
5C07 20235D     0094          JSR RNS          ;UND ERSTEN SETKTOR LESEN
5C0A D027       0095          BNE USER         ;FEHLER IM 1.SEKTOR
5C0C 60         0096          RTS
```

ZEICHEN LESEN

```
5C0D            0098 URC
5C0D 8EE25B     0099          STX SAVX         ;ZUERST INDEX-REGISTER SICHERN
5C10 8CE35B     0100          STY SAVY
5C13 AE36A4     0101          LDX TAPTR        ;POINTER IN X LADEN
5C16 100D       0102          BPL UI10         ;POINTER < 128=ZEICHEN AUS PUFFER HOLEN
5C18 20235D     0103          JSR RNS          ;SONST NAESTEN SEKTOR LESEN
5C1B D016       0104          BNE USER         ;FEHLER BEIM LESEN

5C1D AD7F5B     0105          LDA TABUF+$7F    ;LETZTES ZEICHEN AUS BUFFER HOLEN
5C20 D003       0106          BNE UI10         ;IST DIES <> NULL=DATEI GEHT WEITER
5C22 20155E     0107          JSR DOFF         ;GLEICH NULL=DATEI ZU ENDE=ABSCHALTEN
5C25            0108 UI10
5C25 BD005B     0109          LDA TABUF,X      ;ZEICHEN DEM BUFFER ENTNEHMEN
5C28 E8         0110          INX              ;PONTER ERHOEHEN
5C29 8E36A4     0111          STX TAPTR        ;UND POINTER WIEDER ABLEGEN
5C2C AEE25B     0112          LDX SAVX         ;INDEXREGISTER WIEDER HERSTELLEN
5C2F ACE35B     0113          LDY SAVY
5C32 60         0114          RTS
```

FEHLER IM USER-INPUT

```
5C33            0116 USER
5C33 4CF35D     0117          JMP ERMS         ;VERZWEIGEN ZUM AUSDRUCK FEHLERMELDUNG
```

USER WRITE

```
5C36            0119 UWRT
5C36 B020       0120          BCS UWC          ;WENN CARRY GESETZT=ZEICHN SCHREIBEN
5C38 2C0080     0121          BIT STR          ;SCHREIBSCHUTZ TESTEN
5C3B 5004       0122          BVC UW10         ;NICHT SCHREIBGESCHUETZT=UEBERSPRINGEN
5C3D A2FE       0123          LDX #$FE         ;KEY FUER SCHREIBSCHUTZ SETZEN
5C3F D0F2       0124          BNE USER
5C41            0125 UW10                       ;SCHREIBEN ERLAUBT
5C41 20CB5C     0126          JSR OPEN         ;UND DATEI EROEFFNEN
5C44 D0ED       0127          BNE USER         ;FEHLER BEI OPEN
5C46 A200       0128          LDX #$00         ;SCHREIBPOINTER INITIALISIEREN
5C48 8E37A4     0129          STX TAPTR2
5C4B 8EE85B     0130          STX T1           ;VOR DEM SCHREIBEN LETZTE WERTE
5C4E 8EE95B     0131          STX T2           ;AUF NULL SETZEN
5C51 8EEA5B     0132          STX T3
5C54 8EEB5B     0133          STX ECNT         ;EBENSO EOF-COUNTER FUER 256 VORBESTZEN
5C57 60         0134          RTS
5C58            0135 UWC                        ;EIN ZEICHEN SCHREIBEN
5C58 8EE25B     0136          STX SAVX         ;INDEX-REGISTER SICHERN
5C5B 8CE35B     0137          STY SAVY
```

```
5C5E ADE95B  0138          LDA T2          ;LETZE AUSGEBENE ZEICHEN VERSCHIEBEN
5C61 8DEA5B  0139          STA T3
5C64 ADE85B  0140          LDA T1
5C67 8DE95B  0141          STA T2
5C6A 68      0142          PLA             ;ZEICHEN VOM STACK HOLEN
5C6B 8DE85B  0143          STA T1          ;UND IN LETZTEM ZEICHEN ABLEGEN
5C6E AE37A4  0144          LDX TAPTR2      ;BUFFERPOINTER HOLEN
5C71 9D005B  0145          STA TABUF,X     ;UND ZEICHEN IM BUFFER ABLEGEN
5C74 E8      0146          INX             ;POINTER ERHOEHEN
5C75 8E37A4  0147          STX TAPTR2      ;UND POINTER WIEDER IM SPEICHER ABLEGEN
5C78 100A    0148          BPL UW90        ;POINTER <128, BUFFER NOCH NICHT VOLL
5C7A 20395D  0149          JSR WNS         ;UND BUFFER IN NAECHSETN SEKTOR SCHREIB
5C7D D0B4    0150          BNE USER
5C7F A200    0151          LDX #00
5C81 8EEB5B  0152          STX ECNT        ;EOF-COUNTER WIEDER FUER 256 ZAEHLUNGEN
5C84         0153 UW90
5C84         0154 EOF                       ;AUF EOF PRUEFEN
5C84 CEEB5B  0155          DEC ECNT        ;EOF-COUNTER WIRKT NUR BEI VORBESTZUNG
5C87 F010    0156          BEQ EF70        ;ER IST AUF NULL=DIE AUSGABE IST ZU ENI
5C89 ADE85B  0157          LDA T1          ;FUER EOF-PRUEFUNG LETZTES ZEICHEN
5C8C C91A    0158          CMP #$1A        ;CNTL Z EOF BEI BASIC
5C8E F009    0159          BEQ EF70
5C90 C90D    0160          CMP #CR         ; 2 CR=EOF BEI EDITOR
5C92 D018    0161          BNE EF80
5C94 CDE95B  0162          CMP T2
5C97 D013    0163          BNE EF80
5C99         0164 EF70                      ;DIES WAR DAS LETZTE ZEICHEN
5C99 A900    0165          LDA #$00        ;SEKTORREST MIT NULL FUELLEN
5C9B         0166 EF75
5C9B 9D005B  0167          STA TABUF,X     ;SEKTOR MIT NULL AUFFUELLEN
5C9E E8      0168          INX
5C9F 10FA    0169          BPL EF75
5CA1 20395D  0170          JSR WNS         ;UND LETZTEN SEKTOR SCHREIBEN
5CA4 D08D    0171          BNE USER
5CA6 20155E  0172          JSR DOFF        ;LAUFWERK WIEDER ZUR RUHE BRINGEN
5CA9 4CC45C  0173          JMP EF90        ;UND ZUM ENE ALLEN SCHREIBENS GEHEN
5CAC         0174 EFB0                      ;PRUEFUNG AUF ANFANG DES LETZTEN BLOCK
5CAC ADEA5B  0175          LDA T3          ;DIESER BESTEHT AUS SEMIKOLON UND
5CAF C93B    0176          CMP #SEMI
5CB1 D011    0177          BNE EF90
5CB3 ADE95B  0178          LDA T2          ;LAENGENANGABE NULL
5CB6 C930    0179          CMP #NULL
5CB8 D00A    0180          BNE EF90
5CBA CDE85B  0181          CMP T1
5CBD D005    0182          BNE EF90
5CBF A908    0183          LDA #08         ;D. LETZTE BLOCK ERKANNT, ES FOLGEN NOC
5CC1 8DEB5B  0184          STA ECNT        ;8 ZEICHEN, DIESE ANZAHL IN EOF-COUNTER
5CC4         0185 EF90                      ;HIER ENDET ALLES SCHREIBEN
5CC4 AEE25B  0186          LDX SAVX        ;MIT DER WIEDERHERELLUNG DER REGISTER
5CC7 ACE35B  0187          LDY SAVY
5CCA 60      0188          RTS
-------------------------------------------------------------------
OPEN FILE

5CCB         0190 OPEN
5CCB AD0080  0191          LDA STR         ;MOTOR ON STZEN
5CCE 20125D  0192          JSR DNAM        ;HOLEN FILE-NAME IN FCB
5CD1 20095E  0193          JSR DON
5CD4         0194 OPENP                     ;OPEN FUER PROGRAMM
5CD4 207D5D  0195          JSR RDSO        ;LESEN SPUR 0 SEKTOR 1
5CD7 D035    0196          BNE OPB0        ;LESEN FEHLERHAFT DIREKT ZURUECK
5CD9         0197 OP05                      ;LESEN OK SUCHEN FILE IN DIRECTORY
5CD9 A200    0198          LDX #$00        ;LADEN BUFFER-POINTER
5CDB         0199 OP10
5CDB A000    0200          LDY #$00        ;LADEN FCB-POINTER FUER SUCHE
5CDD         0201 OP20
5CDD BD005B  0202          LDA TABUF,X     ;VERGL. FILENAME IN PUFFER UND FCB
5CE0 D9F05B  0203          CMP FCB,Y
5CE3 D008    0204          BNE OP30        ;UNGLEICH NOCH NICHT GEFUNDEN
```

```
5CE5 E8       0205          INX              ;GLEICH=POINTER ERHOEHEN
5CE6 C8       0206          INY
5CE7 C004     0207          CPY #$04         ;GANZR FILE-NAME VERGLICHEN
5CE9 F00F     0208          BEQ OP40         ;JA, WIR HABEN IHN
5CEB D0F0     0209          BNE OP20         ;NEIN, WEITER VERGLEICHEN
5CED          0210 OP30                       ;FILE NOCH NICHT GEFUNDEN
5CED 8A       0211          TXA              ;BUFFER-POINTER
5CEE 2970     0212          AND #$70         ; AUF ANFANG DES EINTRAGES SETZEN
5CF0 C970     0213          CMP #$70 UND GEGEN HOECHSTWERT VERGLEICHEN
5CF2 F01B     0214          BEQ OPER         ;HOECHSTWERT ERRE.=FILE NICHT GEFUNDEN
5CF4 18       0215          CLC
5CF5 6910     0216          ADC #$10         ;UM EINEN EINTRAG ERHOEHEN
5CF7 AA       0217          TAX              ;UND ZURUECK IN POINTER
5CF8 D0E1     0218          BNE OP10         ;WEITER SUCHEN
5CFA          0219 OP40                       ;WIR HABEN IHN
5CFA 8A       0220          TXA              ;BUFFER-POINTER
5CFB 2970     0221          AND #$70         ;ZUERST AUF ANGANG DES EINTRAGES,
5CFD 090A     0222          ORA #$0A         ;DANN AUF PLATZANGABE SETZEN
5CFF AA       0223          TAX              ;UND ZURUECK IN POINTER
5D00 A0FA     0224          LDY #$FA         ;INDEX FUER PLATZANGABE IN FCB LADEN
5D02          0225 OP50
5D02 BD005B   0226          LDA TABUF,X      ;UND VOM PUFFER
5D05 99005B   0227          STA TABUF,Y      ;IN DEN FCB UEBERTRAGEN
5D08 E8       0228          INX
5D09 C8       0229          INY
5D0A 30F6     0230          BMI OP50
5D0C A200     0231          LDX #00          ;ALLES OK X LOESCHEN
5D0E          0232 OP80                       ;OPEN ERLEDIGT
5D0E 60       0233          RTS
-----------------------------------------------------------------
OPEN ERROR

5D0F          0235 OPER
5D0F A2FF     0236          LDX #$FF         ;OPEN-FEHLER-CODE LADEN
5D11 60       0237          RTS              ;UND ZUREUCKGEHEN
-----------------------------------------------------------------
DISK NAME

5D12          0239 DNAM
5D12 A200     0240          LDX #$00         ;HOLEN FILE-NAME MIT MONITOR-ROUTINE
5D14 20A2E8   0241          JSR FNAM
5D17 A005     0242          LDY #$05
5D19          0243 DN10                       ;UND DIESEN FILE-NAME IM FCB ABLEGEN
5D19 B92EA4   0244          LDA NAME,Y
5D1C 99F05B   0245          STA FCB,Y
5D1F 88       0246          DEY
5D20 10F7     0247          BPL DN10
5D22 60       0248          RTS
-----------------------------------------------------------------
READ NEXT SEKTOR

5D23          0250 RNS
5D23 204F5D   0251          JSR EINS         ;SEKTOS EINSTELLENR EINS
5D26 D010     0252          BNE RN95         ;FEHLER BEI EINSTELLEN=DIREKT ZURUECK
5D28 ADFD5B   0253          LDA ASEC         ;AKTUELLEN SEKTOR LADEN
5D2B 20825D   0254          JSR RDS          ;SEKTOR LESEN
5D2E D008     0255          BNE RN95         ;LESEFEHLER  VOM SEKTOR=DIREKT ZURUECK
5D30 206A5D   0256          JSR INCS         ;SEKTORNUMMER ERHOEHEN
5D33 A200     0257          LDX #$00         ;UND POINTER AUF SEKTORANFANG SETZEN
5D35 8E36A4   0258          STX TAPTR
5D38          0259 RN95
5D38 60       0260          RTS
-----------------------------------------------------------------
WRITE NEXT SEKTOR

5D39          0262 WNS
5D39 204F5D   0263          JSR EINS         ;SEKTOR EINSTELLEN
5D3C D010     0264          BNE WN95         ;FEHLER BEIM EINSTELLEN=DIREKT BEENDEN
5D3E ADFD5B   0265          LDA ASEC         ;AKTUELLEN SEKTOR LADEN
5D41 20B35D   0266          JSR WRS          ;UND SHREIBEN
5D44 D008     0267          BNE WN95         ;FEHLER BEIM SCHREIBEN=DIREKT BEENDEN
```

```
5D46            0268 WN05
5D46 206A5D     0269          JSR INCS          ;SEKTORNUMMER ERHOEHEN
5D49 A200       0270          LDX #$00          ;BUFFERPOINTER WIEDER AUF ANFANG SETZEI
5D4B 8E37A4     0271          STX TAPTR2
5D4E            0272 WN95
5D4E 60         0273          RTS
-------------------------------------------------------------------
EINSTELLEN SEKTOR

5D4F            0275 EINS
5D4F ADFD5B     0276          LDA ASEC          ;LADEN SECTOR AUS FCB
5D52 CDFF5B     0277          CMP HSEC          ;UND GEGEN HOECHSTEWERT VERGLEICHEN
5D55 ADFC5B     0278          LDA ATRK          ;DAS SELBE MIT TRACK
5D58 EDFE5B     0279          SBC HTRK
5D5B 9004       0280          BCC EI10          ;NOCH IM FILE
5D5D A2FD       0281          LDX #$FD          ;AUSSERH. D. FILE-GRENZEN/FEHLER-CODE
5D5F D008       0282          BNE EI95          ;FEHLER=DIREKT ZURUECK
5D61            0283 EI10                        ;TRACK UND SEKTOR IM FILE
5D61 ADFC5B     0284          LDA ATRK          ;TRCK-NUMMER LADEN
5D64 20E45D     0285          JSR SEEK          ;UND SUCHE AUFSETZEN
5D67 A200       0286          LDX #00           ;KENNZEICHEN FUER FEHLERFREI STZEN
5D69            0287 EI95
5D69 60         0288          RTS
-------------------------------------------------------------------
INCREMENT SEKTORT

5D6A            0290 INCS
5D6A EEFD5B     0291          INC ASEC          ;SEKTORNUMMER ERHOEHEN
5D6D ADFD5B     0292          LDA ASEC          ;UND LADEN FUER
5D70 C911       0293          CMP #MSEC+1       ;VERGL. GEGEN WERT SEKTOREN PRO TRACK
5D72 3008       0294          BMI IN10          ;WERT KLEINER NICHTS BESONDERES
5D74 A901       0295          LDA #$01          ;WERT ERREICHT=WIEDER SEKTOR 1 LADEN
5D76 8DFD5B     0296          STA ASEC UND WIESER ABSPEICHERN
5D79 EEFC5B     0297          INC ATRK          ;UND TRACK-NUMMER ERHOEHEN
5D7C            0298 IN10
5D7C 60         0299          RTS
-------------------------------------------------------------------
READ SEKTOR 0

5D7D            0301 RDS0
5D7D 20275E     0302          JSR REST          ;RESTORE AUSFUEHEREN
5D80 A901       0303          LDA #$01          ; IN AKKU SEKTOR-NUMMER 1 LADEN
-------------------------------------------------------------------
READ SEKTOR

5D82            0305 RDS                         ;IN AKKU STEHT SEKTOR-NUMMER
5D82 8D0280     0306          STA SER           ;DIESE IN SEKTOR-REGISTER LADEN
5D85 A908       0307          LDA #TRY          ;ANZAHL DER LESEVERSUCHE LADEN
5D87 8DE05B     0308          STA TCNT
5D8A            0309 RD05
5D8A A984       0310          LDA #%10000100    ;LESE-BEFEHL AUFSETZEN
5D8C A200       0311          LDX #$00          ;PUFFER-POINTER IN INDEX LADEN
5D8E 20335E     0312          JSR ACMD
5D91            0313 RD20                        ;ANGENOMMEN=LESEN IN AKTION
5D91 AD0080     0314          LDA STR           ; 4 USEC STATUS ABFRAGEN
5D94 4A         0315          LSR A             ; 2 USEC
                0316 und 0317: siehe unten
5D98 90F7       0318          BCC RD20          ; 2/3 USEC DATA-FLAG GEL.=KEINE DATEN
5D9A AD0380     0319          LDA DAR           ; 4 USEC DATEN VORHANDEN=DIESE LESEN
5D9D 9D005B     0320          STA TABUF,X       ; 5 USEC UND IN PUFFER ABLEGEN
5DA0 E8         0321          INX               ; 2 USEC UND PUFFER-POINTER ERHOEHEN
DA1 D0EE     0322          BNE RD20          ; 2/3 USEC WEITER MIT NAECHSETM BYTE
5DB3.           0323 RD90                        ;BUSY-FLAG GEL.=IM NORMALFALL SEKTOR G
5DA3 AD0080     0324          LDA STR
5DA6 D004       0325          BNE RDER          ;STATUS-REGISTER <> NULL=LESEN WIEDERH.
5DA8 E080       0326          CPX #$80
5DAA F005       0327          BEQ RD95          ;PUFFER-POINTER NICHT $80=LESEN WIEDERH
-------------------------------------------------------------------
READ FEHLER          Einfügung: 5D95 90F7    0316  BCC RD90  2/3 USEK BUSY-FLAG Gelöscht-
                                5D97 4A      0317  LSR A ;2 USEK     //Sektor gelesen
5DAC            0329 RDER
```

```
5DAC CEE05B  0330          DEC TCNT          ;ANZAHL LESEVERSUCHE VERMINDERN
5DAF D0D9    0331          BNE RD05          ;NOCH NICHT ALLE VERSUCHE=NOCH EINMAL
5DB1         0332 RD95                        ;ALLES VERGEBENS
5DB1 AA      0333          TAX               ;X MIT STATUS LN
5DB2 60      0334          RTS
-------------------------------------------------------------
WRITE SEKTOR

             0336 WRS                         ;IN PUFFER SEKTORINHALT, IN AKKU SEKTO
5DB3 8D0280  0337          STA SER           ;DIESE IN SEKTOR-REGISTER LADEN
5DB6 A908    0338          LDA #TRY          ;ANZAHL DER SCHREIBVERSUCHE LADEN
5DB8 8DE05B  0339          STA TCNT
5DBB         0340 WR05
5DBB A9A4    0341          LDA #%10100100    ;WRITE-BEFEHL AUFSETZEN
5DBD A200    0342          LDX #$00          ;PUFFER-ZEIGER LADEN
5DBF 20335E  0343          JSR ACMD
5DC2         0344 WS20                        ;ANGENOMMEN
5DC2 AD0080  0345          LDA STR           ; 4 USEC STATUS ABFRAGEN
5DC5 4A      0346          LSR A             ; 2 USEC
5DC6 900C    0347          BCC WS90          ; 2/3 USEC BUSY-FLAG GEL.=SEKTOR GESCHR
5DC8 4A      0348          LSR A             ; 2 USEC
5DC9 90F7    0349          BCC WS20          ; 2/3 USEC DATA-FLAG GEL.=K. DATEN ERF.
5DCB BD005B  0350          LDA TABUF,X       ; 5 USEC DATA-FLAG GESETZT=DATEN ERF.
DCE 8D0380   0351          STA DAR           ; 4 USEC UND IN DATA-REGISTER ABLEGEN
5DD1 E8      0352          INX               ; 2 USEC PUFFER-POINTER ERHOEHEN
5DD2 D0EE    0353          BNE WS20          ; 2/3 USEC U. NAECHSTES BYTE ABARBEITEN
5DD4         0354 WS90                        ;SEKTOR GESCHRIEBEN
5DD4 AD0080  0355          LDA STR           ;STATUS LESEN
5DD7 D004    0356          BNE WRER          ;UNGLEICHNULL=FEHLER
5DD9 E080    0357          CPX #$80          ;BUFFERPOINTER AUF RICHT. LAENGE PRUEFE
5DDB F005    0358          BEQ WR95          ;ALES OK=ZUM ENDE
-------------------------------------------------------------
WRITE ERROR

5DDD         0360 WRER
5DDD CEE05B  0361          DEC TCNT          ;ZAEHLER ANZAHL VERSUCHE ERNIEDRIGEN
5DE0 D0D9    0362          BNE WR05          ;WENN NICHT NULL NOCH EINMAL
5DE2         0363 WR95
5DE2 AA      0364          TAX               ;SATUS INX UEBERTRAGEN
5DE3 60      0365          RTS
-------------------------------------------------------------
ZUGREIFEN AUF SEKTOR

5DE4         0367 SEEK                        ;IN AKKU TRACK-NUMMER
5DE4 8D0380  0368          STA DAR           ;DIESE IN DATA-REGISTER ABLEGEN
5DE7 A913    0369          LDA #%00010011    ;SEEK-COMMAND AUF TRACK AUFSETZEN
5DE9 20335E  0370          JSR ACMD
5DEC         0371 SE20                        ;SUCHE IM GANG
5DEC AD0080  0372          LDA STR           ;STATUS LADEN
5DEF 4A      0373          LSR A
5DF0 B0FA    0374          BCS SE20          ;BUSY-FLAG GESETZT=TRACK NOCH N. GEF.
5DF2 60      0375          RTS               ;BUSY-FLAG GEL.=SCHREIBKOPF AUF TRACK
-------------------------------------------------------------
ERROR HANDLING


-------------------------------------------------------------
ERLAUBTE KEYS

5DF3         0378 ; FD  ZUGRIFF AUSSERHALB FILE-GRENZEN
5DF3         0379 ; FE  DISKETTE SCHREIBGESCHUETZT
5DF3         0380 ; FF  OPEN-FEHLER-FILE NOT FOUND
5DF3         0381 ERMS

5DF3 2001E9  0382          JSR OUTLOW        ;OUTPUT AUF DEFAULT STELLEN
5DF6 2013EA  0383          JSR CRLOW         ;ZEILENVORSCHUB AUSGEBEN
5DF9 203BE8  0384          JSR BLANK2        ;2 LEERZEICHEN
5DFC 8A      0385          TXA               ;ARBEITSERGEBNISS IN AKKU ZURUECKBRINGE
5DFD 2046EA  0386          JSR NUMA          ;UND ALS HEXA AUSGEBEN
5E00 2094E3  0387          JSR CKER00        ;ERROR-LOGO AUSGEBEN
5E03 20155E  0388          JSR DOFF          ;UND LAUFWERK ABSCHALTEN
5E06 4C82E1  0389          JMP START         ;ZURUECK ZUM AIM-MONITOR
```

```
-------------------------------------------------------------
DISK ON

5E09            0391 DON
5E09 AE34A4     0392        LDX TAPIN       ;TAPE-NUMMER DURCH FNAM GESETZT HOLEN
5E0C BD135E     0393        LDA DETAB,X     ;LAUFWERKSANSCHALTUNG AUS TABELLE
5E0F BD00A8     0394        STA DRB         ;UND DRIVE-CONTROL-REGISTER LADEN
5E12 60         0395        RTS             ;DAMIT IST LAUFWERK GESTARTRT UND SELEK
5E13 14         0396 DETAB  .BYT $14,$24    ;TABELLE ZUR LAUFWERKANST.
5E14 24         0396
-------------------------------------------------------------
DISK OFF

5E15            0398 DOFF
5E15 20275E     0399        JSR REST        ;RESTORE AUSFUEHREN
5E18            0400 DF10
5E18 AD0080     0401        LDA STR         ;WARTEN BIS COMMANDO ERLEDIGT
5E1B 4A         0402        LSR A
5E1C B0FA       0403        BCS DF10
5E1E A904       0404        LDA #04
5E20 8D00A8     0405        STA DRB         ;UND DCR AUSSCHALTEN
5E23 AD0083     0406        LDA $8300
5E26 60         0407        RTS
-------------------------------------------------------------
RESTORE BEFEHL

5E27            0409 REST
5E27 A903       0410        LDA ##03        ;COMMANDO SETZEN
5E29 20335E     0411        JSR ACMD
5E2C            0412 RS10
5E2C AD0080     0413        LDA STR
5E2F 4A         0414        LSR A
5E30 B0FA       0415        BCS RS10
5E32 60         0416        RTS
-------------------------------------------------------------
ACCEPT COMMAND

5E33            0418 ACMD
5E33 8D0080     0419        STA CMR         ;COMMAND REGISTER LADEN
5E36            0420 AC10
5E36 AD0080     0421        LDA STR         ;STATUS REGISTEN ABFRAGEN
5E39 4A         0422        LSR A
5E3A 90FA       0423        BCC AC10        ;UND WARTEN BIS BEFEHL AKZEPTIERT
5E3C 60         0424        RTS
-------------------------------------------------------------
BEFEHLE DER PROGRAMM-SCHNITTSTELLE

5E3D            0426 PCMD
5E3D BA         0427        TSX             ;STACK-POINTER IN INDEX UEBERTRAGEN
5E3E BD0101     0428        LDA $101,X      ;VOM STACK POINTER A. PARAMETERTABELLE
5E41 85FA       0429        STA PPTR        ;DES BEFEHLES HOLEN UND IN
5E43 BD0201     0430        LDA $102,X      ;PARAMETER-POUNTER ABLEGEN
5E46 85FB       0431        STA PPTR+1
5E48 A003       0432        LDY ##03        ;LAENGE PARAMETERFELD SETZEN
5E4A            0433 PC10
5E4A B1FA       0434        LDA (PPTR),Y    ;PARAMETER IN PAGE ZERO ABLEGEN
5E4C 99FC00     0435        STA SPTR,Y
5E4F FE0101     0436        INC $101,X      ;STACK UM ENTNOMMENE BYTES KORRIGIEREN
5E52 D003       0437        BNE PC20
5E54 FE0201     0438        INC $102,X
5E57            0439 PC20
5E57 88         0440        DEY
5E58 D0F0       0441        BNE PC10
5E5A A5FD       0442        LDA SPTR+1
5E5C 48         0443        PHA             ;COMMANDO AUF STACK SICHERN
5E5D 20CF5E     0444        JSR GFCB        ;UND FCB IN ARBEITSBEREICH HOLEN
5E60 68         0445        PLA             ;COMMANDO WIEDER HOLEN
5E61 C94F       0446        CMP #'O'        ;IST ES OPEN
5E63 F012       0447        BEQ POPEN
5E65 C952       0448        CMP #'R'        ;ODER READ
5E67 F015       0449        BEQ PREA
```

```
5E69 C957     0450          CMP #'W'          ;ODER WRITE
5E6B F01B     0451          BEQ PWRT
5E6D C953     0452          CMP #'S'          ;ODER SEEK
5E6F F021     0453          BEQ PSEEK
5E71 A9C0     0454          LDA ##$C0         ;CODE FUER KEIN GUELTIGES COMMANDO
5E73 8DF75B   0455          STA FCBS
5E76 60       0456          RTS               ;UND ZURUECK ZUM AUFRUFENDEN BENUTERPRO
-------------------------------------------------------------
OPEN UEBER PROGRAMM

5E77          0458 POPEN
5E77 20D45C   0459          JSR OPENP         ;PROGRAMMIERTES OPEN ABFAHREN
5E7A 20E45E   0460          JSR PFCB          ;UND KOMMPLETIERTEN FCB INS RPROGRAMM
5E7D 60       0461          RTS
-------------------------------------------------------------
READ UEBER PROGRAMM

5E7E          0463 PREA
5E7E 20235D   0464          JSR RNS           ;NAECHSTEN SEKTOR LESEN
5E81 20FA5E   0465          JSR PSEC          ;DEM BENUTZERPROGRAMM ZUR VERFUEGUNG
5E84 20E45E   0466          JSR PFCB          ;EBENSO FCB IN PROG)AMM ZURUECKBRINGEN
5E87 60       0467          RTS
-------------------------------------------------------------
WRITE UEBER PROGRAMM

5E88          0469 PWRT
5E88 20EF5E   0470          JSR FSEC          ;SECTOR AUS BENUTZERPROGRAMM IN PUFFER
5E8B 20395D   0471          JSR WNS           ;SECTOR SCHREIBEN
5E8E 20E45E   0472          JSR PFCB          ;FCB IN PROGRAMM ZURUECKSTELLEN
5E91 60       0473          RTS
-------------------------------------------------------------
SEEK UEBER PROGRAMM

5E92          0475 PSEEK
5E92 ADFC5B   0476          LDA ATRK          ;TRACK UND SECTORNUMMER IN TEMPORAERSP.
5E95 8DEC5B   0477          STA TTRK          ;ABSTELLEN
5E98 ADFD5B   0478          LDA ASEC
5E9B 8DED5B   0479          STA TSEC
5E9E ADFA5B   0480          LDA VTRK          ;VON-TRACK UND SECTOR IN
5EA1 8DFC5B   0481          STA ATRK          ;AKTUELLEN TRACK UND SEKTOR KOPIEREN
5EA4 ADFB5B   0482          LDA VSEC
5EA7 8DFD5B   0483          STA ASEC
5EAA          0484 PT10
5EAA 38       0485          SEC
5EAB ADED5B   0486          LDA TSEC          ;JETZT ZUERST TRACK DES GESUCHTEN SEKTO
5EAE E910     0487          SBC #MSEC         ;ERMITTELN D. WIEDERHOLTES SUBTRHIEREN
5EB0 8DED5B   0488          STA TSEC          ;DER ANZAHL SEKTOREN PRO TACK
5EB3 ADEC5B   0489          LDA TTRK          ;TTRK ENHAELLT DABEI HOEHERWERTIGE
5EB6 E900     0490          SBC #00           ;BITS DER RELATIVEN SEKTORNUMMER
5EB8 8DEC5B   0491          STA TTRK
5EBB 3005     0492          BMI PT20          ;HIER WURDE TRACK  ERMITTELT
5EBD EEFC5B   0493          INC ATRK          ;NOCH NICHT AUF NULL ABSOLUTE TRACK ERH
5EC0 D0E8     0494          BNE PT10          ;UND WEITER ABZIEHEN
5EC2          0495 PT20                        ;TRACK GEFUNDEN
5EC2 38       0496          SEC
5EC3 ADED5B   0497          LDA TSEC          ;NACHDEM EINGESTELLT WURDE
5EC6 6910     0498          ADC #MSEC         ;KORREKTURADDITION DURCHFUEHREN
5EC8 8DFD5B   0499          STA ASEC          ;UND SEKTOR-ABSOLUT SETEZEN
5ECB 20E45E   0500          JSR PFCB          ;FCB IN BENUTZERBEREICH ZUREUCKBRINGEN
5ECE 60       0501          RTS
-------------------------------------------------------------
GET USER FCB

5ECF          0503 GFCB
5ECF A00F     0504          LDY ##$0F         ;LAENGE LADEN
5ED1          0505 GF10
5ED1 B1FE     0506          LDA (FPTR),Y      ;UND FCB INDIREKT AUS PROGRAMM
5ED3 99F05B   0507          STA FCB,Y         ;IN ARBEITSSPEICHER UEBERTRAGEN
5ED6 88       0508          DEY
5ED7 10F8     0509          BPL GF10
5ED9 ADF85B   0510          LDA SADR          ;JETZT SEKTORADRESSE AUS BENUTZERPROGRA
```

```
5EDC 85FC     0511          STA SPTR          ;DEM FCB ENTN. UND IN PAGE ZERO POINTER
5EDE ADF95B   0512          LDA SADR+1        ;ABLEGEN
5EE1 85FD     0513          STA SPTR+1
5EE3 60       0514          RTS
-------------------------------------------------------------------
PUT USER FCB

5EE4          0516 PFCB
5EE4 A00F     0517          LDY #$0F          ;LAENGE LADEN
5EE6          0518 PF10
5EE6 B9F05B   0519          LDA FCB,Y         ;UND FCB AUS ARBEITSSPEICHER ENTNEHEM
5EE9 91FE     0520          STA (FPTR),Y      ;MIT POINTER IM BENUTZERPROGRAMM
5EEB 88       0521          DEY               ;ABLEGEN
5EEC 10F8     0522          BPL PF10
5EEE 60       0523          RTS
-------------------------------------------------------------------
HOLEN SEKTOR

5EEF          0525 FSEC
5EEF A07F     0526          LDY #$7F          ;SEKTORLAENGE LADEN
5EF1          0527 GS10
5EF1 B1FC     0528          LDA (SPTR),Y      ;SEKTORINHALT AUS BENUTZERPROGRAMM
5EF3 99005B   0529          STA TABUF,Y       ;UND IN SEKTORPUFFER ABLEGEN
5EF6 88       0530          DEY
5EF7 10F8     0531          BPL GS10
5EF9 60       0532          RTS
-------------------------------------------------------------------
ABLEGEN SEKTOR

5EFA          0534 PSEC
5EFA A07F     0535          LDY #$7F          ;SEKTORLAENGE LADEN
5EFC          0536 PS10
5EFC B9005B   0537          LDA TABUF,Y       ;SEKTORINHALT AUS DEM SEKTORPUFFER
5EFF 91FC     0538          STA (SPTR),Y      ;UND IM BENUTZERPROGRAMM ABLEGEN
5F01 88       0539          DEY
5F02 10F8     0540          BPL PS10
5F04 60       0541          RTS
-------------------------------------------------------------------
SPRUNGPUNKTE

5F05          0543          *=$5FE0
5FE0 4C125D   0544 JDNAM    JMP DNAM
5FE3 4CE45D   0545 JSEEK    JMP SEEK
5FE6 4CB35D   0546 JWRS     JMP WRS
5FE9 4C825D   0547 JRDS     JMP RDS
5FEC 4C7D5D   0548 JRDS0    JMP RDS0
5FEF 4C275E   0549 JREST    JMP REST
5FF2 4C095E   0550 JDON     JMP DON
5FF5 4C155E   0551 JDOFF    JMP DOFF
5FF8 4C3D5E   0552 JPCMD    JMP PCMD
5FFB          0553          .END

  ERRORS= 0000
```

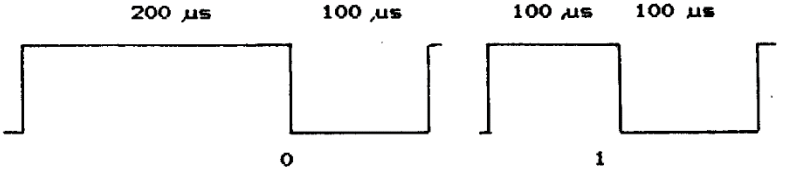
Bücher

Koch, J.: Der 16bit-Mikroprozessor SC68000 - Befehlsvorrat. Valvo-Buch im Verlag Boysen und Maasch, Hamburg 1983, ISBN 3-87095-257-1, 200 S., DM 24,--. Es ist das zweite Valvo-Buch zum 68000 in sorgfältiger, lesbarer deutscher Übersetzung amerikanischer Quellen, vor allem der Motorola-Schrift Dokument MC68000UM(AD) (User's Manual). Den Hauptteil nehmen die Übersichten für Befehlsgruppen und für die Einzelbefehle ein. Weitere Abschnitte beschreiben die Register und die Datenorganisation, Befehlsformate und Adressierungsarten, Ausnahmeverarbeitung und -vektoren, Berechnung der Bedingungscodes, Mikroprogramm-Architektur des 68000, Prefetch-Mechanismus und 12 Programmbeispiele. - Es ist ein Handbuch, das man bei Assemblerprogrammierung neben den Computer legt, um eine schnelle und übersichtliche Referenz zu haben. Es enthält keine 'Geheimnisse', die nicht auch schon von Motorola dokumentiert worden wären, es hat den Vorzug der Verständlichkeit für den deutschen Betreiber.

Ing. Louis Rutten, B-3980 Tessenderlo

Supertape für CBM

Based on the idea "supertape" (Micro Mag Nr 31), following program runs on CBM3000/4000 and saves a 4 k block of memory on tape in 12 seconds (included 3 s wait-time at the beginning).
I shooted it into an EPROM, used it for two months and never had a load-error.
The bits are saved as shown on the figure.



Loading, the bits are detected by measuring the time between two Low-High transitions.
If t < 250 µs the bit is 1, if t > 250 µs the bit is 0.

Running the program for SAVE, the program waits for 3 s, after the buttons are pressed. Then 256 sync. pulses are sent, followed by a first mark and the name. Again 256 sync pulses are sent, followed by a second mark, checksum, adresses and text.
At the end of the procedure (SAVE and LOAD) a jump follows
- in case of a M.L.-program: to MON (protecting the program for BASIC by setting the highest RAM-vector);
- in case of a BASIC-program: to BASIC READY (adjusting the endadress);
- in case of a load-error: to BRK.

FORMATS:

Start: SYS 11447 or JMP $2BC2
CBM answers with CLR and *

SAVE: M.L.: *S NAME,XXXX,YYYY
XXXX: startadress
YYYY: endadress+1

BASIC:*S NAME
The name will be not longer as 16 characters.

LOAD: *L
Loads the first program.
*L NAME
Loads the program named NAME.

```
.; 2BC2 A0 08     LDY #$08     INPUT BYTE
.; 2BC4 20 E5 2B  JSR $2BE5    GET A BIT
.; 2BC7 88        DEY
.; 2BC8 D0 FA     BNE $2BC4    FOR 8 TIMES
.; 2BCA A5 BC     LDA $BC      BYTE IN A
.; 2BCC 60        RTS
.; 2BCD A2 0D     LDX #$0D     PRINT CR,*
.; 2BCF A9 2A     LDA #$2A
.; 2BD1 20 84 E7  JSR $E784    $D731
.; 2BD4 60        RTS
.; 2BD5 20 A7 E7  JSR $E7A7    GET 2 ADRESSES
.; 2BD8 20 97 E7  JSR $E797    $D744
.; 2BDB 20 EB E7  JSR $E7EB    $D798
.; 2BDE 20 A7 E7  JSR $E7A7    $D754
.; 2BE1 20 97 E7  JSR $E797
.; 2BE4 60        RTS
.; 2BE5 AD 12 E8  LDA $E812    GET A BIT
.; 2BE8 C9 EF     CMP #$EF     IF "STOP"
.; 2BEA D0 05     BNE $2BF1
.; 2BEC 68        PLA
.; 2BED 68        PLA
.; 2BEE 4C F8 F8  JMP $F8F8    $F93D BREAK
.; 2BF1 2C 11 E8  BIT $E811    WAIT FOR
.; 2BF4 10 EF     BPL $2BE5    LOW - HIGH TRANS.
.; 2BF6 AD 10 E8  LDA $E810    CLR FLAG
.; 2BF9 AD 44 E8  LDA $E844    GET TIME
.; 2BFC 85 BD     STA $BD
.; 2BFE 38        SEC          112 µS<T<112 µS?
.; 2BFF A5 BE     LDA $BE
.; 2C01 E5 BD     SBC $BD
.; 2C03 C9 70     CMP #$70     C=1 IF BIT=1
.; 2C05 66 BC     ROR $BC      SAVE BIT
.; 2C07 A5 BD     LDA $BD
.; 2C09 85 BE     STA $BE
.; 2C0B 60        RTS
.; 2C0C 85 BC     STA $BC      SEND BYTE
.; 2C0E A0 08     LDY #$08     FOR 8 BIT
.; 2C10 46 BC     LSR $BC
.; 2C12 20 31 2C  JSR $2C31    WAIT FOR 96 µS
.; 2C15 AD 40 E8  LDA $E840    ALTER OUTPUT
.; 2C18 49 08     EOR #$08
.; 2C1A 8D 40 E8  STA $E840
.; 2C1D 20 31 2C  JSR $2C31    WAIT FOR 96 µS
.; 2C20 B0 03     BCS $2C25    IF BIT=0
.; 2C22 20 31 2C  JSR $2C31    WAIT AGAIN
.; 2C25 AD 40 E8  LDA $E840
.; 2C28 49 08     EOR #$08     ALTER OUTPUT
.; 2C2A 8D 40 E8  STA $E840
.; 2C2D 88        DEY
.; 2C2E D0 E0     BNE $2C10
.; 2C30 60        RTS
.; 2C31 2C 4D E8  BIT $E84D    WAIT 96 µS
.; 2C34 50 FB     BVC $2C31
.; 2C36 AD 44 E8  LDA $E844
.; 2C39 60        RTS
.; 2C3A BD C0 2E  LDA $2EC0,X  PRINT MESSAGE
.; 2C3D 29 FF     AND #$FF
.; 2C3F 30 06     BMI $2C47
.; 2C41 91 C4     STA ($C4),Y
.; 2C43 E8        INX
.; 2C44 C8        INY
.; 2C45 D0 F3     BNE $2C3A
.; 2C47 A5 C4     LDA $C4      1 LINE DOWN
.; 2C49 69 28     ADC #$28
.; 2C4B 85 C4     STA $C4
.; 2C4D 90 02     BCC $2C51
.; 2C4F E6 C5     INC $C5
.; 2C51 A5 C5     LDA $C5
.; 2C53 C9 83     CMP #$83
.; 2C55 D0 0D     BNE $2C64
.; 2C57 A9 80     LDA #$80
.; 2C59 25 C5     AND $C5
.; 2C5B 85 C5     STA $C5
.; 2C5D 38        SEC
.; 2C5E A5 C4     LDA $C4
.; 2C60 E9 11     SBC #$11
.; 2C62 85 C4     STA $C4
.; 2C64 60        RTS
```

```
.; 2C65 20 CF FF    JSR $FFCF    GET CHAR.
.; 2C68 C9 2A       CMP #$2A     DROP * ,SP.
.; 2C6A F0 F9       BEQ $2C65
.; 2C6C C9 20       CMP #$20
.; 2C6E F0 F5       BEQ $2C65
.; 2C70 60          RTS
.; 2C71 60          RTS
.; 2C72 A0 00       LDY #$00     CHECKSUM
.; 2C74 A5 58       LDA $58
.; 2C76 18          CLC
.; 2C77 71 54       ADC ($54),Y
.; 2C79 90 02       BCC $2C7D
.; 2C7B E6 59       INC $59
.; 2C7D 85 58       STA $58
.; 2C7F E6 54       INC $54
.; 2C81 D0 02       BNE $2C85
.; 2C83 E6 55       INC $55
.; 2C85 A5 55       LDA $55
.; 2C87 C5 57       CMP $57
.; 2C89 D0 E9       BNE $2C74
.; 2C8B A5 54       LDA $54
.; 2C8D C5 56       CMP $56
.; 2C8F D0 E3       BNE $2C74
.; 2C91 A2 01       LDX #$01
.; 2C93 B5 FB       LDA $FB,X
.; 2C95 95 54       STA $54,X
.; 2C97 CA          DEX
.; 2C98 10 F9       BPL $2C93
.; 2C9A 60          RTS
.; 2C9B 20 47 2C    JSR $2C47    WAIT BUTTON
.; 2C9E AD 10 E8    LDA $E810    PRESSED
.; 2CA1 29 10       AND #$10
.; 2CA3 F0 11       BEQ $2CB6
.; 2CA5 A2 05       LDX #$05
.; 2CA7 20 3A 2C    JSR $2C3A
.; 2CAA AD 10 E8    LDA $E810
.; 2CAD 29 10       AND #$10
.; 2CAF D0 F9       BNE $2CAA
.; 2CB1 A2 10       LDX #$10
.; 2CB3 20 3A 2C    JSR $2C3A
.; 2CB6 60          RTS
.; 2CB7 A9 93       LDA #$93     START
.; 2CB9 20 D2 FF    JSR $FFD2    CLR
.; 2CBC 20 CD 2B    JSR $2BCD    PRINT CR, *
.; 2CBF A2 00       LDX #$00     CHSUM=0
.; 2CC1 86 59       STX $59
.; 2CC3 86 58       STX $58
.; 2CC5 20 65 2C    JSR $2C65    GET CHAR.
.; 2CC8 C9 4C       CMP #$4C     IF L
.; 2CCA F0 04       BEQ $2CD0
.; 2CCC C9 53       CMP #$53     OR S
.; 2CCE D0 1A       BNE $2CEA
.; 2CD0 A8          TAY          SAVE CHAR.
.; 2CD1 20 CF FF    JSR $FFCF    GET NAME
.; 2CD4 C9 20       CMP #$20
.; 2CD6 F0 F9       BEQ $2CD1
.; 2CD8 C9 0D       CMP #$0D     IF CR THEN DONE
.; 2CDA F0 11       BEQ $2CED
.; 2CDC C9 2C       CMP #$2C     IF "," THEN DONE
.; 2CDE F0 0D       BEQ $2CED
.; 2CE0 29 3F       AND #$3F
.; 2CE2 9D 09 02    STA $0209,X
.; 2CE5 E8          INX
.; 2CE6 E0 10       CPX #$10     NO MORE THEN
.; 2CE8 D0 E7       BNE $2CD1    16(D) CHAR.
.; 2CEA 4C 56 FD    JMP $FD56    $D4BA TO MON.
.; 2CED 86 5A       STX $5A      LENGTH NAME
.; 2CEF C0 4C       CPY #$4C     IF L THEN LOAD
.; 2CF1 F0 13       BEQ $2D06
.; 2CF3 E0 00       CPX #$00     SAVE
.; 2CF5 F0 F3       BEQ $2CEA    NO NAME, TO MON.
.; 2CF7 C9 2C       CMP #$2C     IF ","
.; 2CF9 F0 0E       BEQ $2D09    THEN MACH.LANG.
.; 2CFB A2 04       LDX #$04     BASIC ADRESSES
.; 2CFD B5 27       LDA $27,X
.; 2CFF 95 FA       STA $FA,X
.; 2D01 CA          DEX
```

```
2D02 D0 F9      BNE $2CFD
2D04 F0 06      BEQ $2D0C
2D06 4C DE 2D   JMP $2DDE
2D09 20 D5 2B   JSR $2BD5    M.L. ADRESSES
2D0C A2 04      LDX #$04
2D0E B5 FA      LDA $FA,X
2D10 95 53      STA $53,X
2D12 CA         DEX
2D13 D0 F9      BNE $2D0E
2D15 20 72 2C   JSR $2C72    CHECKSUM
2D18 20 9B 2C   JSR $2C9B    BUTTON PRESSED?
2D1B 78         SEI          SAVE
2D1C AD 42 E8   LDA $E842    BIT 3 IS OUTPUT
2D1F 09 08      ORA #$08
2D21 8D 42 E8   STA $E842
2D24 AD 40 E8   LDA $E840    BIT 3=0
2D27 29 F7      AND #$F7
2D29 8D 40 E8   STA $E840
2D2C A9 40      LDA #$40     T1 FREE RUN
2D2E 8D 4B E8   STA $E84B
2D31 A9 35      LDA #$35     START MOTOR
2D33 8D 13 E8   STA $E813
2D36 20 D3 2E   JSR $2ED3    WAIT FOR 3 S
2D39 A9 60      LDA #$60     T1=0060(H)
2D3B 8D 44 E8   STA $E844    OR 96 µS
2D3E 8E 45 E8   STX $E845
2D41 A9 16      LDA #$16     256(D) SYNC. PULS.
2D43 20 0C 2C   JSR $2C0C
2D46 CA         DEX
2D47 D0 F8      BNE $2D41
2D49 A9 2A      LDA #$2A     FIRST MARK
2D4B 20 0C 2C   JSR $2C0C
2D4E A5 5A      LDA $5A      LENGTH NAME
2D50 AA         TAX
2D51 20 0C 2C   JSR $2C0C
2D54 CA         DEX
2D55 BD 09 02   LDA $0209,X  NAME
2D58 EA         NOP
2D59 EA         NOP
2D5A 20 0C 2C   JSR $2C0C
2D5D 8A         TXA
2D5E D0 F4      BNE $2D54
2D60 A9 16      LDA #$16     256(D) SYNC.PULS.
2D62 20 0C 2C   JSR $2C0C
2D65 CA         DEX
2D66 D0 F8      BNE $2D60
2D68 A9 2B      LDA #$2B     SECOND MARK
2D6A 20 0C 2C   JSR $2C0C
2D6D A2 05      LDX #$05
2D6F B5 54      LDA $54,X    CHSUM AND ADRESSES
2D71 20 0C 2C   JSR $2C0C
2D74 CA         DEX
2D75 10 F8      BPL $2D6F
2D77 A0 00      LDY #$00     TEXT
2D79 B1 54      LDA ($54),Y
2D7B 20 0C 2C   JSR $2C0C
2D7E E6 54      INC $54
2D80 D0 02      BNE $2D84
2D82 E6 55      INC $55
2D84 A5 55      LDA $55
2D86 C5 57      CMP $57
2D88 D0 EF      BNE $2D79
2D8A A5 54      LDA $54
2D8C C5 56      CMP $56
2D8E D0 E9      BNE $2D79
2D90 A5 FB      LDA $FB      START ADRESS
2D92 20 0C 2C   JSR $2C0C
2D95 A5 FC      LDA $FC
2D97 20 0C 2C   JSR $2C0C
2D9A 20 0C 2C   JSR $2C0C    END BYTE
2D9D 20 7B FC   JSR $FC7B    $FCC0 END
2DA0 A9 01      LDA #$01     FLAG STOP MOTOR
2DA2 85 F9      STA $F9
2DA4 A2 05      LDX #$05
2DA6 20 3A 2C   JSR $2C3A    PRINT MESS.
2DA9 A2 10      LDX #$10     16(D) X DOWN
2DAB A9 0D      LDA #$0D
2DAD 20 D2 FF   JSR $FFD2
```

```
.; 2DB0 CA          DEX
.; 2DB1 10 FA       BPL $2DAD
.; 2DB3 A9 04       LDA #$04      IF BASICPR.
.; 2DB5 C5 FC       CMP $FC
.; 2DB7 D0 0C       BNE $2DC5
.; 2DB9 A5 56       LDA $56       ADJUST ENDADRESS
.; 2DBB 85 2A       STA $2A
.; 2DBD A5 57       LDA $57
.; 2DBF 85 2B       STA $2B
.; 2DC1 58          CLI           AND JMP TO
.; 2DC2 4C 39 C4    JMP $C439     $B4BF B. READY ; B4AD
.; 2DC5 A5 56       LDA $56       NOT USED
.; 2DC7 C9 F1       CMP #$F1
.; 2DC9 D0 08       BNE $2DD3
.; 2DCB A5 57       LDA $57
.; 2DCD C9 32       CMP #$32
.; 2DCF D0 02       BNE $2DD3
.; 2DD1 58          CLI
.; 2DD2 60          RTS
.; 2DD3 A5 FB       LDA $FB       HIGHEST RAMVECTOR
.; 2DD5 85 34       STA $34       (PROTECTION)
.; 2DD7 A5 FC       LDA $FC
.; 2DD9 85 35       STA $35
.; 2DDB 4C 56 FD    JMP $FD56     $D4BA TO MON.
.; 2DDE A0 00       LDY #$00      LOAD
.; 2DE0 78          SEI
.; 2DE1 A9 40       LDA #$40      T1 FREE  RUN
.; 2DE3 8D 4B E8    STA $E84B
.; 2DE6 A9 FF       LDA #$FF
.; 2DE8 8D 44 E8    STA $E844     T1=FFFF(H)
.; 2DEB 8D 45 E8    STA $E845
.; 2DEE A9 00       LDA #$00
.; 2DF0 85 5B       STA $5B       LENGHT NAME=0
.; 2DF2 85 CB       STA $CB       CHSUM=0
.; 2DF4 85 CC       STA $CC
.; 2DF6 20 9B 2C    JSR $2C9B     BUTT.PRESSED?
.; 2DF9 A9 35       LDA #$35      START MOTOR
.; 2DFB 8D 13 E8    STA $E813
.; 2DFE A9 06       LDA #$06      SET L-H TRANS.
.; 2E00 8D 11 E8    STA $E811
.; 2E03 20 47 2C    JSR $2C47
.; 2E06 AD 10 E8    LDA $E810     CLR FLAG
.; 2E09 20 E5 2B    JSR $2BE5     GET A BIT MORE
.; 2E0C A5 BC       LDA $BC       IN $BC
.; 2E0E C9 16       CMP #$16      NO SYNC.P.
.; 2E10 D0 F7       BNE $2E09     AGAIN
.; 2E12 A2 0A       LDX #$0A      AT LEAST
.; 2E14 20 C2 2B    JSR $2BC2     10(D)
.; 2E17 C9 16       CMP #$16      SYNC.P.
.; 2E19 D0 EE       BNE $2E09
.; 2E1B CA          DEX
.; 2E1C D0 F6       BNE $2E14
.; 2E1E 20 C2 2B    JSR $2BC2     DROP SYNC.P.
.; 2E21 C9 16       CMP #$16
.; 2E23 F0 F9       BEQ $2E1E
.; 2E25 C9 2A       CMP #$2A      IF FIRST MARK
.; 2E27 D0 40       BNE $2E69
.; 2E29 84 5C       STY $5C       REMEMBER IT
.; 2E2B 20 C2 2B    JSR $2BC2     GET LENGTH NAME
.; 2E2E 85 D4       STA $D4
.; 2E30 AA          TAX
.; 2E31 CA          DEX
.; 2E32 20 C2 2B    JSR $2BC2     GET NAME
.; 2E35 9D 19 02    STA $0219,X
.; 2E38 CA          DEX
.; 2E39 10 F7       BPL $2E32
.; 2E3B A2 00       LDX #$00      PRINT "VOND"
.; 2E3D 20 3A 2C    JSR $2C3A
.; 2E40 A4 D4       LDY $D4
.; 2E42 88          DEY
.; 2E43 B9 19 02    LDA $0219,Y   PRINT NAME
.; 2E46 91 C4       STA ($C4),Y
.; 2E48 88          DEY
.; 2E49 10 F8       BPL $2E43
.; 2E4B 20 47 2C    JSR $2C47
.; 2E4E A6 5A       LDX $5A       GO ON IF NO NAME
```

```
.;  2E50 F0 17      BEQ $2E69    TO SEARCH
.;  2E52 E4 D4      CPX $D4      IF RIGHT LENGHT
.;  2E54 D0 0E      BNE $2E64
.;  2E56 CA         DEX
.;  2E57 BD 19 02   LDA $0219,X  AND RIGHT NAME
.;  2E5A DD 09 02   CMP $0209,X
.;  2E5D D0 05      BNE $2E64
.;  2E5F CA         DEX
.;  2E60 10 F5      BPL $2E57
.;  2E62 30 02      BMI $2E66    GO ON
.;  2E64 E6 5C      INC $5C      OTHERW. SET FLAG
.;  2E66 4C 06 2E   JMP $2E06    AND AGAIN
.;  2E69 C9 2B      CMP #$2B     SECOND MARK?
.;  2E6B D0 99      BNE $2E06    NO, AGAIN
.;  2E6D A5 5C      LDA $5C      OTHER PR, AGAIN
.;  2E6F D0 95      BNE $2E06
.;  2E71 20 C2 2B   JSR $2BC2    INPUT CHSUM
.;  2E74 85 CC      STA $CC
.;  2E76 20 C2 2B   JSR $2BC2
.;  2E79 85 CB      STA $CB
.;  2E7B A2 03      LDX #$03     ADRESSES
.;  2E7D 20 C2 2B   JSR $2BC2
.;  2E80 95 54      STA $54,X
.;  2E82 CA         DEX
.;  2E83 10 F8      BPL $2E7D
.;  2E85 20 C2 2B   JSR $2BC2    TEXT
.;  2E88 91 54      STA ($54),Y
.;  2E8A E6 54      INC $54
.;  2E8C D0 02      BNE $2E90
.;  2E8E E6 55      INC $55
.;  2E90 A5 55      LDA $55
.;  2E92 C5 57      CMP $57
.;  2E94 D0 EF      BNE $2E85
.;  2E96 A5 54      LDA $54
.;  2E98 C5 56      CMP $56
.;  2E9A D0 E9      BNE $2E85
.;  2E9C 20 C2 2B   JSR $2BC2    START ADRESS
.;  2E9F 85 FB      STA $FB
.;  2EA1 85 54      STA $54
.;  2EA3 20 C2 2B   JSR $2BC2
.;  2EA6 85 FC      STA $FC
.;  2EA8 85 55      STA $55
.;  2EAA 20 72 2C   JSR $2C72
.;  2EAD A5 CC      LDA $CC      CMP CHSUM
.;  2EAF C5 59      CMP $59
.;  2EB1 D0 09      BNE $2EBC    LOAD ERROR
.;  2EB3 A5 CB      LDA $CB
.;  2EB5 C5 58      CMP $58
.;  2EB7 D0 03      BNE $2EBC
.;  2EB9 4C 9D 2D   JMP $2D9D    END
.;  2EBC 58         CLI
.;  2EBD 4C F7 E7   JMP $E7F7    $D7A4 PRINT "?"

.:  2EC0 16 0F 0E 04 FF 04 12 15
.:  2EC8 0B 20 14 0F 05 14 13 FF
.:  2ED0 0F 0B FF A9 04 85 B9 A9



.;  2ED3 A9 04      LDA #$04     DELAY
.;  2ED5 85 B9      STA $B9
.;  2ED7 A9 FF      LDA #$FF
.;  2ED9 85 BA      STA $BA
.;  2EDB A9 FF      LDA #$FF
.;  2EDD 85 BB      STA $BB
.;  2EDF C6 BB      DEC $BB
.;  2EE1 D0 FC      BNE $2EDF
.;  2EE3 C6 BA      DEC $BA
.;  2EE5 D0 F4      BNE $2EDB
.;  2EE7 C6 B9      DEC $B9
.;  2EE9 D0 EC      BNE $2ED7
.;  2EEB A2 00      LDX #$00
.;  2EED 60         RTS
```

Horst Brettin, 1000 Berlin 44

EVAL für CBM 3001

Manchmal ergibt sich die Notwendigkeit, daß man einen als String vorliegenden mathematischen Ausdruck berechnen muß. Einige BASIC-Dialekte haben hierfür die EVAL-Funktion, z.B. auch das EXBASIC Level II. - Hier nun wurde die EVAL-Funktion als USR-Funktion für den CBM 3001 erstellt. Dabei wurde gleichzeitig das Problem gelöst, wie man mit der USR-Funktion einen String verarbeitet. Ähnliche Lösungen sollten für alle Commodore-Rechner möglich sein, auch für den AIM 65, dessen BASIC ja recht ähnlich ist.

Programmbeschreibung: Der Aufruf der Routine FRESTR ist für jede Stringverarbeitung des Interpreters unbedingt notwendig, sonst kommt die interne String-Organisation durcheinander. Der auszuwertende String wird - wenn er nicht länger als der BASIC-Input-Buffer ist - in diesen transportiert, dort in ein Token gewandelt und dann vom Interpreter mit der Routine FORNUM ausgewertet. Das Ergebnis steht dann im BASIC-Accu1, wo es am Ende der USR-Funktion wieder an den Interpreter übergeben wird. Der BASIC-Programmzähler PC muß solange auf den Stapel gerettet werden. - Das Programm ist selbst voll verschieblich.

Das BASIC-Programm EVAL-DEMO dient zum Eingeben des Programmes ohne Assembler (per DATA) und zur Vorführung. Es hat darüber hinaus keinen besonderen Sinn. Zur Syntax z.B.:

```
X=USR(F$) ODER AUCH: PRINT USR("3"+F$)/USR(X$)
```

Der String darf jeden mathematischen Ausdruck darstellen,, der auch im BASIC erlaubt ist. Der String selbst darf auch wiederum durch einen Ausdruck dargestellt werden (2. Beispiel). Die USR-Funktion kann beliebig in weiteren Formeln eingebaut werden.

Einschränkungen: Die EVAL-Funktion darf nicht im Direktmodus verwendet werden, da ja der Input-Buffer gebraucht wird. Und: Die Funktion darf sich nicht selbst enthalten.

(Anmerkungen des Herausgebers: Der Autor weist darauf hin, daß es sich um eine eigene Entwicklung handelt, auch wenn sich problembedingt Ähnlichkeiten zum EXBASIC II zufällig ergeben sollten. - Weil BASIC viele Befehle zur Stringmanipulation enthält, ist es denkbar, daß der vorliegende Ansatz zu einem 'lernfähigen' Formelauswertungssystem führen könnte.)

EVAL-DEMO

```
 100 REM EVAL-DEMO       (C) BY H.BRETTIN          13.07.83
 110 REM        SYNTAX:  USR( STRING$ )
 120 Y=826:X%=Y/256
 130 IF PEEK(2)=X% GOTO 200
 140 POKE 1,Y AND 255:POKE 2,X%
 150 FOR Y=Y TO Y+57:READ X:POKE Y,X :NEXT
 200 X=99:Y=4*ATN(1)      :VARIABLEN FUER DEMO
 210 PRINT "X="X,"Y="Y
 220 INPUT"FUNKTION X=";F$
 230 PRINT F$ "=" USR( F$ )
 240 GOTO 220
 250 END
 900 DATA 32,128,210,32,125,213,168,240
 910 DATA 46,192,80,176,39,165,119,72
 920 DATA 165,120,72,169,0,162,2,133
 930 DATA 119,134,120,44,177,31,153,0
 940 DATA 2,136,16,248,32,149,196,32
 950 DATA 112,0,32,139,204,104,133,120
 960 DATA 104,133,119,96,76,50,213,76
 970 DATA 35,209
READY.
```

```
0001 033A                      ;EVAL  1.1                    13.07.83
0002 033A                      ;(C) BY HORST BRETTIN
0003 033A                      ;SYNTAX:  USR(STRING$)
0004 033A
0005 033A           STRP       =$1F      STRING-POINTER
0006 033A           CHRGET     =$70
0007 033A           PC         =$77      BASIC-PROGRAMM-COUNTER
0008 033A
0009 033A           INBUF      =$200
0010 033A           MAKTKN     =$C495
0011 033A           FORNUM     =$CC8B
0012 033A           ILLEGAL    =$D123
0013 033A           DIREKT     =$D280
0014 033A           TOLONGE    =$D532
0015 033A           FRESTR     =$D57D
0016 033A
0017 033A
0018 033A 20 80 D2 EVAL        JSR DIREKT
0019 033D 20 7D D5             JSR FRESTR          ;IF A STRING
0020 0340 A8                   TAY
0021 0341 F0 2E                BEQ NOSTR
0022 0343 C0 50                CPY#80              ;STRING > 80
0023 0345 B0 27                BCS TOLONG
0024 0347 A5 77                LDA PC              ;BASIC-PC RETTEN
0025 0349 48                   PHA
0026 034A A5 78                LDA PC+1
0027 034C 48                   PHA
0028 034D A9 00                LDA#<INBUF
0029 034F A2 02                LDX#>INBUF
0030 0351 85 77                STA PC
0031 0353 86 78                STX PC+1
0032 0355 2C                   .BYT $2C
0033 0356 B1 1F     LOOP       LDA(STRP),Y
0034 0358 99 00 02             STA INBUF,Y
0035 035B 88                   DEY
0036 035C 10 F8                BPL LOOP
0037 035E 20 95 C4             JSR MAKTKN
0038 0361 20 70 00             JSR CHRGET
0039 0364 20 8B CC             JSR FORNUM
0040 0367 68                   PLA
0041 0368 85 78                STA PC+1
0042 036A 68                   PLA
0043 036B 85 77                STA PC
0044 036D 60                   RTS
0045 036E
0046 036E 4C 32 D5 TOLONG      JMP TOLONGE         ;'STRING TOO LONG ERROR
0047 0371 4C 23 D1 NOSTR       JMP ILLEGAL         ;'ILLEGAL QUANTITY ERRO
SYMBOLTABELLE

  *=00884                                          ILLEGAL   =D123=53539
ZDL:00014                                          INBUF     =0200=00512
 UD:00000                                          LOOP      =0356=00854
 PA:00002                                          MAKTKN    =C495=50325
CHRGET    =0070=00112                              NOSTR     =0371=00881
DIREKT    =D280=53888                              PC        =0077=00119
EVAL      =033A=00826*                             STRP      =001F=00031
FORNUM    =CC8B=52363                              TOLONG    =036E=00878
FRESTR    =D57D=54653                              TOLONGE   =D532=54578
```

Dipl.-Ing. Rüdiger Wollenberg, 4600 Dortmund 50

ROFF (2)

Textformatierung für 6502-Mikroprozessoren

(Hinweis des Hrsg.: Der erste Teil des ROFF erschien in Heft 33 mit der Beschreibung und mit der Liste seiner Befehle. Aus Platzgründen wird hier zunächst die vollständige Programm-Liste abgedruckt, damit der Leser schon jetzt eine Implementierung vornehmen kann. Im nächsten Heft wird dann der noch fehlende Abschnitt 3 veröffentlicht, der Hintergründe zum Programmaufbau enthält.)

```
*************************************************************
FORMATIERER     V 3.2    3.09.1983   WOLLENBERG
0000
*************************************************************
DEFINITIONEN

0000
0000 MAXLIN         =61             ;MAXIMALE EDITORZEILENZAHL
0000 MAXOUT         =255            ;MAXIMALE AUSGABE-ZEICHENZAHL
0000 NEWLIN         =$0             ;SATZTRENNER CARRIAGE RETURN
0000 BLANK          =$20            ;LEERZEICHEN
0000 SEMIBL         =$5E            ;ERSATZVARIABLE FÜR BLANKS
0000 EOF            =0              ;TEXTENDE
0000 ESCAPE         =$1B            ;ESCAPESEQUENZEN
0000 PAGNUM         =$23            ;PLATZHALTER '#' FÜR SEITENZAHL
0000 PAGWID         =70             ;LÄNGE EINER ZEILE
0000 PAGLEN         =66             ;SEITENLÄNGE PRO SEITE
0000 M1VAL          =2              ;OBERER RAND EINSCHL. KOPFZEILE
0000 M2VAL          =2              ;RAND NACH KOPFZEILE
0000 M3VAL          =2              ;RAND NACH LETZTER TEXTZEILE
0000 M4VAL          =2              ;UNTERER RAND EINSCHL.FUSSZEILE
0000 HUGE           =127            ;GRÖSSE ZAHL
0000 COMIN          =$E1A1          ;EINSPRUNG INS MONITORPROGRAMM
0000 OUTFLG         =$A413          ;FLAG FÜR AUSGABEEINHEIT
0000 WHEREO         =$E871          ;VERTEILER FÜR DRUCKAUSGABE
0000 RCHEK          =$E907          ;UNTERBRECHUNG?
0000 OUTALL         =$E98C          ;AUSGABEROUTINE FÜR USER-DEVICE
0000 OUTPUT         =$E97A          ;AUSGABEROUTINE FÜR DISPLAY/PR.
0000 PHXY           =$EB9E          ;PUSH X AND Y ON STACK
0000 PLXY           =$EBAC          ;PULL X AND Y FROM STACK
0000
*************************************************************
ZERO-PAGE-BELEGUNG

0000                *=0
0000 D              *=*+1           ;DEZIMALWERT
0001 F              *=*+1           ;HILFSREGISTER
0002 F              *=*+1           ;HILFSREGISTER
0003 I              *=*+1           ;LAUFINDEX
0004 IY             *=*+1           ;LAUFINDEX
0005 JX             *=*+1           ;LAUFINDEX
0006 W              *=*+1           ;WORTLÄNGE IN WRDBUF
0007 OUTP           *=*+1           ;ZEICHENZAHL IN OUTBUF
0008 OUTW           *=*+1           ;DRUCKBARE ZEICHEN IN OUTBUF
0009 OUTWDS         *=*+1           ;WORTANZAHL IN OUTBUF
000A LAST           *=*+1           ;LETZTES ZEICHEN IN OUTBUF
```

```
000B NEXTRA         *=*+1           ;ZAHL DER AUFZUFÜLLENDEN BLANKS
000C NHOLES         *=*+1           ;ZAHL DER BLANK-LÖCHER
000D NB             *=*+1           ;HILFSVARIABLE IN SPREAD
000E NEE            *=*+1           ;HILFSVARIABLE IN SPREAD
000F DIR            *=*+1           ;BLANKAUFFÜLLUNG RECHTS/LINKS
0010 FILL           *=*+1           ;EINRÜCKUNG 0=JA/255=NEIN
0011 SKIFLG         *=*+1           ;ÜBERSPRINGE TEXT
0012 LSVAL          *=*+1           ;LAUFENDE ZEILE
0013 INVAL          *=*+1           ;LAUFENDE EINRÜCKUNG
0014 RMVAL          *=*+1           ;LAUFENDER RECHTER RAND
0015 TIVAL          *=*+1           ;LAUFENDE TEMPORÄRE EINRÜCKUNG
0016 LLVAL          *=*+1           ;LAUFENDE EINRÜCKUNG
0017 CEVAL          *=*+1           ;ANZAHL ZU ZENTRIERENDER ZEILEN
0018 ULVAL          *=*+1           ;ANZAHL ZU UNTERSTREICHENDER Z.
0019 FAVAL          *=*+1           ;ANZAHL FETT ZU SCHREIBENDER Z.
001A SPVAL          *=*+1           ;ANZAHL DER LEERZEILEN
001B CURPAG         *=*+1           ;LAUFENDE SEITENNUMMER
001C NEWPAG         *=*+1           ;NÄCHSTE SEITENNUMMER
001D LINENO         *=*+1           ;NÄCHSTE ZEILE
001E PLVAL          *=*+1           ;SEITENLÄNGE
001F BOTTOM         *=*+1           ;LETZTE TEXTZEILE
0020 HEADER         *=*+1           ;KOPFZEILE
0021                *=*+MAXLIN
005E FOOTER         *=*+1           ;FUSSZEILE
005F                *=*+MAXLIN
009C PARAM          *=*+1           ;KALKULIERTER COMMANDOPARAMETER
009D VAL            *=*+1           ;EINGEGEBENER COMMANDOPARAMETER
009E ARGTYP         *=*+1           ;VORZEICHEN COMMANDOPARAMETER
009F DEFVAL         *=*+1           ;VORDEFINIERTER COMMANDOPARAM.
00A0 MINVAL         *=*+1           ;MINIMALWERT COMMANDOPARAMETER
00A1 MAXVAL         *=*+1           ;MAXIMALWERT COMMANDOPARAMETER
00A2                *=$DF
00DF NOWLN          *=*+2           ;CURRENT EDITORLINE
00E1 BOTLN          *=*+2           ;LAST AKTIVE, SO FAR
00E3 START          *=*+2           ;STARTLIMIT OF BUFFER
00E5 END            *=*+2           ;ENDLIMIT OF BUFFER
00E7                *=$200
0200 INBUF          *=*+1           ;INPUTBUFFER FÜR EINE ZEILE
0201                *=*+MAXLIN
023E WRDBUF         *=*+1           ;ZWISCHENBUFFER FÜR WORTE
023F                *=*+MAXLIN
027C OUTBUF         *=*+1           ;AUSGANGSBUFFER FÜR EINE ZEILE
027D                *=*+MAXOUT
037C                *=$A47D
A47D JUMP           *=*+2           ;SPRUNGVEKTOR FÜR JSR(VEKTOR)
A47F
*************************************************************
HAUPTPROGRAMM

A47F                *=$C800
C800 FORMAT 2025C8  JSR INIT        ;INITIALISIERUNG
C803 FORMA1 2007E9  JSR RCHEK       ;TASTATURUNTERBRECHUNG?
C806        2002CB  JSR GETLIN      ;LIES EDITORZEILE
C809        C900    CMP #EOF        ;1. ZEICHEN = TEXTENDE?
C80B        F010    BEQ FORMA3
C80D        C92E    CMP #'.'        ;COMMANDO ODER TEXT ?
C80F        D006    BNE FORMA2
```

```
C811        206CC8 JSR COMAND      ;COMMANDO
C814        4C03C8 JMP FORMA1
C817 FORMA2 2072C8 JSR TEXT        ;TEXT
C81A        4C03C8 JMP FORMA1
C81D FORMA3 A97F   LDA #HUGE       ;VIELE BLANKS AUSGEBEN
C81F        204FCD JSR SPACE
C822        4CA1E1 JMP COMIN       ;EINSPRUNG IN MONITOR
C825
*************************************************************
INITIALISIERUNG

C825 INIT   A946   LDA #PAGWID     ;MAXIMALE EDITOR-ZEILEN-ZAHL
C827        8514   STA RMVAL       ;RECHTER RAND
C829
C829        A900   LDA #0
C82B        8510   STA FILL        ;FILL=YES
C82D        8511   STA SKIFLG      ;SKIP=NO
C82F        8513   STA INVAL       ;LFD. EINRÜCKUNG
C831        8515   STA TIVAL       ;TEM. EINRÜCKUNG
C833        8517   STA CEVAL       ;ZEENNTRIERUNG=NO
C835        8518   STA ULVAL       ;UNTERSTREICHUNG=NO
C837        8519   STA FAVAL       ;FETT SCHREIBEN=NO
C839        8518   STA CURPAG      ;ERSTE SEITE=0
C83B        851C   STA NEWPAG
C83D        851D   STA LINENO      ;ERSTE ZEILE=0
C83F        8507   STA OUTP
C841        8508   STA OUTW
C843        8509   STA OUTWDS
C845        850F   STA DIR
C847
C847        A901   LDA #1
C849        8512   STA LSVAL       ;ZEILENABSTAND=1
C84B        851A   STA SPVAL       ;ZEILENSPRUNG=1
C84D
C84D        A90D   LDA #NEWLIN     ;KOPFZEILE=LEER
C84F        A001   LDY #1          ;FUSSZEILE=LEER
C851        992000 STA HEADER,Y
C854        995E00 STA FOOTER,Y
C857        A942   LDA #PAGLEN     ;TEXTENDE
C859        851E   STA PLVAL
C85B        38     SEC
C85C        E904   SBC #M3VAL+M4VAL
C85E        851F   STA BOTTOM
C860
C860        A5E3   LDA START       ;BEGINNE VORNE IM TEXTSPEICHER
C862        85DF   STA NOWLN
C864        A5E4   LDA START+1
C866        85E0   STA NOWLN+1
C868        2071E8 JSR WHEREO      ;WELCHES OUTPUT-DEVICE
C86B
C86B        60     RTS
C86C
*************************************************************
COMMANDO-INTERPRETATION

C86C COMAND 20F5C8 JSR GETVAL      ;HOLE COMMANDOPARAMETER
C86F        A200   LDX #0
C871        A002   LDY #2
```

```
C873 COMAN1 B90002 LDA INBUF,Y     ;1. COMMANDOZEICHEN
C876        20CDCF JSR FILT1       ;KLEIN/GROSS
C879        DDA8C8 CMP COMTAB,X    ;IN TABELLE?
C87C        F008   BEQ COMAN2      ;JA
C87E        E8     INX             ;NEIN
C87F        E8     INX
C880
C880        BDA8C8 LDA COMTAB,X    ;COMMANDOTABELLE ZU ENDE?
C883        D0EE   BNE COMAN1      ;NEIN
C885        60     RTS             ;JA, ÜBERSPRINGE COMMANDO
C886
C886 COMAN2 B90102 LDA INBUF+1,Y   ;2. COMMANDOZEICHEN
C889        20CDCF JSR FILT1       ;KLEIN/GROSS
C88C        DDA9C8 CMP COMTAB+1,X  ;IN TABELLE?
C88F        F008   BEQ COMAN3      ;JA, ZEICHEN GEFUNDEN
C891        E8     INX             ;NEIN
C892        E8     INX
C893        BDA8C8 LDA COMTAB,X    ;COMMANDOTABELLE ZU ENDE?
C896        D0DB   BNE COMAN1
C898        60     RTS
C899
C899 COMAN3 BDCFC8 LDA COMJMP,X    ;HOLE EINSPRUNGMARKE
C89C        8D7DA4 STA JUMP
C89F        BDD0C8 LDA COMJMP+1,X
C8A2        8D7EA4 STA JUMP+1
C8A5
C8A5        6C7DA4 JMP (JUMP)      ;FÜHRE COMMANDO AUS
C8A8
C8A8 COMTAB 4649   .BYT 'FI','NF','BR','LS','SP','UL','HE','FO'
C8AA        4E46
C8AC        4252
C8AE        4C53
C8B0        5350
C8B2        554C
C8B4        4845
C8B6        464F
C8B8        504C   .BYT 'PL','IN','RM','TI','BP','FA','NE','PP'
C8BA        494E
C8BC        524D
C8BE        5449
C8C0        4250
C8C2        4641
C8C4        4E45
C8C6        5050
C8C8        4345   .BYT 'CE','BS','ES'
C8CA        4253
C8CC        4553
C8CE        00     .BYT 0
C8CF
C8CF COMJMP 8EC9   .WOR FI,NF,BR,LS,SP,UL,HE,FO
C8D1        96C9
C8D3        9EC9
C8D5        A1C9
C8D7        B7C9
C8D9        D1C9
C8DB        EBC9
C8DD        1CCA
C8DF        4DCA   .WOR PL,IN,RM,TI,BP,FA,NE,PP
```

```
C8E1        68CA
C8E3        81CA
C8E5        9ACA
C8E7        B3CA
C8E9        D7CA
C8EB        F2CA
C8ED        25CB
C8EF        4DCB    .WOR CE,BS,ES
C8F1        68CB
C8F3        6DCB
C8F5
*************************************************************
HOLE COMMANDOPARAMETER

C8F5 GETVAL A000    LDY #0
C8F7 GETVA1 C8      INY
C8F8        B90002 LDA INBUF,Y      ;LIES ZEICHEN
C8FB        C920    CMP #BLANK      ;ÜBERSPRINGE COMMANDO
C8FD        F007    BEQ GETVA2
C8FF        C90D    CMP #NEWLIN     ;ZEILENENDE?
C901        D0F4    BNE GETVA1
C903        859E    STA ARGTYP
C905        60      RTS
C906
C906 GETVA2 C8      INY
C907        B90002 LDA INBUF,Y      ;ÜBERSPRINGE BLANKS
C90A        C920    CMP #BLANK
C90C        F0F8    BEQ GETVA2
C90E
C90E        859E    STA ARGTYP      ;+/- ?
C910        C92B    CMP #'+'
C912        F004    BEQ GETVA3
C914        C92D    CMP #'-'
C916        D001    BNE GETVA4
C918 GETVA3 C8      INY             ;JA, ERHÖHE ZEICHEN
C919 GETVA4 B90002 LDA INBUF,Y
C91C        C920    CMP #BLANK      ;ÜBERSPRINGE BLANKS
C91E        F0F8    BEQ GETVA3
C920
C920 GETVA5 A900    LDA #0          ;WERT = 0
C922        859D    STA VAL
C924
C924 GETVA6 B90002 LDA INBUF,Y      ;HOLE ZEICHEN
C927        C90D    CMP #NEWLIN     ;ZEILE ZU ENDE?
C929        F00C    BEQ GETVA8
C92B
C92B        A200    LDX #0
C92D GETVA7 DD4CC9 CMP DIGITS,X     ;GÜLTIGE ZAHL?
C930        F006    BEQ GETVA9
C932        E8      INX
C933        E00A    CPX #10         ;ALLE ZIFFERN DURCH?
C935        D0F6    BNE GETVA7      ;NEIN
C937 GETVA8 60      RTS             ;JA, ALSO ABBRUCH
C938
C938 GETVA9 8600    STX D           ;ZAHLENWERT RETTEN
C93A        A209    LDX #9          ;ALTE ZAHL * 10
C93C        18      CLC
C93D        A59D    LDA VAL
```

```
C93F GETVAA 659D   ADC VAL
C941        CA     DEX
C942        D0FB   BNE GETVAA
C944        6500   ADC D
C946        859D   STA VAL
C948
C948        C8     INY             ;NÄCHSTE ZIFFER
C949        4C24C9 JMP GETVA6
C94C
C94C DIGITS 3031   .BYT '0123456789'
C956
**********************************************************
PARAMETERSETZUNG

C956 SET    A59E   LDA ARGTYP
C958
C958        C90D   CMP #NEWLIN     ;KEINE PARAMETER
C95A        D005   BNE SET1
C95C        A59F   LDA DEFVAL      ;ALSO VOREINGESTELLTER WERT
C95E        4C7BC9 JMP SET4
C961
C961 SET1   C92B   CMP #'+'        ;POSITIVER OFFSET
C963        D008   BNE SET2
C965        18     CLC
C966        A59C   LDA PARAM
C968        659D   ADC VAL
C96A        4C7BC9 JMP SET4
C96D
C96D SET2   C92D   CMP #'-'        ;NEGATIVER OFFSET
C96F        D008   BNE SET3
C971        38     SEC
C972        A59C   LDA PARAM
C974        E59D   SBC VAL
C976        4C7BC9 JMP SET4
C979
C979 SET3   A59D   LDA VAL         ;ABSOLUTWERT
C97B
C97B SET4   859C   STA PARAM       ;MAXIMALWERT ÜBERSCHRITTEN?
C97D        C5A1   CMP MAXVAL
C97F        9004   BCC SET5        ;NEIN
C981        A5A1   LDA MAXVAL      ;JA, ALSO AUSTAUSCH
C983        859C   STA PARAM
C985
C985 SET5   C5A0   CMP MINVAL      ;MINIMALWERT UNTERSCHRITTEN?
C987        B004   BCS SET6        ;NEIN
C989        A5A0   LDA MINVAL      ;JA, ALSO AUSTAUSCH
C98B        859C   STA PARAM
C98D SET6   60     RTS
C98E
**********************************************************
FILL-COMMANDO

C98E FI     2074CE JSR BREAK       ;ZEILENABBRUCH
C991        A900   LDA #0          ;FILL SETZEN MIT 0
C993        8510   STA FILL
C995        60     RTS
C996
**********************************************************
NOFILL-COMMANDO
```

```
C996 NF     2074CE JSR BREAK       ;ZEILENABBRUCH
C999        A9FF   LDA #255        ;FILL LÖSCHEN MIT 255
C99B        8510   STA FILL
C99D        60     RTS
C99E
*************************************************************
BREAK-COMMANDO

C99E BR     4C74CE JMP BREAK       ;ERZEUGE ZEILENABBRUCH
C9A1
*************************************************************
LINE-SPACE-COMMANDO

C9A1 LS     A512   LDA LSVAL
C9A3        859C   STA PARAM
C9A5        A901   LDA #1
C9A7        859F   STA DEFVAL
C9A9        85A0   STA MINVAL
C9AB        A97F   LDA #HUGE
C9AD        85A1   STA MAXVAL
C9AF        2056C9 JSR SET
C9B2        A59C   LDA PARAM
C9B4        8512   STA LSVAL
C9B6        60     RTS
C9B7
*************************************************************
SPACE-COMMANDO

C9B7 SP     A51A   LDA SPVAL
C9B9        859C   STA PARAM
C9BB        A901   LDA #1
C9BD        859F   STA DEFVAL
C9BF        A900   LDA #0
C9C1        85A0   STA MINVAL
C9C3        A97F   LDA #HUGE
C9C5        85A1   STA MAXVAL
C9C7        2056C9 JSR SET
C9CA        A59C   LDA PARAM
C9CC        851A   STA SPVAL
C9CE        4C4FCD JMP SPACE
C9D1
*************************************************************
UNDERLINE-COMMANDO

C9D1 UL     A518   LDA ULVAL
C9D3        859C   STA PARAM
C9D5        A901   LDA #1
C9D7        859F   STA DEFVAL
C9D9        A900   LDA #0
C9DB        85A0   STA MINVAL
C9DD        A97F   LDA #HUGE
C9DF        85A1   STA MAXVAL
C9E1        2056C9 JSR SET
C9E4        A59C   LDA PARAM
C9E6        8518   STA ULVAL
C9E8        4C4FCF JMP UNDON
C9EB
```

```
**************************************************************
KOPFZEILE EINLESEN

C9EB HE     A200    LDX #0
C9ED        A000    LDY #0
C9EF HE1    C8      INY             ;COMMANDO ÜBERSPRINGEN
C9F0        B90002  LDA INBUF,Y
C9F3        C920    CMP #BLANK
C9F5        F004    BEQ HE2
C9F7        C90D    CMP #NEWLIN
C9F9        D0F4    BNE HE1
C9FB HE2    C8      INY             ;BLANKS ÜBERSPRINGEN
C9FC        B90002  LDA INBUF,Y
C9FF        C920    CMP #BLANK
CA01        F0F8    BEQ HE2
CA03        C927    CMP #'''            ;IGNORIERE ''' UND '"'
CA05        F004    BEQ HE3
CA07        C922    CMP #'"'
CA09        D001    BNE HE4
CA0B HE3    C8      INY             ;UMSPEICHERN
CA0C HE4    E8      INX
CA0D        B90002  LDA INBUF,Y
CA10        C90D    CMP #NEWLIN
CA12        F005    BEQ HE5
CA14        9520    STA HEADER,X
CA16        4C0BCA  JMP HE3
CA19 HE5    9520    STA HEADER,X
CA1B        60      RTS
CA1C
**************************************************************
FUSSZEILE LESEN

CA1C F0     A200    LDX #0
CA1E        A000    LDY #0
CA20 F01    C8      INY             ;COMMANDO ÜBERSPRINGEN
CA21        B90002  LDA INBUF,Y
CA24        C920    CMP #BLANK
CA26        F004    BEQ F02
CA28        C90D    CMP #NEWLIN
CA2A        D0F4    BNE F01
CA2C F02    C8      INY             ;BLANKS ÜBERSPRINGEN
CA2D        B90002  LDA INBUF,Y
CA30        C920    CMP #BLANK
CA32        F0F8    BEQ F02
CA34        C927    CMP #'''            ;IGNORIERE ''' UND '"'
CA36        F004    BEQ F03
CA38        C922    CMP #'"'
CA3A        D001    BNE F04
CA3C F03    C8      INY             ;UMSPEICHERN
CA3D F04    E8      INX
CA3E        B90002  LDA INBUF,Y
CA41        C90D    CMP #NEWLIN
CA43        F005    BEQ F05
CA45        955E    STA FOOTER,X
CA47        4C3CCA  JMP F03
CA4A F05    955E    STA FOOTER,X
CA4C        60      RTS
CA4D
```

```
**************************************************************
SEITENLÄNGE ÄNDERN
CA4D PL     A51E   LDA PLVAL
CA4F        859C   STA PARAM
CA51        A942   LDA #PAGLEN
CA53        859F   STA DEFVAL
CA55        A909   LDA #M1VAL+M2VAL+M3VAL+M4VAL+1
CA57        85A0   STA MINVAL
CA59        A97F   LDA #HUGE
CA5B        85A1   STA MAXVAL
CA5D        2056C9 JSR SET
CA60        851E   STA PLVAL
CA62        38     SEC              ;BOTTOM KORREGIEREN
CA63        E904   SBC #M3VAL+M4VAL
CA65        851F   STA BOTTOM
CA67        60     RTS
CA68
**************************************************************
SETZE EINRÜCKUNGEN

CA68 IN     A513   LDA INVAL
CA6A        859C   STA PARAM
CA6C        A900   LDA #0
CA6E        859F   STA DEFVAL
CA70        85A0   STA MINVAL
CA72        A614   LDX RMVAL
CA74        CA     DEX
CA75        86A1   STX MAXVAL
CA77        2056C9 JSR SET
CA7A        A59C   LDA PARAM
CA7C        8513   STA INVAL
CA7E        8515   STA TIVAL
CA80        60     RTS
CA81
**************************************************************
SETZE RECHTEN RAND

CA81 RM     A514   LDA RMVAL
CA83        859C   STA PARAM
CA85        A946   LDA #PAGWID
CA87        859F   STA DEFVAL
CA89        A615   LDX TIVAL
CA8B        E8     INX
CA8C        86A0   STX MINVAL
CA8E        A97F   LDA #HUGE
CA90        85A1   STA MAXVAL
CA92        2056C9 JSR SET
CA95        A59C   LDA PARAM
CA97        8514   STA RMVAL
CA99        60     RTS
CA9A
**************************************************************
SETZE TEMPORÄRE EINRÜCKUNG

CA9A TI     2074CE JSR BREAK
CA9D        A515   LDA TIVAL
CA9F        859C   STA PARAM
```

```
CAA1        A900    LDA #0
CAA3        859F    STA DEFVAL
CAA5        85A0    STA MINVAL
CAA7        A514    LDA RMVAL
CAA9        85A1    STA MAXVAL
CAAB        2056C9  JSR SET
CAAE        A59C    LDA PARAM
CAB0        8515    STA TIVAL
CAB2        60      RTS
CAB3
*************************************************************

BEGIN PAGE-COMMANDO

CAB3 BP     A51D    LDA LINENO
CAB5        F005    BEQ BP1
CAB7        A97F    LDA #HUGE
CAB9        204FCD  JSR SPACE
CABC BP1    A518    LDA CURPAG
CABE        859C    STA PARAM
CAC0        AA      TAX
CAC1        E8      INX
CAC2        8A      TXA
CAC3        859F    STA DEFVAL
CAC5        A500    LDA 0
CAC7        85A0    STA MINVAL
CAC9        A97F    LDA #HUGE
CACB        85A1    STA MAXVAL
CACD        2056C9  JSR SET
CAD0        A59C    LDA PARAM
CAD2        851B    STA CURPAG      ;AKTUALISIERE SEITENZAHL
CAD4        851C    STA NEWPAG
CAD6        60      RTS
CAD7
*************************************************************

FETTSCHREIB-COMMANDO

CAD7 FA     A519    LDA FAVAL
CAD9        859C    STA PARAM
CADB        A901    LDA #1
CADD        859F    STA DEFVAL
CADF        A900    LDA #0
CAE1        85A0    STA MINVAL
CAE3        A97F    LDA #HUGE
CAE5        85A1    STA MAXVAL
CAE7        2056C9  JSR SET
CAEA        A59C    LDA PARAM
CAEC        8519    STA FAVAL
CAEE        208ECF  JSR FATON
CAF1        60      RTS
CAF2
*************************************************************

NEED N LINES OF TEXT

CAF2 NE     2074CE  JSR BREAK
CAF5        A900    LDA #0
CAF7        859C    STA PARAM
CAF9        A904    LDA #4
CAFB        859F    STA DEFVAL
CAFD        A900    LDA #0
```

```
CAFF        85A0    STA MINVAL
CB01        A97F    LDA #HUGE
CB03        85A1    STA MAXVAL
CB05        2056C9  JSR SET
CB08        A51D    LDA LINENO      ;IM BEREICH DER KOPFZEILE?
CB0A        C904    CMP #M1VAL+M2VAL
CB0C        B002    BCS NE1
CB0E        A904    LDA #M1VAL+M2VAL
CB10 NE1    8500    STA D
CB12        A51F    LDA BOTTOM      ;IST NOCH PLATZ?
CB14        38      SEC
CB15        E500    SBC D
CB17        F007    BEQ NE2
CB19        9005    BCC NE2
CB1B        C59C    CMP PARAM
CB1D        9001    BCC NE2
CB1F        60      RTS
CB20 NE2    A59C    LDA PARAM       ;NEIN, ALSO UMBLÄTTERN
CB22        4C4FCD  JMP SPACE
CB25
*************************************************************
PARAGRAPH, NEUER ABSATZ

CB25 PP     2074CE  JSR BREAK
CB28        A903    LDA #3          ;SIND NOCH DREI ZEILEN FREI?
CB2A        859C    STA PARAM
CB2C        A51D    LDA LINENO
CB2E        C904    CMP #M1VAL+M2VAL
CB30        B002    BCS PP1
CB32        A904    LDA #M1VAL+M2VAL
CB34 PP1    8500    STA D
CB36        A51F    LDA BOTTOM
CB38        38      SEC
CB39        E500    SBC D
CB3B        F008    BEQ PP2
CB3D        9009    BCC PP2
CB3F        C59C    CMP PARAM
CB41        9005    BCC PP2
CB43        A901    LDA #1          ;JA, ALSO EINE FREIZEILE
CB45        4C4FCD  JMP SPACE
CB48 PP2    A59C    LDA PARAM
CB4A        4C4FCD  JMP SPACE       ;NEIN, ALSO UMBLÄTTERN
CB4D
*************************************************************
CENTRIERE ZEILEN

CB4D CE     2074CE  JSR BREAK
CB50        A517    LDA CEVAL
CB52        859C    STA PARAM
CB54        A901    LDA #1
CB56        859F    STA DEFVAL
CB58        A900    LDA #0
CB5A        85A0    STA MINVAL
CB5C        A97F    LDA #HUGE
CB5E        85A1    STA MAXVAL
CB60        2056C9  JSR SET
CB63        A59C    LDA PARAM
CB65        8517    STA CEVAL
CB67        60      RTS
```

```
*************************************************************
BEGIN SKIP

CB68 BS     A9FF   LDA #$FF
CB6A        8511   STA SKIFLG
CB6C        60     RTS
CB6D
*************************************************************
END SKIP

CB6D ES     A900   LDA #0
CB6F        8511   STA SKIFLG
CB71        60     RTS
CB72
*************************************************************
DRUCKE TEXTZEILEN

CB72 TEXT   A001   LDY #1          ;FÜHRENDE BLANKS OD. FREIZEILE?
CB74        B90002 LDA INBUF,Y
CB77        C920   CMP #BLANK
CB79        F004   BEQ TEXT1
CB7B        C90D   CMP #NEWLIN
CB7D        D003   BNE TEXT3
CB7F TEXT1  2089CE JSR LEADBL      ;JA
CB82 TEXT3  A517   LDA CEVAL       ;ZENTRIERUNG?
CB84        F008   BEQ TEXT4
CB86        202FCF JSR CENTER
CB89        2024CC JSR PUT
CB8C        C617   DEC CEVAL
CB8E        4CBBCB JMP TEXT8
CB91
CB91 TEXT4  A001   LDY #1          ;FREIZEILE
CB93        B90002 LDA INBUF,Y
CB96        C90D   CMP #NEWLIN
CB98        D006   BNE TEXT5
CB9A        2024CC JSR PUT
CB9D        4CB8CB JMP TEXT8
CBA0
CBA0 TEXT5  A510   LDA FILL
CBA2        F006   BEQ TEXT6
CBA4        2024CC JSR PUT         ;FILL = NO
CBA7        4CBBCB JMP TEXT8
CBAA
CBAA TEXT6  A001   LDY #1          ;FILL = YES
CBAC        8403   STY I
CBAE TEXT7  20F9CB JSR GETWRD
CBB1        C900   CMP #0
CBB3        F006   BEQ TEXT8
CBB5        20D2CC JSR PUTWRD
CBB8        4CAECB JMP TEXT7
CBBB
CBBB TEXT8  A518   LDA ULVAL       ;UNTERSTREICHUNG?
CBBD        F007   BEQ TEXT9
CBBF        C618   DEC ULVAL
CBC1        D003   BNE TEXT9
CBC3        206CCF JSR UNDOFF
CBC6
CBC6 TEXT9  A519   LDA FAVAL       ;FETTSCHRIFT
```

```
CBC8          F007    BEQ TEXTE
CBCA          C619    DEC FAVAL
CBCC          D003    BNE TEXTE
CBCE          20ABCF  JSR FATOFF
CBD1 TEXTE    60      RTS
CBD2
*************************************************************
LIEST EINE ZEILE AUS DEM EDITORSPEICHER

CBD2 GETLIN A200      LDX #0
CBD4        A000      LDY #0
CBD6 GETLI1 A1DF      LDA (NOWLN,X)   ;HOLE ZEILE AUS TEXTSPEICHER
CBD8        E6DF      INC NOWLN
CBDA        D002      BNE GETLI2
CBDC        E6E0      INC NOWLN+1
CBDE GETLI2 C900      CMP #EOF
CBE0        F016      BEQ GETLI4      ;TEXTBUFFERER LEER?
CBE2        C8        INY
CBE3        990002    STA INBUF,Y     ;NEIN, ALSO UMSPEICHERN
CBE6        C90D      CMP #NEWLIN
CBE8        F009      BEQ GETLI3
CBEA        C03D      CPY #MAXLIN
CBEC        D0E8      BNE GETLI1      ;NEIN, NÄCHSTES ZEICHEN
CBEE        A90D      LDA #NEWLIN
CBF0        990002    STA INBUF,Y
CBF3 GETLI3 A001      LDY #1          ;BEGINN EINER NEUEN ZEILE
CBF5        B90002    LDA INBUF,Y
CBF8 GETLI4 60        RTS             ;TEXTENDE
CBF9
*************************************************************
WORTÜBERTRAGUNG VON INBUF NACH WRDBUF, LÄNGE IM ACCU

CBF9 GETWRD A403      LDY I
CBFB        88        DEY
CBFC GETWR1 C8        INY             ;IGNORIERE BLANKS
CBFD        B90002    LDA INBUF,Y
CC00        C920      CMP #BLANK
CC02        F0F8      BEQ GETWR1
CC04        A200      LDX #0
CC06        88        DEY             ;UMSPEICHERN, WENN WEDER BLANKS
CC07 GETWR2 E8        INX             ;NOCH NEWLIN
CC08        C8        INY
CC09        B90002    LDA INBUF,Y
CC0C        C90D      CMP #NEWLIN
CC0E        F00A      BEQ GETWR3
CC10        C920      CMP #BLANK
CC12        F006      BEQ GETWR3
CC14        9D3E02    STA WRDBUF,X
CC17        4C07CC    JMP GETWR2
CC1A GETWR3 A90D      LDA #NEWLIN     ;SCHLIESSE ZEILE MIT NEWLIN
CC1C        9D3E02    STA WRDBUF,X
CC1F        CA        DEX
CC20        8A        TXA             ;WORTLÄNGE IM ACCU
CC21        8403      STY I           ;WORTENDE IN I
CC23        60        RTS
CC24
*************************************************************
INBUF MIT ZWISCHENRÄUMEN UND EINRÜCKUNGEN AUSGEBEN
```

```
CC24 PUT    A51D   LDA LINENO      ;EV. KOPFZEILE AUSGEBEN
CC26        F006   BEQ PUT1
CC28        C51F   CMP BOTTOM
CC2A        F005   BEQ PUT2
CC2C        9003   BCC PUT2
CC2E PUT1   2091CD JSR PHEAD       ;KOPFZEILE
CC31 PUT2   A000   LDY #0          ;EV. EINRÜCKUNGEN
CC33        A920   LDA #BLANK
CC35 PUT3   C8     INY
CC36        C415   CPY TIVAL
CC38        F002   BEQ PUT4
CC3A        1006   BPL PUT5
CC3C PUT4   2060CE JSR PUTC        ;BLANK AUSGEBEN
CC3F        4C35CC JMP PUT3
CC42 PUT5   A513   LDA INVAL
CC44        8515   STA TIVAL
CC46        206DCC JSR PUTLIN      ;TEXTZEILE AUSGEBEN
CC49        A612   LDX LSVAL       ;LEERZEILEN AUSGEBEN
CC4B        CA     DEX
CC4C        8600   STX D
CC4E        A51F   LDA BOTTOM
CC50        38     SEC
CC51        E51D   SBC LINENO
CC53        C500   CMP D
CC55        3002   BMI PUT6
CC57        A500   LDA D
CC59 PUT6   2085CD JSR SKIP
CC5C        A51D   LDA LINENO      ;ZEILENNUMMER AKTUALISIEREN
CC5E        18     CLC
CC5F        6512   ADC LSVAL
CC61        851D   STA LINENO
CC63        C51F   CMP BOTTOM      ;EV. FUSSZEILE AUSGEBEN
CC65        F005   BEQ PUT7
CC67        3003   BMI PUT7
CC69        20DFCD JSR PFOOT
CC6C PUT7   60     RTS
CC6D
*************************************************************
GIBT INBUF AUS

CC6D PUTLIN A000   LDY #0
CC6F PUTLI1 C8     INY
CC70        B90002 LDA INBUF,Y
CC73        2060CE JSR PUTC
CC76        C90D   CMP #NEWLIN
CC78        D0F5   BNE PUTLI1
CC7A        60     RTS
CC7B
*************************************************************
OUTBUF MIT ZWISCHENRÄUMEN UND EINRÜCKUNGEN AUSGEBEN

CC7B TUP    A51D   LDA LINENO      ;EV. KOPFZEILE AUSGEBEN
CC7D        F006   BEQ TUP1
CC7F        C51F   CMP BOTTOM
CC81        F005   BEQ TUP2
CC83        9003   BCC TUP2
CC85 TUP1   2091CD JSR PHEAD       ;KOPFZEILE
CC88 TUP2   A000   LDY #0          ;EV. EINRÜCKUNGEN
```

```
CC8A        A920    LDA #BLANK
CC8C TUP3   C8      INY
CC8D        C415    CPY TIVAL
CC8F        F002    BEQ TUP4
CC91        1006    BPL TUP5
CC93 TUP4   2060CE  JSR PUTC        ;BLANK AUSGEBEN
CC96        4C8CCC  JMP TUP3
CC99 TUP5   A513    LDA INVAL
CC98        8515    STA TIVAL
CC9D        20C4CC  JSR TUPLIN      ;TEXTZEILE AUSGEBEN
CCA0        A612    LDX LSVAL       ;LEERZEILEN AUSGEBEN
CCA2        CA      DEX
CCA3        8600    STX D
CCA5        A51F    LDA BOTTOM
CCA7        38      SEC
CCA8        E51D    SBC LINENO
CCAA        C500    CMP D
CCAC        3002    BMI TUP6
CCAE        A500    LDA D
CCB0 TUP6   2085CD  JSR SKIP
CCB3        A51D    LDA LINENO      ;ZEILENNUMMER AKTUALISIEREN
CCB5        18      CLC
CCB6        6512    ADC LSVAL
CCB8        851D    STA LINENO
CCBA        C51F    CMP BOTTOM      ;EV. FUSSZEILE AUSGEBEN
CCBC        F005    BEQ TUP7
CCBE        3003    BMI TUP7
CCC0        20DFCD  JSR PFOOT
CCC3 TUP7   60      RTS
CCC4
************************************************************
GIBT OUTBUF AUS

CCC4 TUPLIN A000    LDY #0
CCC6 TUPLI1 C8      INY
CCC7        B97C02  LDA OUTBUF,Y
CCCA        2060CE  JSR PUTC
CCCD        C90D    CMP #NEWLIN
CCCF        D0F5    BNE TUPLI1
CCD1        60      RTS
CCD2
************************************************************
BRINGT WRDBUF NACH OUTBUF

CCD2 PUTWRD A000    LDY #0
CCD4 PUTWR1 C8      INY
CCD5        B93E02  LDA WRDBUF,Y
CCD8        C90D    CMP #NEWLIN     ;LÄNGE VON WRDBUF FESTSTELLEN
CCDA        D0F8    BNE PUTWR1
CCDC        8406    STY W
CCDE        C606    DEC W
CCE0        98      TYA             ;LAST=W+OUTP+1
CCE1        18      CLC
CCE2        6507    ADC OUTP
CCE4        850A    STA LAST
CCE6 PUTWR2 A514    LDA RMVAL       ;LLVAL=RMVAL-TIVAL
CCE8        38      SEC
CCE9        E515    SBC TIVAL
```

```
CCEB        8516   STA LLVAL
CCED        A508   LDA OUTW        ;PRÜFE, OB ZEILENAUSGABE
CCEF        18     CLC
CCF0        6506   ADC W
CCF2        C516   CMP LLVAL
CCF4        F002   BEQ PUTWR3
CCF6        B006   BCS PUTWR4
CCF8 PUTWR3 A50A   LDA LAST
CCFA        C9FB   CMP #MAXOUT-4
CCFC        9027   BCC PUTWR7
CCFE PUTWR4 A50A   LDA LAST        ;JA, LAST=LAST-OUTP
CD00        38     SEC
CD01        E507   SBC OUTP
CD03        850A   STA LAST
CD05
CD05        A616   LDX LLVAL       ;NEXTRA=LLVAL-OUTW+1
CD07        E8     INX
CD08        8A     TXA
CD09        38     SEC
CD0A        E508   SBC OUTW
CD0C        850B   STA NEXTRA
CD0E        A509   LDA OUTWDS      ;KEIN AUFÜLLEN BEI EINEM WORT
CD10        C902   CMP #2
CD12        900E   BCC PUTWR6
CD14        A50B   LDA NEXTRA      ;KEIN AUFÜLLEN NÖTIG
CD16        F00A   BEQ PUTWR6
CD18
CD18        20B8CE JSR SPREAD      ;RICHTE RAND AUS
CD1B        A507   LDA OUTP
CD1D        18     CLC
CD1E        650B   ADC NEXTRA
CD20        8507   STA OUTP        ;AKTUALISIERE ZEICHENZAHL
CD22
CD22 PUTWR6 2074CE JSR BREAK       ;ERZWINGE AUSGABE VON OUTBUF
CD25 PUTWR7 A200   LDX #0
CD27        A407   LDY OUTP
CD29 PUTWR8 E8     INX
CD2A        C8     INY             ;KOPIERE WRDBUF NACH OUTBUF
CD2B        BD3E02 LDA WRDBUF,X
CD2E        C90D   CMP #NEWLIN
CD30        F006   BEQ PUTWR9
CD32        997C02 STA OUTBUF,Y
CD35        4C29CD JMP PUTWR8
CD38 PUTWR9 997C02 STA OUTBUF,Y    ;NEWLIN EINSPEICHERN
CD3B        A50A   LDA LAST        ;ZÄHLVARIABLEN AKTUALISIEREN
CD3D        8507   STA OUTP
CD3F        A8     TAY
CD40        A920   LDA #BLANK
CD42        997C02 STA OUTBUF,Y
CD45        A508   LDA OUTW
CD47        38     SEC
CD48        6506   ADC W
CD4A        8508   STA OUTW
CD4C        E609   INC OUTWDS
CD4E        60     RTS
```

```
*************************************************************
LEERZEILEN AUSGEBEN ENTSPRECHEND ACCU-INHALT

CD4F SPACE  8502   STA F           ;WERT SICHERN
CD51        2074CE JSR BREAK       ;AUSGABE ERZWINGEN
CD54        A51D   LDA LINENO      ;NOCH IM TEXTBEREICH?
CD56        C51F   CMP BOTTOM
CD58        9003   BCC SPACE1
CD5A        F001   BEQ SPACE1
CD5C        60     RTS             ;NEIN, RETURN
CD5D SPACE1 A51D   LDA LINENO      ;JA
CD5F        D003   BNE SPACE2
CD61        2091CD JSR PHEAD       ;DRUCKE KOPFZEILE
CD64 SPACE2 A61F   LDX BOTTOM
CD66        E8     INX
CD67        8A     TXA
CD68        38     SEC
CD69        E51D   SBC LINENO
CD6B        C502   CMP F
CD6D        9002   BCC SPACE3
CD6F        A502   LDA F
CD71 SPACE3 2085CD JSR SKIP        ;DRUCKE LEERZEILEN
CD74        A51D   LDA LINENO
CD76        18     CLC
CD77        6502   ADC F
CD79        851D   STA LINENO
CD7B        C51F   CMP BOTTOM
CD7D        9005   BCC SPACE4
CD7F        F003   BEQ SPACE4
CD81        20DFCD JSR PFOOT       ;FUSSZEILE AUSDRUCKEN
CD84 SPACE4 60     RTS
CD85
*************************************************************
ERZEUGT LEERE ZEILEN ENTSPRECHEND ACCU-INHALT

CD85 SKIP   AA     TAX
CD86        F008   BEQ SKIP2
CD88 SKIP1  A90D   LDA #NEWLIN
CD8A        2060CE JSR PUTC
CD8D        CA     DEX
CD8E        D0F8   BNE SKIP1
CD90 SKIP2  60     RTS
CD91
*************************************************************
SEITENKOPF AUSGEBEN UND SEITENNR. ERHÖHEN

CD91 PHEAD  2084CF JSR UNDOFD      ;UNTESTREICHEN AUSSCHALTEN
CD94        20C3CF JSR FATOFD      ;FETTSCHRIFT AUSSCHALTEN
CD97        A51C   LDA NEWPAG      ;SEITENZAHL ERHÖHEN
CD99        851B   STA CURPAG
CD9B        E61C   INC NEWPAG
CD9D        A902   LDA #M1VAL      ;KOPZEILE ERWÜNSCHT?
CD9F        F026   BEQ PHEAD4
CDA1        A202   LDX #M1VAL      ;JA, ALSO SKIP
CDA3        CA     DEX
CDA4        8A     TXA
CDA5        2085CD JSR SKIP
CDA8        A000   LDY #0
```

```
CDAA PHEAD1 C8      INY             ;DRUCKE KOPFZEILE
CDAB        B92000  LDA HEADER,Y
CDAE        C90D    CMP #NEWLIN
CDB0        F012    BEQ PHEAD3
CDB2        C923    CMP #PAGNUM     ;PLATZHALTER FÜR SEITENNUMMER?
CDB4        D008    BNE PHEAD2
CDB6        A518    LDA CURPAG
CDB8        2021CE  JSR PUTDEC      ;DRUCKE SEITENNUMMER
CDBB        4CAACD  JMP PHEAD1
CDBE PHEAD2 2060CE  JSR PUTC
CDC1        4CAACD  JMP PHEAD1
CDC4 PHEAD3 2060CE  JSR PUTC
CDC7 PHEAD4 A902    LDA #M2VAL      ;DRUCKE FREIZEILEN
CDC9        2085CD  JSR SKIP
CDCC        A905    LDA #M1VAL+M2VAL+1
CDCE        851D    STA LINENO      ;ZEILENNUMMER SETZEN
CDD0        A518    LDA ULVAL       ;UNTERSTREICHUNG WIEDER EIN
CDD2        F003    BEQ PHEAD5
CDD4        2062CF  JSR UNDOND
CDD7 PHEAD5 A519    LDA FAVAL       ;FETTSCHRIFT WIEDER EIN
CDD9        F003    BEQ PHEAD6
CDDB        20A1CF  JSR FATOND
CDDE PHEAD6 60      RTS
CDDF
*****************************************************************
SEITENFUSS AUSGEBEN

CDDF PFOOT  2084CF  JSR UNDOFD      ;UNTERSTREICHUNG AUSSCHALTEN
CDE2        20C3CF  JSR FATOFD      ;FETTSCHRIFT AUSSCHALTEN
CDE5        A902    LDA #M3VAL      ;DRUCKE FREIZEILEN
CDE7        2085CD  JSR SKIP
CDEA        A902    LDA #M4VAL      ;FUSSZEILE ERWÜNSCHT?
CDEC        F024    BEQ PFOOT4
CDEE        A000    LDY #0          ;JA, ALSO DRUCKE FUSSZEILE
CDF0 PFOOT1 C8      INY
CDF1        B95E00  LDA FOOTER,Y
CDF4        C90D    CMP #NEWLIN
CDF6        F012    BEQ PFOOT3
CDF8        C923    CMP #PAGNUM     ;PLATZHALTER FÜR SEITENNUMMER?
CDFA        D008    BNE PFOOT2
CDFC        A518    LDA CURPAG
CDFE        2021CE  JSR PUTDEC      ;DRUCKE SEITENNUMMER
CE01        4CF0CD  JMP PFOOT1
CE04 PFOOT2 2060CE  JSR PUTC
CE07        4CF0CD  JMP PFOOT1
CE0A PFOOT3 2060CE  JSR PUTC
CE0D        A901    LDA #M4VAL-1    ;DRUCKE FREIZEILEN
CE0F        2085CD  JSR SKIP
CE12 PFOOT4 A518    LDA ULVAL       ;UNTERSTREICHUNG WIEDER EIN
CE14        F003    BEQ PFOOT5
CE16        2062CF  JSR UNDOND
CE19 PFOOT5 A519    LDA FAVAL       ;FETTSCHRIFT WIEDER
CE1B        F003    BEQ PFOOT6
CE1D        20A1CF  JSR FATOND
CE20 PFOOT6 60      RTS
CE21
*****************************************************************
HEXZAHL IM ACCU DEZIMAL AUSGEBEN
```

```
CE21 PUTDEC 8500    STA D
CE23        8501    STA E
CE25        A230    LDX #$30        ;ZIFFER '0'
CE27 PUTDE1 C964    CMP #100        ;PRINZIP DES ABZÄHLENS
CE29        9007    BCC PUTDE2      ;HUNDERTER
CE2B        E8      INX
CE2C        38      SEC
CE2D        E964    SBC #100
CE2F        4C27CE  JMP PUTDE1
CE32 PUTDE2 8500    STA D
CE34        A501    LDA E
CE36        C964    CMP #100
CE38        9004    BCC PUTDE3
CE3A        8A      TXA
CE3B        2060CE  JSR PUTC
CE3E PUTDE3 A500    LDA D
CE40        A230    LDX #$30        ;ZIFFER '0'
CE42 PUTDE4 C90A    CMP #10         ;ZEHNER
CE44        9007    BCC PUTDE5
CE46        E8      INX
CE47        38      SEC
CE48        E90A    SBC #10
CE4A        4C42CE  JMP PUTDE4
CE4D PUTDE5 8500    STA D
CE4F        A501    LDA E
CE51        C90A    CMP #10
CE53        9004    BCC PUTDE6
CE55        8A      TXA
CE56        2060CE  JSR PUTC
CE59 PUTDE6 A500    LDA D           ;EINER
CE5B        0930    ORA #$30
CE5D        4C60CE  JMP PUTC
CE60
*****************************************************************
SCHREIBT DEN ACCU-INHALT AUF DEN DRUCKER

CE60 PUTC   209EEB  JSR PHXY        ;RETTE X UND Y
CE63        A611    LDX SKIFLG      ;ÜBERSPRINGE AUSGABE?
CE65        D009    BNE PUTC2
CE67        C95E    CMP #SEMIBL     ;ERSATZBLANK IN BLANK WANDELN
CE69        D002    BNE PUTC1
CE6B        A920    LDA #BLANK
CE6D PUTC1  208CE9  JSR OUTALL      ;GIB ZEICHEN AUS
CE70 PUTC2  20ACEB  JSR PLXY        ;ALTES X, Y
CE73        60      RTS
CE74
*****************************************************************
SEITENABBRUCH

CE74 BREAK  A407    LDY OUTP        ;OUTPUF SCHON LEER?
CE76        F008    BEQ BREAK1
CE78        A90D    LDA #NEWLIN     ;NEIN, FÜGE NEWLIN ZU
CE7A        9972D2  STA OUTBUF,Y
CE7D        207BCC  JSR TUP         ;AUSDRUCK
CE80 BREAK1 A900    LDA #0          ;VARIABLENINITIALISIERUNG
CE82        8507    STA OUTP
CE84        8508    STA OUTW
CE86        8509    STA OUTWDS
CE88        60      RTS
```

```
*************************************************************
FÜHERENDE LEERZEICHEN LÖSCHEN UND TIVAL SETZEN

CE89 LEADBL 2074CE JSR BREAK
CE8C        A000   LDY #0
CE8E LEADB1 C8     INY              ;FOR(I=1,INBUF(I)=BLANK;I=I+1)
CE8F        B90002 LDA INBUF,Y
CE92        C920   CMP #BLANK
CE94        F0F8   BEQ LEADB1
CE96        C90D   CMP #NEWLIN      ;IF(INBUF(I)()NEWLIN)
CE98        F008   BEQ LEADB2       ;    TIVAL=TIVAL+I-1
CE9A        98     TYA
CE9B        18     CLC
CE9C        6515   ADC TIVAL
CE9E        8515   STA TIVAL
CEA0        C615   DEC TIVAL
CEA2 LEADB2 A200   LDX #0           ;INBUF(J)=INBUF(I)
CEA4        88     DEY
CEA5 LEADB3 E8     INX
CEA6        C8     INY
CEA7        B90002 LDA INBUF,Y
CEAA        C90D   CMP #NEWLIN
CEAC        F006   BEQ LEADB4
CEAE        9D0002 STA INBUF,X
CEB1        4CA5CE JMP LEADB3
CEB4 LEADB4 9D0002 STA INBUF,X      ;NEWLIN ANFÜGEN
CEB7        60     RTS
CEB8
*************************************************************
RECHTEN RAND AUSRICHTEN

CEB8 SPREAD A50F   LDA DIR          ;AUFFÜLLEN VON RECHTS O. LINKS
CEBA        49FF   EOR #255         ;INVERTIEREN
CEBC        850F   STA DIR
CEBE        A508   LDA NEXTRA       ;NEE=NEXTRA
CEC0        850E   STA NEE
CEC2        A609   LDX OUTWDS       ;NHOLES=OUTWDS-1
CEC4        CA     DEX
CEC5        860C   STX NHOLES
CEC7        A407   LDY OUTP         ;I=OUTP-1
CEC9        88     DEY
CECA        8404   STY IY
CECC        98     TYA              ;J=MIN(MAXOUT-1,I+NE)
CECD        18     CLC
CECE        650E   ADC NEE
CED0        C9FE   CMP #MAXOUT-1
CED2        9002   BCC SPREA1
CED4        A9FE   LDA #MAXOUT-1
CED6 SPREA1 8505   STA JX
CED8
CED8 SPREA2 A404   LDY IY           ;WHILE(I(J)
CEDA        C405   CPY JX
CEDC        B050   BCS SPREA9
CEDE        A605   LDX JX
CEE0        B97C02 LDA OUTBUF,Y     ;UMSPEICHERN
CEE3        9D7C02 STA OUTBUF,X
CEE6        C920   CMP #BLANK
CEE8        D03D   BNE SPREA8
```

```
CEEA        A50F   LDA DIR
CEEC        D011   BNE SPREA4
CEEE
CEEE        A60E   LDX NEE          ;NB=(NEE-1)/NHOLES+1
CEF0        CA     DEX
CEF1        8A     TXA
CEF2        38     SEC
CEF3        A200   LDX #0
CEF5 SPREA3 E8     INX
CEF6        E50C   SBC NHOLES
CEF8        B0FB   BCS SPREA3
CEFA        860D   STX NB
CEFC        4C0CCF JMP SPREA6
CEFF
CEFF SPREA4 A50E   LDA NEE          ;NB=NEE/NHOLES
CF01        38     SEC
CF02        A200   LDX #0
CF04 SPREA5 E8     INX
CF05        E50C   SBC NHOLES
CF07        B0FB   BCS SPREA5
CF09        CA     DEX
CF0A        860D   STX NB
CF0C
CF0C SPREA6 A50E   LDA NEE          ;NE=NE-NB
CF0E        38     SEC
CF0F        E50D   SBC NB
CF11        850E   STA NEE
CF13        C60C   DEC NHOLES       ;NHOLES=NHOLES-1
CF15 SPREA7 A50D   LDA NB           ;BLANK EINFÜGEN
CF17        F00E   BEQ SPREA8
CF19        C605   DEC JX
CF1B        A605   LDX JX
CF1D        A920   LDA #BLANK
CF1F        9D7C02 STA OUTBUF,X
CF22        C60D   DEC NB
CF24        4C15CF JMP SPREA7
CF27
CF27 SPREA8 C605   DEC JX           ;ERHÖHE ZÄHLER
CF29        C604   DEC IY
CF2B        4CD8CE JMP SPREA2
CF2E
CF2E SPREA9 60     RTS
CF2F
*************************************************************
CENTRIERROUTINE

CF2F CENTER A000   LDY #0
CF31 CENTE1 C8     INY              ;WIEVIEL ZEICHEN IN DER ZEILE?
CF32        B90002 LDA INBUF,Y
CF35        C90D   CMP #NEWLIN
CF37        D0F8   BNE CENTE1
CF39        88     DEY
CF3A        8400   STY D
CF3C        A514   LDA RMVAL        ;PASST ZEILE IN BEGRENZUNG?
CF3E        18     CLC
CF3F        6515   ADC TIVAL
CF41        38     SEC
CF42        E500   SBC D
```

```
CF44        9004   BCC CENTE3
CF46        4A     LSR A
CF47        8515   STA TIVAL       ;JA, ALSO TIVAL SETZEN
CF49 CENTE2 60     RTS
CF4A CENTE3 A900   LDA #0          ;NEIN, ALSO TIVAL=0
CF4C        8515   STA TIVAL
CF4E        60     RTS
CF4F
*************************************************************
UNTERSTREICHEN EINSCHALTEN

CF4F UNDON  A407   LDY OUTP        ;FILL ODER NOFILL
CF51        F00F   BEQ UNDOND
CF53        C8     INY
CF54        A91B   LDA #ESCAPE
CF56        997C02 STA OUTBUF,Y
CF59        C8     INY
CF5A        A974   LDA #'t'
CF5C        997C02 STA OUTBUF,Y
CF5F        8407   STY OUTP
CF61        60     RTS
CF62 UNDOND A91B   LDA #ESCAPE     ;DIREKT
CF64        2060CE JSR PUTC
CF67        A974   LDA #'t'
CF69        4C60CE JMP PUTC
CF6C
*************************************************************
UNTERSTREICHEN AUSSCHALTEN

CF6C UNDOFF A407   LDY OUTP        ;FILL ODER NOFILL
CF6E        F014   BEQ UNDOFD
CF70        A91B   LDA #ESCAPE
CF72        997C02 STA OUTBUF,Y
CF75        C8     INY
CF76        A954   LDA #'T'
CF78        997C02 STA OUTBUF,Y
CF78        C8     INY
CF7C        A920   LDA #' '
CF7E        997C02 STA OUTBUF,Y
CF81        8407   STY OUTP
CF83        60     RTS
CF84
CF84 UNDOFD A91B   LDA #ESCAPE     ;DIREKT
CF86        2060CE JSR PUTC
CF89        A954   LDA #'T'
CF8B        4C60CE JMP PUTC
CF8E
*************************************************************
FETTSCHRIFT EINSCHALTEN

CF8E FATON  A407   LDY OUTP        ;FILL ODER NOFILL
CF90        F00F   BEQ FATOND
CF92        C8     INY
CF93        A91B   LDA #ESCAPE
CF95        997C02 STA OUTBUF,Y
CF98        C8     INY
CF99        A975   LDA #'u'
CF9B        997C02 STA OUTBUF,Y
```

```
CF9E        8407   STY OUTP
CFA0        60     RTS
CFA1 FATOND A91B   LDA #ESCAPE     ;DIREKT
CFA3        2060CE JSR PUTC
CFA6        A975   LDA #'u'
CFA8        4C60CE JMP PUTC
CFAB
*************************************************************
FETTSCHRIFT AUSSCHALTEN

CFAB FATOFF A407   LDY OUTP        ;FILL ODER NOFILL
CFAD        F014   BEQ FATOFD
CFAF        A91B   LDA #ESCAPE
CFB1        997C02 STA OUTBUF,Y
CFB4        C8     INY
CFB5        A955   LDA #'U'
CFB7        997C02 STA OUTBUF,Y
CFBA        C8     INY
CFBB        A920   LDA #' '
CFBD        997C02 STA OUTBUF,Y
CFC0        8407   STY OUTP
CFC2        60     RTS
CFC3
CFC3 FATOFD A91B   LDA #ESCAPE     ;DIREKT
CFC5        2060CE JSR PUTC
CFC8        A955   LDA #'U'
CFCA        4C60CE JMP PUTC
CFCD
*************************************************************
WANDELT KLEINE ALPHANUMERISCHE ZEICHEN IN GROSSE

CFCD FILT1  C940   CMP #$40
CFCF        3002   BMI FILT2
CFD1        29DF   AND #$DF
CFD3 FILT2  60     RTS
CFD4
CFD4               .END
 ERRORS= 0000
```

Peter W. Arps, 2000 Hamburg 73

Einkommensteuerberechnung 1983

Heft 19 dieser Zeitschrift enthielt das Grundprogramm für die Einkommensteuerberechnung 1980. In den Heften 21 und 29 erfolgten die Aktualisierungen für 1981 und 1982. Nachfolgend die zu aktualisierenden Programmzeilen für 1983. - Die wesentliche Änderung ist die Änderung der Vorsorgepauschale. Durch den Erfindungsreichtum des Gesetzgebers gibt es jetzt folgende Möglichkeiten: 1. den Normalfall, 2. die Ausnahme (für Beamte), 3. die Ausnahme von der Ausnahme (ein Ehegatte Beamter, der andere nicht), 4. die Ausnahme von der Ausnahme (wie Punkt 3).

```
110 DIM AL(1),BE(1),VB(1),WK(1),FB(5),KV(3),VV(1),LR(1),LF(1),GW(1)
740 IF X>60000 THEN X=60000
750 Z=Z+X:NEXT :HO=INT(Z*.09+.9)
760 IF BE(0)>0 OR VB(0)>0 OR LR(0)>0 THEN MX=1
770 IF BE(1)>0 OR VB(1)>0 OR LR(1)>0 THEN MX=MX+2
780 IF SK<3 THEN GOSUB 3500
790 IF SK=3 THEN GOSUB 3600
```

```
910 X=X1+INT((BE(0)+BE(1))*.09)
950 SA=SA-H3:Z=Z+H3:IF SA<0 THEN Z=Z+SA:GOTO 970
960 SA=SA/2:X=H3/2:SA=SA-X:Z=Z+X:IF SA<0 THEN Z=Z+SA
1600 REM  VZ'83 DELETE
1780 PRINT "Verlust-Vortrag/-Ruecktrag 1978-1985":gosub 2160
1850 z=z+kd*432+ku*216
2495 print "davon aus Beamtentaetigkeit";:gosub 2170:be(i)=x
2740 input "2.1.1919 geboren (j=ja/n=nein)  n^ü^ü^ü";x$:if x$="n" then 2
770
3280 x=x(14):if x>0 then print# 4:x$="- Verlust-Vor/Ruecktr.1978-1985":g
osub 2930
3340 x=kd*432+ku*216:if x>0 then print# 4:x$="- Kinderfreibetrag":gosub
2930
3500 h1=2340+kd*600+ku*300:h3=h1
3510 if mx=1 then 3530
3520 h2=1170+kd*300+ku*150:goto 3550
3530 h1=1000+kd*600+ku*300
3540 h2=1000+kd*300+ku*150
3550 z=h1:if h0<h1 then z=h0
3560 y=h2:if h0<h2 then z=h0
3570 vs=fnd(z+y):if vs<p3 then vs=p3
3580 return
3590 :
3600 if mx=1 then if al(1)=0 then mx=3
3605 if mx=1 or mx=2 then 3710
3610 h1=4680+kd*600+ku*300:h3=h1
3620 if mx>0 then 3640
3630 h2=2340+kd*300+ku*150:goto 3660
3640 h1=2000+kd*600+ku*300
3650 h2=2000+kd*300+ku*150
3660 z=h1:if h0<h1 then z=h0
3670 y=h2:if h0<h2 then y=h0
3680 vs=fnd(z+y):if vs<p3*2 then vs=p3*2
3690 return
3700 :
3710 if mx=1 then i=1:j=0:gosub 3850
3720 if mx=2 then i=0:j=1:gosub 3850
3730 h0=v:gosub 3640:w=vs
3740 h3=4680+kd*600+ku*300
3750 h1=1000+kd*600+ku*300
3760 h2=1000+kd*300+ku*150
3770 x=v:if x>h1 then x=h1
3780 y=v:if y>h2 then y=h2
3790 vs=u+x+y:x=vs:if vs>h3 then vs=h3
3800 x=(x-vs)/2:if x<0 then x=0
3810 if x>h3/2 then x=h3/2
3820 vs=fnd(vs+x):if vs<p3*2 then vs=p3*2
3830 if vs<w then vs=w
3840 return
3850 x=al(i)-600:if x>60000 then x=60000
3860 y=al(j)-600:if y>60000 then y=60000
3870 u=int(x*.18+.9):v=int(y*.09+.9)
3880 return
```

Bücher

Zech, R.: Die Programmiersprache FORTH. Franzis-Verlag, München 1984, ISBN 3-7723-7261-9, 312 S., DM 68,--. Der Untertitel dieses brandneuen Buches lautet 'Eine kompakte Darstellung der Programmiersprache mit vielen Beispielen', und es gliedert sich in folgende Hauptabschnitte: Aufbau des FORTH, Programmieren in FORTH, Programmstrukturen und strukturierte Programmierung, das Vocabulary, FORTH für Fortgeschrittene, FORTH I/O und die Installation auf konkreten Systemen. Es folgen 11 Anhänge, ein Literaturverzeichnis sowie ein knappes Sachverzeichnis. Verstreut im Text finden wir 47 schematische Abbildungen, die vor allem Speicherbelegungen oder Programmablaufpläne verdeutlichen, sowie 36 Tafeln vorwiegend mit der Erklärung von Definitionen wichtiger FORTH-Worte.

Für die Programmiersprache FORTH, speziell FIG-FORTH, konnten bisher im wesentlichen nur englische Bücher empfohlen werden, wenn wir vom bereits besprochenen 'FORTH-Handbuch' (E. Floegel, Hofacker-Verlag) einmal absehen. Sehr beliebt ist von Leo Brodie das 'Starting FORTH' und als Orientierungsbuch 'FORTH User's Manual' (Rockwell-Buch für das AIM 65-FORTH). Das von Zech verfaßte Buch ist anders als alle anderen. Es erhebt zwar den Anspruch, auch ein Anfängerbuch zu sein, in Wirklichkeit schaut es der Sprache von Anfang an sehr 'unter die Haube' (Zitat nach Brodie). Nach dem Urteil des Rezensenten, der selbst viel zu FORTH geschrieben hat, ist das Buch mehr ein Werk für Fortgeschrittene und Systemprogrammierer. Wenn man es mit dieser Erwartung aufschlägt, dann findet man wirklich handfeste Information auch zu schwierigen Gegenständen der Sprache. Zu kritisieren ist dann erstens, daß der Autor einen manchmal akademischen, sehr englisch-deutschen Computer-Jargon schreibt, der den Anfänger stören könnte. Zweitens wurde die nachlässige Arbeit des Setzers zu wenig korrigiert. Drittens fehlt in diesem Buch jede Referenz auf Quellen im Text. Es hätte sich im laufenden Text gehört, die jeweils benutzte Vorlage zu zitieren. - Auch in mc werden nur wenig Quellen (außer mc) zitiert.

Derick, M.: Baker, L.: FORTH-Encyclopedia. Mountain View Press, CA., 1982, ohne ISBN, 333 S, DM 104,--. Für das FIG-FORTH wird in alphabetischer Reihenfolge jedes Wort auf einer oder mehreren Textseiten dargestellt. Es sind jeweils alle Informationen zusammengetragen, die zu einem Wort gehören: Eingangs- und Ausgangsbedingungen (Parameter), Beschreibung der Funktion, mögliche Fehlermeldungen, Aussprache des Namnes, Hinweis auf verwandte Worte, Äquivalent im FORTH-79-Standard und ferner Definition des Wortes als Secondary oder Primitive (maschinensprachliches Wort), zusammen mit einem Programmablaufplan! Es handelt sich damit um eine wirkliche Enzyklopädie, die mühseliges Nachschlagen in verschiedenen Quellen oder ein Disassembleren des FORTH erspart und die zuverlässige und übersichtliche Information gibt. Der recht hohe Preis für die Anschaffung ist zu vertreten, wenn man häufiger einmal nachschlagen muß.

Kernighan, B. W.; Ritchie, D. M.: The C Programming Language. Bei Prentice Hall, Englewood Cliffs, N.J. (USA) 1978, ISBN 0-13-110163-3, 228 S., DM 84,20. Die Zeitschrift BYTE hat im August alljährlich das Schwerpunktthema Sprachen. 1983 wurde 'C' behandelt. Dieses Buch aus der Feder der Sprachschöpfer wurde dort als ein 'Muß' bezeichnet, aber als etwas schwierig klassifiziert. Das kann der Rezensent eigentlich nicht bestätigen, es ist in so klarer Sprache geschrieben, daß man die Autoren fast vor sich dozieren hört.. 'C' ist vorwiegend eine Sprache für Systemprogrammierer, stark pointer-orientiert, höhere mathematische Funktionen kennt sie im Kernal nicht, dafür erlaubt sie, weil selbst weitgehend in 'C' geschrieben (nur der Kern ist maschinenabhängig), die Implementierung verbreiteter Betriebssysteme auf jedem geläufigen Rechner und natürlich auch von 'C' selber. Eine Schulung der Fa. Digital Research (Schöpfer des CP/M) im Zusammenhang mit der SYSTEMS '83 in München führte das sehr deutlich vor Augen. Dort schreibt man Betriebssysteme in 'C', speziell auch anwenderfreundliche Softwareinterfaces, die mit denen der LISA von APPLE vergleichbar sind. - Im Zusammenhang mit dem großen Erfolg des IBM-PC und angesichts der Sprachimplementierung auch für den 68000 sollte man dieses solide Buch zur Einarbeitung heranziehen. - Hinweise auf Betriebserfahrungen und erhältliche Implementierungen sind willkommen.

Förster, H.-P.: Bilschirmtext. Humboldt-Taschenbuch, München 1983, ISBN 3-581-66457-7, 158 S., DM 7,80. Im Jahre 1984 wird Bildschirmtext bundesweit eingeführt. Das Buch beschreibt allgemeinverständlich die wichtigsten Begriffe der Kommunikation zwischen Anwendern und Anbietern von Bildschirmseiten mit den Btx-Zentralen, es erklärt, wie Btx funktioniert, wie man sich daran beteiligen kann und wie die Kosten sind. Daneben sind die Standards und möglichen

Gerätekonfigurationen erklärt. Der Text wird durch zahlreiche Abbildungen und Übersichten ergänzt. Für eine erste zusammenhängende Übersicht ist das Buch empfehlenswert, auch wenn man damit rechnen sollte, daß die heute doch noch sehr teuren Endgeräte künftig von preiswerteren Personal Computern abgelöst werden, die Btx-fähig sind.

Wozu Heimcomputer? Eine Verbraucherfibel des Deutsches Video Instiut e.V., Berlin 1983, ISBN 3-923220-05-7, 66 S., DM 3,30. Dieses Buch lehnt sich an eine hervorragende Vorlage aus den USA an. Es ist vor allem zur Verteilung/zum Verkauf durch den Rundfunk- und Fernsehfachhandelt an ein breites Publikum gedacht. Es informiert mit klaren, einfachen deutschen Worten, worauf der Laie achten soll, ehe er einen Heimcomputer kauft. Empfehenswert zur ersten Orientierung.

Treitz, N.: Besser Programmieren mit dem VC 20. Das Buch zum Handbuch. Lehrmittelverlag Hagemann, Düsseldorf 1983, ISBN 3-544-53001-0, 186 S., DM 18,30. Das Buch wendet sich an Anwender des Volks-Computers, die mehr als spielen wollen. Es setzt keine Fachkenntnisse voraus, sondern baut auf den Informationen des Rechner-Handbuches auf. Es enthält knappe Beschreibungen (keine langatmigen Programme zum Nachtippen), die zu wichtigen Befehlen des BASIC hilfreiche Tricks enthalten. Es wird gezeigt, wie man die Fähigkeiten der Maschine für Grafik und die Zeichengenerierung ausnutzen kann. Der Band ist lesbar geschrieben und regt vor allem den Anfänger zum Weitermachen an.

Commodore 64 Programmer's Reference Guide. Herausgegeben von Commodore Business Machines Inc. und Howard W. Sams & Co, Indianapolis 1983, ISBN 0-672-22056-3, 487 S, U$19,95 bzw. DM 98,--. Commodore sah sich oft peinlichen Fragen gegenüber, wie es mit der Dokumentation bestellt sei. Das vorliegende Ringbuch zeigt, daß man hervorragend dokumentieren kann. Wir finden die Hauptabschnitte: Programmierregeln und Befehlsvorrat (BASIC), Graphik auf dem VC 64, Geräusche und Töne, Grundlagen der Maschinenprogrammierung, Ein- und Ausgabe, Anhänge, Index und Schaltplan des Computers. Es sind ferner die Datenblätter der neueren Chips nebst Registerplänen und Pinbelegungen in der Maschine enthalten, und zwar für die CPU 6510, 6526 CIA, 6566/67 VIC-II-Chip und 6581 SID. Dokumentiert ist die Belegung des System-RAMs, Namen und Einsprungspunkte der System-Routinen, ihre Beschreibung mit ausgeführter Funktion und Eingangs- und Ausgangsparametern. Wenn man einmal vom nicht enthaltenen Listing des Betriebsprogrammes und es BASIC absieht (das findet man in einem DATA BECKER-Buch), so hat man mit diesem Handbuch eine fast vollständige empfehlenswerte Beschreibung des beliebten Volks-Computers.

Angerhausen, M.; Becker, A.; Englisch, L.; Gerits, K.: 64 intern. DATA Becker-Buch, Düsseldorf 1983, ohne ISBN, 306 S., DM 69,--. An anderer Stelle ist das Buch 'Commodore 64 Programmers Reference Guide' besprochen worden. Das Buch '64 intern' füllt die dort erwähnte Lücke, es enthält die wieder-assemblierte und nachkommentiere Quell-Liste des Betriebssystems und des BASIC. Das ist der Hauptteil. Die Kommentare sind eigentlich zu knapp ausgefallen, und die Algortihmen sind zu wenig erkannt und aufgeklärt worden. Wir finden oft auf 10 bis 30 Statements nur nackt disassemblierten Maschinencode. Aus eigenem Nachvollzug darf der Rezensent berichten, daß diese Liste dem Anwender noch viel Mühe des Nachdenkens überläßt. Gleichwohl ist zu sagen, daß eine Programmliste besser ist als gar keine oder als mehrtägige Arbeit am Disassembler. Zu den positiven Eigenschaften des Buches gehört die Beschreibung der Register, Funktionen und Pins der Hardware, die man inzwischen auch im o.e. Reference Guide findet. Beide Bücher gehören eigentlich zusammen in die Hand eines system-orientierten Betreibers. Die Geldausgabe ist dann schon beachtlich.

Wagner, H. C.: More on the Sixtyfour. Hofacker-Verlag, Holzkirchen 1983, ISBN 3-88963-183-5, 149 S. Es ist ein Buch für die maschinensprachliche Programmierung des VC 64, das häufig gewünschte Dienstprogramme unter Mitbenutzung des Betriebssystems enthält: Zahlenein- und -ausgabe, Texteingabe, Rekursion, File-System nebst SAVE und LOAD, RS232-Interface, Echtzeituhr, neue BASIC-Befehle, hohe Bildauflösung, Hardcopy vom Bildschirm, Centronics-Interface. Die enthaltenen zahlreichen Assembler-Listen sind leider nicht kommentiert, so daß der Nachvollzug der Programmlogik erschwert ist. Es sind jedoch die jeweils benutzten Systemvariablen, Einsprungspunkte und Gleichsetzungen dokumentiert. Mit ihnen und ggfs. weiterer Systemliteratur (s.o.) sollte es möglich sein, die Routinen mit Gewinn auf dem VC 64 einzusetzen.

Reinking, E. F.: Mathematische Routinen VC 20. Vieweg-Verlag, Braunschweig 1983, ISBN 3-528 04234-6, 78 S., DM 24,50. Dieser Band aus der Reihe 'Anwendung von Microcomputern' wendet sich vor allem an Studierende und Praktiker der Fachgebiete Elektrotechnik/Elektronik

und enthält häufig benötigte Berechnungsprogramme in BASIC: Quadratische Gleichung, Rechnen mit komplexen Zahlen, Nullstellen einer Funktion, Polynomapproximation. Typisch elektrotechnisch sind: Berechnung von Ersatzschaltbildern, Filter- und Anpaßschaltungen, Netzwerksanalyse. Jeder Abschnitt enthält die Problemstellung, die benutzte Formel, das Programm, den schematischen Programmablaufplan sowie den Ergebnisausdruck für Beipiele, so daß man auch das richtige Arbeiten des Programmes kontrollieren kann. Die vielen fertigen Lösungen werden sicher erfreuten Zuspruch finden.

Rector, R.; Alexy, G.: Das 8086/8088-Buch. te-wi-Verlag, München 1982, ISBN 3-9211803-11-x, ca. 520 S. DM 69,--. Der Untertitel lautet: Programmieren in Assembler und Systemarchitektur. Es handelt sich um die deutsche Übersetzung eines McGraw-Hill-Buches, wie es auch für andere Prozessoren z.B. von Leventhal geschrieben wurde, allerdings unter Beisteuerung aktueller Unterlagen durch die Fa. Siemens. - Bekanntlich ist der 8088/8086 das Herz des erfolgreichen IBM-PC, Anlaß also, sich auch gewissermaßen 'mit der Konkurrenz' zu befassen, um auf dem laufenden Stand zu bleiben. Es ist ein Buch, an dem man die unterschiedliche Architektur der CPU's im Vergleich zu den mit '6' beginnenden Typenreihen sehr klar erkennen kann. Wer sich mit dem 68000 befaßt hat, wir mehr für dessen Architektur eingenommen sein, gleichwohl hat der 8086 Befehle, die man dort so bequem nicht hat. Und schließlich gibt es auf 8080, Z80 aufbauend eine große Auswahl an Software. Und man sollte weiter sehen, daß viele geläufige und erfolgreiche Anbieter mit einer anderen CPU (z.B. Commodore) den Computer als E/A-Prozessor für mit '8' bestückte Karten einsetzbar machen. - Das Buch ist mit seinen umfangreichen Darstellungen zur Hardware und zur Dokumentation der Befehle kein Lehrbuch für die 8086 Assembler-Programmierung, dafür sind der Beispiele zu wenig. Aber es taugt gut, um die grundsätzlichen Fähigkeiten zu erkennen. Wir finden folgende Hauptkapitel: Programmieren in Assembler, Entwicklungsgeschichte der CPU, Register und Befehle, Befehle nach Funktionsgruppen, Interrupts, Programmbeispiele (12 Seiten), Architektur und E/A, Systeme mit 1 Prozessor, Multibus, Systeme mit mehreren Prozesssoren und Anhänge.

Alt, H.; Schumny,H. (Hrsg.): Kryptologie (HP-41 C/CV). Vieweg-Verlag, Braunschweig 1983, 94 S., DM 24,80. In der Reihe Programmbibliothek Taschenrechner ist dieses der Band 7. Er befaßt sich mit verschiedenen Arten der Textmanipulation, nämlich Chiffrierung, Textverschlüsselung, Datenschutz, Dechiffrieren, Morsealphabet. Für diese speziellen Sachgebiete sind jeweils vollständig gelistete Programme nebst Ablaufdiagrammen enthalten.

Schumy, H. (Hrsg.): Programme für den Einplatinencomputer TM 990/189 (16-Bit-Prozessor TMS 9980A). Ebenfalls bei Vieweg, 1983, ISBN 3-528-04255-9, 66 S. Für diesen Lerncomputer werden Dienstprogramme geboten: Verschiebbare Maschinenprogramme, HEXEDIT zur Programmentwicklung, AD-/DA-Wandlereinheit, genaue Zeitschleifen, Interruptverarbeitung, einfache Arithmetik-Routinen, Morse-Trainer.

Ortlieb, C. P.: Lehr- und Übungsbuch für den Rechner SHARP PC-1500. Verlag Vieweg, Braunschweig 1983, ISBN 3-528 04245-1, 150 S., DM 29,60. Für diesen Taschenrechner mit seinem komfortablen BASIC werden 25 lauffähige Programme vor allem aus dem mathematisch-naturwissenschaftlichen Bereich vorgestellt. Es wendet sich vor allem an Betreiber ohne Programmierkenntnisse und führt schrittweise in das BASIC ein. Es ergänzt damit die Bedienungsanleitung.

Poole, L. (Hrsg.): Praktische BASIC-Programme für den IBM Personal Computer. Bei McGraw-Hill, Hamburg 1983, 172 S., DM 29,80. Es ist das erste deutschsprachige Buch der deutschen Verlagsniederlassung zum Thema Mikrocomputer. Es sind 34 Programme enthalten mit Beschreibung ihrer Aufgabe, Beispielen, Übungsproblem, Programmanmerkung und Programmliste. Wir finden Programme für verschiedene Zinsberechnungen, Bilanz- und GuV-Analyse, Methode des kritischen Pfades, PERT, Transportalgorithmus, Warteschlangen, Markov-Analyse, Wirtschaftliche Bestellmenge und Losgröße, statische Schätzungen, Chi-Quadrat,Interpolation nach Newton und Lagrange u.a.m.. Die Programme decken damit häufiger benötigte Problemlösungen auf eine klare Weise ab.

Kohler, H.: FORTRAN-Trainer. Verlag Vieweg, Braunschweig 1983, ISBN 3-528 04257-5, 261 S., DM 24,80. Der Untertitel lautet: Skriptum für Hörer aller Fachrichtungen ab 1. Semester. Im Vergleich zu vielen Lehrbüchern, die nur Kurzhinweise geben können, wird eine anwendungsbezogene Vertiefung in der Darstellung programmtechnischer Zusammenhänge vorgenommen. Der Stoff reicht von Konstanten und Variablen bis hin zu anspruchsvollen algorithmischen Übungenmit Formelprogrammierung und Textananlyse. Das erste Drittel des Buches enthält jeweils Aufgaben-

stellungen, die zur intensiven Eigenbefassung motivieren sollen. Der zweite Teil enthält die Lösungn dazu, zusammen mit dem Ausdruck von Ergebnissen.

Schneider, W.: Einführung in die Anwendung des Betriebssystems CP/M. Verlag Vieweg, Braunschweig 1983, ISBN 3-528-04252-4, 146 S., DM 29,80. Das Buch aus der Feder dieses erfahrenen und gründlichen Autors wendet sich an Leser, die eine grundlegende Einführung in die Anwendung des CP/M wünschen. Es enthält dazu 76 Beispiele und 48 Übungsaufgaben. Die wichtigsten Kommandos werden in einzelnen Kapiteln abgehandelt. Die Reihenfolge wurde nach Möglichkeit so gewählt, wie es der Abfolge beim Arbeiten mit CP/M entspricht. Der Text ist wiederum sauber gegliedert und an den wichtigen Stellen zum schnelleren Erkennen eingerahmt. Die hinzukommenden Betreiber eines CP/M-Systems sollten dieses Hand- und Lehrbuch ernsthaft einplanen.

Editorial

Am Ende dieses Jahres stellen viele Anbieter von Hardware fest, daß die Marktentwicklung stürmisch war und anders verlief, als man es zunächst erhoffte oder plante. Durchaus große und interessante Hersteller von Computern haben nach finanziellen Mißerfolgen die Tore schließen oder diesen Produktionszweig radikal absägen müssen. Andere traten trotz Liefervorbereitungen gar nicht erst an den Markt. Und weiter andere haben Spaß am Geschäft und machen Gewinne, so Commodore und IBM. Ihre Besorgnis gilt derzeit mehr den Beschaffungsschwierigkeiten bei Bauteilen, um die Nachfrage annähernd befriedigen zu können. Bei Commodore sagt man: 'Wir haben fünf Jahre gebraucht, um eine Million Computer verkaufen zu können. Jetzt im sechsten Jahr verkaufen wir weltweit drei Millionen'.Wenn man sich diesen Sprung ansieht, dann sind die logistischen Probleme schon klar. Und man sagt dort weiter; 'Wir waren nicht zu den Preisermäßigungen gezwungen, sondern konnten mit dem Eintreten der erwarteten Gewinne unsere Einsparungen an den Endverbraucher weitergeben'. So Harald Speyer (Commodore-Chef) sinngemäß in einem persönlichen Gespräch im Oktober. Und vom IBM-PC dürften bis August 83 schon über 800.000 Stück verkauft worden sein.

Wir sollten uns also bewußt machen, daß für den Bereich der Personal- und Heimcomputer schnelle Marktänderungen nach wie vor typisch sind. Trotz des vielfältigen Geräteangebotes werden auf Sicht nur wenige Firmen einen bedeutenden Marktanteil haben. Zu ihnen werden sicher die beiden vorerwähnten gehören. Und im Sinne einer Austauschbarkeit von Informationen ist die Marktbereinigung oder Konzentration wohl auch gar nicht so schlecht, denn es gibt noch viele Klein- und Mittelbetriebe, die noch ohne eigenen Computer arbeiten. Ihnen wird die Wahl und die Inbetriebnahme erleichtert, wenn sie bei ihren Lieferanten oder Vereinigungen Disketten mit branchenspezifischer Information erhalten können, die sie mit ihrem System lesen und verwerten können. Kompatibilität ist also wahrscheinlicher, wenn man im großen Geräteangebot nicht so leicht ein unkompatibles Gerät beschaffen kann. Immerhin gibt es ja inzwischen einige Standards, die sich durch Marktführerschaft eingebürgert haben, z.B. Centronics-Schnittstelle, IEEE488bus oder IBM-kompatible Floppies. - Damit soll nicht in Abrede gestellt sein, daß auch die Marktführer Wettbewerb mit besseren Ideen haben sollen, der sie zu zeitgemäßen Anpassungen zwingt.

Weitere Änderungen sollte man auch bei der Ausstattung der Geräte und ihrem Anwenderinterface erwarten. Die Halbleitertechnik erlaubt mehr Leistung auf weniger Raum und zu geringeren Kosten. Insbesondere ist anzunehmen, daß sich auch die Computerhersteller des vor der Einführung stehenden Bildschirmtextes (Btx) annehmen werden. Vorbereitungen werden in jedem Fall schon bei den traditionellen Fernseh-Herstellern getroffen, allerdings noch mit recht teuren Gerätelösungen. Die Btx-fähigen Personal- und Heimcomputer kommen damit auf uns zu. - Und schließlich verändert auch die Softwareentwicklung den Computer. Wie der Bericht 'Erste Erfahrungen mit LISA' in Heft 33 schon erkennen ließ, hat Apple ein besonders anwenderfreundliches Software-Interface Mensch-Maaschine geschaffen, das Benutzer ohne jegliche Programmierkenntnisse durch die Dienstleistungen des Systems führt (mehr über LISA wird für das Februar-Heft erwartet). Etwa vergleichbare Leistungen erbringen die Systempakete der Fa. Digital Research Inc., etwa mit dem 'Visual Information Processor' (VIP) und den graphischen Systemen GSX, DR GRAPH, DR DRAW und DR ACCESS 10 (DR steht für den Firmennamen), alle in der Programmiersprache 'C' entwickelt (Literaturhinweis in diesem Heft). 'C' ist weitgehend selbst in 'C' formuliert. Nur der Kern der Sprache ist maschinenabhängig. Damit ist ein hoher Grad an Portabilität von Betriebssystemen und Programmen zwischen zwei Rechnern gewährleistet, sobald der maschinensprachliche Kern für geläufige CPU's implementiert ist. - Man darf also erwarten, daß die anwenderfreundlichen Software-Interfaces in einiger Zeit für alle besseren und geläufigen Rechner ein 'Muß' sind.

Soweit Überlegungen, die beim und nach dem Besuch der SYSTEMS '83 in München eintraten. Es soll hier nicht weiter von einzelnen Herstellern und Produkten berichtet werden, zumal einige auch schon wieder vom Markt verschwunden sind. - Erwähnenswert mag sein, daß der sog. BBC-Computer ACORN B mit CPU 6502 vor allem in England mit einigen hunderttausend abgesetzten Exemplaren ein großer Erfolg geworden ist. Möglicherweise gewinnt 6502 mit einer in diesem Lande in Vorbereitung befindlichen Fernsehserie weitere Freunde. - Die Zeitschrift mc bringt im Heft 1/1984 einen AIM 65 auf Europakarte mit 64K RAM. Wie beim Franzis-Verlag üblich, wird eine entsprechende Platine beim Autor erhältlich sein.

Beim Besuch der SYSTEMS stieß der Herausgeber auf etwa vier neue Computer-Zeitschriften, die für ein breiteres Publikum bestimmt sind. Man wird sehen, ob ihre z.T. großen Verlage vom Trittbrett in die zugfreie Zone des schon fahrenden Busses vordringen können. Wir finden ja z.B. auch in den USA eine Vielzahl von Zeitschriften, vor allem mit viel Werbung und meist freundlichen Produktberichten. Der Markt drüben ist allerdings wesentlich größer, entwickelter und werbefreundlicher als der hiesige. - Es bleibt festzustellen, daß trotz des vermehrten Angebotes der Bedarf für die gründlicheren Darstellungen bleibt, die vom 65xx MICRO MAG in der redaktionellen Arbeit angestrebt werden. In Zeitschriften, die sich mit vielen zehntausend Exemplaren an die Leser wenden, sind die (oft nur wenigen) Textseitenfür größere Auslistungen einfach zu teuer. Die größte Zeitschrift am Markt, die zugleich die meiste Werbung enthält, schafft es offensichtlich nicht mehr, monatlich ein Verzeichnis der vielen Inserenten aufzustellen, obwohl es ja möglich sein müßte, wenn man den Computer propagiert, ihn für solche Zwecke zu benutzen. Die Suche nach einer bestimmten Anzeige wird damit enorm erschwert, und der Wert einer Anzeige wird vermindert, wenn man sie nicht finden kann. - Im Sinne solider Darstellungen sind die Leser dieser Zeitschrift weiterhin gebeten, über ihre Arbeiten zu berichten und die Themen möglichst im Vorwege mit dem Herausgeber abzustimmen.

Auch bei Büchern wird zunehmend unschuldiges Papier bedruckt. Und Buchbesprechungen sind rar, weil man möglichst nur Bücher aus dem eigenen Verlag bespricht. Die zahlreichen Buchbesprechungen in diesem Heft zeigen, daß es viele interessante Titel am Markt gibt, die die Arbeit befördern. Störend ist der oftmals sehr hohe Preis, speziell für USA-Bücher.

Das Jahr klingt aus. Der Herausgeber wünscht seinen Lesern, Autoren und Geschäftsfreunden
FROHE FESTTAGE UND EIN ERFOLGREICHES GLÜCKLICHES NEUES JAHR!
Er bedankt sich zugleich herzlich für die Zusammenarbeit im alten Jahr.

Dipl.-Ing. Harald Grundwald, 2150 Buxtehude

64K DRAM am AIM

Aufbau und Anwendungen

Die in /1/ von Herrn Pobloth veröffentlichte Schaltung einer 64K DRAM-Karte wurde erfolgreich nachgebaut. Der Aufbau erwies sich als völlig problemlos. Dennoch zeigte sich bei der ersten Inbetriebnahme zunächst nur ein geringer Erfolg, das System hängte sich auf. Nach genauem Studium der Kennblätter des TMS 4500A und Überprüfung des Timings stellten sich folgende Störquellen heraus:

- der Eingang $\overline{\text{REFREQ}}$ des TMS 4500A muß mit einem Pull-up-Widerstand versehen werden,
- die Phase Ø 1 (Pin 3 an J3) ist nicht gepuffert und hat nur geringe Flankensteilheit. Phase $\overline{\text{Ø2}}$ (Pin Y an J3) ist hier besser.
- die Betriebsspannung war sehr unsauber (parallele Leitungsführung). Jeder Baustein wurde unmittelbar an seiner Versorgung mit 10 nF geblockt.

Nach Beseitigung der vorgenannten Störquellen funktionierte die Schaltung einwandfrei. Der Vollständigkeit halber sollte jedoch erwähnt werden, daß die geforderte 2 MHz-Frequenz am Pin 5 des 7474 nicht greifbar ist. Dieser teilt die Quarzfrequenz durch vier, so daß an Pin 5 lediglich eine um 250 ns gegenüber Pin 9 phasenverschobene 1 MHz vorliegt. Da es aber nur auf die fallende Flanke innerhalb der ersten 500 ns des CPU-Taktes ankommt /2/, sind alle Forderungen wieder erfüllt.

Nachfolgend einige Tips und Software-Splitter als Denkanstoß für eine sinnvolle Nutzung beider

32K-Bänke: Wer keinen Steuer- oder Portausgang der User-VIA verschenken will, der kann z.B. CB2 des 6520 (PIA) am Pin 19 verwenden /3/. Mit $E0FE LDA #30 und in $E10B LDA #34 werden CA2/CB2 des 6520 auf '0' initialisiert.

Wer REMON /4/ implementiert hat, sollte ausreichend freien Raum im Bereich E000 bis EFFF haben, um die folgenden Routinen dort unterzubringen und zu erproben:

```
        *=$E19E
        JSR HELP5       ;MONITOR-ERWEITERUNG
-------------------------------------------------

HELP5   CMP #'0'        ;TASTE '0' FUER BANKSWITCH
        BNE HELP5A
        LDA $AC03       ;SCHALTET CB2 DES 6520 UM
        EOR #8
        STA $AC03
        AND #8
        BNE HELP5B
        LDA #31         ;UND ZEIGT BANK NR. 1
        .BYT $2C
HELP5B  LDA #32         ;ODER BANK NR. 2 AN
        JSR $E97A
        JMP $E182
HELP5A  JMP ($A40A)     ;MOLINK DES REMON
```

Mit Hilfe der vorstehenden Routine kann über Taste '0' aus der Monitorebene die Bank umgeschaltet werden.

Die Monitor-Routine MREAD in $FAD0 sollte u.a. aus /5/ für die Eingabe von BASIC-Programmen vom Editor her bekannt sein. Um in einer Bank BASIC zu editieren und anschließend über den User- Input-Vektor in die andere Bank zu laden, mußte MREAD angepaßt werden:

```
MREAD1  LDA $AC03       ;UMSCHALTEN
        EOR #8
        STA $AC03
        JSR MREAD
```

Der User Input-Vektor wird in der BASIC-Bank auf MREAD1 gesetzt. Mit LOAD IN=U beginnt dann das Laden.

MREAD wird auch beim Assemblieren genutzt. Es sollte also eine Trennung von Source- und Object-Code in zwei Bänke möglich sein. Daran wird z.Zt. noch gearbeitet. Für Hinweise wäre der Autor dankbar.

Mit den Routinen SWIT1 und SWITCH können Unterprogramme in der jeweils anderen Bank aufgerufen werden. Die Unterprogramm-Adresse wird dabei vom Hauptprogramm in $A40E und A40F als Vektor abgelegt. Der Ablauf ist wie folgt:

```
SWITCH  LDA $AC03       ;ZURUECKSCHALTEN
        EOR #8
        STA $AC03
        LDA $A42A       ;WERT UEBERGEBEN
        RTS

SWIT1   LDA $AC03       ;UMSCHALTEN
        EOR #8
        STA $AC03
        JMP ($A40E)     ;INDIR. SPRUNG GEM. VEKTOR
```

	Bank 1	Bank 2
RTS-Adr. in Stack1	Hauptprogr. Vektor $A40E $A40F laden JSR SWIT1 : : Fortsetzung	Vektor→ Unterprogr. : : JMP SWITCH

EPROM (oder geschütztes RAM)

SWIT1 Bankumschaltung
Ind.-Sprung gem. Vektor

SWITCH Bankumschaltung
ggf. Wertübergabe in $A42A
RTS zur letzten Stackadr.
dieser Bank

Man kann hier beliebig springen, da die Rücksprungadresse immer im Stack der jeweiligen Bank abgelegt ist und erst nach Rückkehr (JMP SWITCH) wieder erreichbar ist. Wer probieren will, ohne gleich neue EPROMs zu brennen, der kann die freien 1K bzw. 4K RAM on board gem. /6/ in einen Bereich ab $8000 schalten. Ist evtl. noch eine 16K RAM-Karte vorhanden, so kann diese beispielsweise in den Bereich B000 bis EFFF geschaltet werden. Die 4K RAM on Board schaltet man über eine entsprechende Logik zusammen mit einem Boot-EPROM in den Bereich Fxxx.

Beim Kaltstart initialisiert das Boot-EPROM eine Urlade-Routine in einen 32K DRAM-Bereich. Der Urlader seinerseits blendet das Boot-EPROM aus sowie die 4K RAM on board ein und lädt das gewünschte Betriebssystem von einer Floppy oder über Ports von einer externen EPROM-Karte. Damit sind einige Möglichkeiten genannt, die diese 64K DRAM-Technik mittels DRAM-Controller bietet. – Betrachtet man die vergangenen Ausgaben des MICRO MAG (REMON /4/, EXEDIT /7/, SCREEN /8/ und ROFF /9/), so wird ein großer Speicher immer zwingender erforderlich. Hier bietet sich jedem 6502-Benutzer die von Herrn Pobloth erarbeitete Minimalkonfiguration zum preiswerten und einfachen Erweitern des Systems an.

Literatur

/1/ 64K DRAM, 65xx MICRO MAG Nr. 30, 1983
/2/ 64K DRAM, ebendort, Nr. 31, 1983
/3/ AIM Spezial (11), ebendort, Nr. 22, 1981
/4/ REMON, ebendort, Nr. 28, 1982
/5/ AIM Spezial (3), ebendort, Nr. 8, 1979
/6/ AIM Spezial (10), ebendort, Nr. 21, 1981
/7/ EXEDIT, ebendort, Nr. 31, 1983
/8/ SCREEN, ebendort, Nr. 30, 1983
/9/ ROFF-Textformatierer, ebendort, Nr. 33, 1983

[□]

AIM Spezial (16)

Der Herausgeber erhielt dankenswerterweise folgende Hinweise aus der Leserschaft:

Herr Peter Müller aus Düsseldorf beantwortet die im 'AIM Spezial (15)' gestellte Frage, warum der formatierende Assembler Blanks verschluckt, wie folgt: Die Ursache für die 'verschluckten Blanks zwischen Hexcode und Befehl ist ein Fehler in der Assembler-Routine, die diese Blanks erzeugt.

Dort wird am Ende der Schleife der Accu nicht erneut mit $20 geladen, so daß der Accu-Inhalt am Ende der U-Routine auf den Stack gelegt wird. War dieses ein nicht druckbares Zeichen, so entsteht der beschriebene Effekt. Beim Benutzen des formatierenden Assemblers ist daher darauf zu achten, daß die U-Routine für die Ausgabe des Listings den Accu-Inhalt rettet.

Herr Stephan Leicht aus Groß-Gerau weist auf eine Ungereimtheit im AIM-Monitor hin: Es betrifft die Routinen PHXY und PLXY in den Zeilen 1791 und 1803 des Monitor-Listings. Hier wird der Akku in CPIY+3 zwischengespeichert. In Interrupt-Routinen sollte man daher den Aufruf von PHXY und PLXY vermeiden, da an dieser Stelle ein Verändern des 'gesicherten' Akku-Inhaltes möglich ist.

Aus der Branche

Rockwell International: Die jetzt in serienmäßiger Auslieferung befindliche CMOS-CPU R65C02 soll lt. Herstellerangabe am Pin SYNC mit 20-100 pF gegen GND beschaltet werden.

Das Board RM65-5302E kann bis zu 16 unipolare oder 8 Differenz-Analog-Eingangsspannungen verarbeiten. Die A/D-Umsetzung kann auf eine Meßfehlergenauigkeit von 12 bit kalibriert werden.

Das R6500 Software Preparation System (SPS) ist ein Upgrade-Kit für den AIM 65, auch mit RM-Modulen für CRTC und Prom-Programmer betrieben. Es soll ab Jahresende lieferbar sein und etwa $99 kosten. - In der Summe ist das SPS eine Verbesserung und Erweiterung des Assemblers für die CMOS-CPU und alle 65xx CPU's mit einem Teilbeffehlssatz derselben sowie geänderter Monitor und Editor für den AIM. Zum Kit gehören 4 Festwertspeicher von je 4 KB sowie eine CPU R65C02. Offensichtlich werden zwei weitere Festwertspeicher wahlweise für CRT-Controller und Prommer angeboten. Ferner gibt es ein User's Manual (Bestell-Nr. 2167), das das AIM-Anwenderhandbuch auf den neuesten Leistungsstand bringt. - Der mögliche Anwender sollte sich vor Augen halten, daß das Kit eine neue Speicherbelegung bewirkt, denn der Assembler nebst mnemonischer Befehlseingabe für den erweiterten Befehlssatz liegt jetzt in den Adressen von $B000-CFFF, so daß hier nicht zugleich auch eine Sprache gefahren werden kann. Das Kit bedeutet damit vor allem eine Verbesserung der Assembler-Programmierung mit Merkmalen (und auch weiteren), wie sie in dieser Zeitschrift in den letzten Jahren an verschiedener Stelle hergestellt wurden. Wir haben insbesondere: Der gesamte Befehlssatz kann assembliert werden (auch im I-Kommando) und disassembliert werden. Der Assembler erlaubt jetzt eine Ausgabe in Zeilenbreite 80 Zeichen und hat auch einen Centronics Interface-Treiber für parallele Druckerausgabe. Es kann eine Symbol- und eine Cross-Referenz-Tafel erzeugt werden. Es kann mit 2 Textspeichern gearbeitet werden. Der anwenderbestimmte Input/Output (User) ist auf jeweils 10 Möglichkeiten gespreizt, ebenso die Anwendung der F3-Taste. Für diese liegen 30 Sprungbefehle im RAM ab Adresse hex 560, wo sie ggfs. vom Anwender für seine Zwecke nachgerichtet werden können. - In der Summe verspricht das SPS die maschinensprachliche Programmentwicklung und -erprobung mit mehr Komfort zu versehen, wobei es sicher den Vorteil hat, von einem Autor im Zusammenhang entwickelt worden zu sein.

Auf der SYSTEMS '83 zeigte Rockwell sein neues **Rockwell Design Center (RDC),** das aus drei Teilen besteht: Einschubgehäuse für Module, Zentraleinheit mit Bildschirm und Floppy Disk und schließlich absetzbare Tastatur. Die Grundversion hat 64 KB RAM, 2 Laufwerke DD/DS mit 1,3 MB und ein voll graphikfähiges Terminal. Das RDC kann bis zu 4 Zielprozessoren gleichzeitig unter Echtzeitbedingungen emulieren und hat 4 asynchrone Target-Busse, die vom Systembus getrennt sind. Es kann stand alone oder als Satellit an einem Host betrieben werden, um dessen Betriebssystem, Editor oder Massenspeicher zu nutzen.

Rockwell macht auch auf eine Reihe neuer Modems und Bausteine aufmerksam: R1212xx, R2424xx mit Mikroprozessorinterface für die voll duplexe Datenübertragung für alle amerikanischen und Europäischen Normen (zur einfachen Ansteuerung von alphanumerischen und graphischen Terminals, intelligenten Arbeitsplätzen und Heimcomputern), - Der Rockwell Bauelementesatz T-1 erlaubt die Datenfernübertragung in 24 Kanälen über eine einzige Zweidrahtleitung. Es werden alle notwendigen Kanal- und Rahmensignale übertragen. Damit ist der Aufbau lokaler Netzwerke mit geringen Verkabelungskosten möglich.

Ein High Resolution (512x512 Pixels) Graphic Controller wurde für AIM 65 und RM-65-Systeme wurde auf dem Rockwell-Stand durch die Mikrodatentechnik Dipl.-Phys. Peter W. Rott vorgestellt (Info: Wernhüterstr. 7, 89 Augsburg 1, T.: 0821-71 29 23). Es gibt verschiedene Module mit VG-Leiste für den RM 65bus mit Betriebssoftware: GC 65 ist monochrom, GM65 ermöglicht bis zu 256 Farben, GP 65 für Lichtgriffel, GRAF 65 ist ein tastaturorientiertes Graphik-Betriebssystem, GEDIT ein entspr. Editor, GPLOT 65 ermöglichst Hardcopies vom Bildschirm. Im Farbsystem können bis zu 8 memory planes angesprochen werden, die individuelle Masken haben können. Das Graphik-System ist sowohl unter Assembler-Programmierung wie in den Sprachen ansprechbar. - Zum System gehört eine 38-seite Dokumentation.

Valco/Signetics und der 68000: Der Herausgeber erhielt vor einigen Tagen eine ganze Reihe Datenblätter vom Zweitlieferanten Valvo: Für die CPU SCN68000 für 4, 6, 8 und 10 MHz. Dieses Datenblatt von Mai 1983 dürfte mit anderen bekannten gleichlautend sein. Ebenfalls vom Mai datiert die Unterlage für den SCN68230 Parallel Interface/Timer (PI/T), der in Heft 30 vorgestellt wurde. Ein Voraus-Datenblatt (noch ohne pin-out) beschreibt den SCN68454 Intelligent Multiple Disk Controller (IMDC). Er ist buskompatibel und unterstützt bis zu 4 Festplatten/Floppies in 8/16 Bit Datenübertragungen, ebenso Winchester-Interfaces SA1000 ST506. Datenraten 2 bis 10Mbits/Sek. für FM bzw. MFM. - Das Datenblatt für den SCB68430 Direct Memory Access Interface (DMAI) datiert von Mai 83. Er arbeitet im Cycle Steal- und im Burst-Modus (bis 4 MB). Operanden mögen 8/16/32 Bit breit sein. Er hat einen 24 Bit Adreßzähler und einen 16 Bit Transferzähler. - Beschrieben sind ferner der SCN68681 Dual Asynchronous Receiver/Transmitter (DUART), der SCN68652 Multi-Protocol Communications Controller, der 68661 Enhanced Programmable Communications Interface, der SCB68459 Disk Phase Locked Loop (DPLL) und der SCN68653 Polynomial Generator Checker (PGC).

Als VMEbus-Karten sind die folgenden beschrieben: System Controller SMVME1500, CPU-Modul SMVME2000, 256K Memory Modul SMVME3100, Disk Controller-Modul SMVME4300.

6502- und 6809-Karten und Systeme von hid, Nürnberg: Mit 'VauPorta' wird ein speziell für den professionellen Anwender im Industrie- und Ingenieurbereich geschaffenes portables Mikrocomputersystem bezeichnet. In der Frontwand hat es einen 9" Monitor mit 22 MHz Bandbreite. Die Tastatur ist klappbar angebracht. Es können ferner bis zu 3 Floppy-Disk-Laufwerke, 2 Mini-DCR oder andere Peripherieeinheiten eingebaut werden. Auf der Rückseite befinden sich ein Einschubrahmen mit 3 HE/ 84 TE.und eine HE darunter für Peripheriestecker. Als EuroHID 65 ist es ein zum AIM vollkommen kompatibles Baugruppensystem. Als VauSYS baut es auf dem 6809 auf und kann mit FLEX und OS9 betrieben werden. Es steht für den Aufbau eine große Reihe von Modulkarten zur Verfügung. Info: hid, Dipl.-Ing. B. Heckl, Alte Wallensteinstr. 146 in 8500 Nürnberg 80, T. 0911/65 17 47.

Commodore: Anläßlich der Internationalen Commodore-Fachausstellung (CFA) in Frankfurt und anläßlich der SYSTEMS '83 in München legte die Firma umfangreiche Informationen über ihre Geschäftserfolge vor, die hier nicht im einzelnen zitiert werden sollen. Der 42-jährige bisherige Geschäftsführer Harald Speyer wurde Vice President der amerikanischen Muttergesellschaft und European Manager mit vergrößertem Verantwortungsbereich. In der frankfurter Geschäftsführung folgt ihm der bisherige Vertriebschef Alwin Stumpf (36).

Die genannte 3. CFA-Ausstellung wurde von 80 Ausstellern beschickt und von 14000 Besuchern frequentiert. Während es vor einem Jahr noch etwa 90% Erwachsene waren, so belief sich heuer der Anteil der Kinder und Jugendlichen bereits auf etwa 50%, ein Zeichen dafür, wie sehr sich die Jugend für den Computer interessiert.

Der SX 64 als portables Modell des erfolgreichen VC 64 wird ab sofort in größeren Stückzahlen ausgeliefert. Ein Steckmodul und Tastenkappen rüsten ihn und alle anderen Modelle dieser Reihe auf den deutschen Zeichensatz um.

Entgegen im Herbst umlaufenden Gerüchten in der Branche werden die sehr erfolgreichen 8000er-Modelle und die der Serien 600 und 700 voll weiter produziert, wie dem Berichterstatter nach gezielter Befragung persönlich bestätigt wurde. - Der neue Software-Katalog von Commodore wird mehr als 500 Seiten umfassen. - Zum 64 werden folgende Softwarepakete besonders erwähnt: Simon's BASIC mit mehr als 100 zusätzlichen komfortablen Befehlen, Multidata (Datenverwaltung)und Calc-Result. - Die 8-Zoll-Laufwerke 8280 von Commodore können jetzt per Programm

Disketten im IBM-Format verarbeiten. - Neue Rechner sollen zu Beginn des nächsten Jahres angekündigt werden, welche die bisherigen Hardware-Linien nach oben wie auch nach unten abrunden. - Im Augenblick sieht die Firma ihr Wachstum durch Kapazitätsengpässe beengt. Der Handel klagt bereits seit Wochen über zunehmende Lieferzeiten.

Interessante Marktdaten der IDC Deutschland GmbH, Wiesbaden, enthielt die von Commodore im Oktober verbreitete Pressemappe. Die Frage 'Wofür werden Microcomputer benutzt?' wurde (gerundet) wie folgt beantwortet: Speichern von Informationen 75%, Selbst programmieren 63%, Spielen 58%, Texte bearbeiten 50%, Grafiken und Tabellen erstellen 50%, Haushaltsbudgets und Steuern verwalten 44%, Lernprogramme für die Familie 37%, Zugriff auf andere Computerdateien 32%, Börsenkurse/Aktienhandel 17%, Nachrichten und elektronische Briefe weiterleiten 10%. Diese Kategorien weisen darauf hin, daß es sich um amerikanische Daten handeln könnte. Der Markterfolg wird weltweit für Mikrocomputer wie folgt aufgeteilt. Insges. 5,763 Mio.Systeme abgesetzt, davon 2,746 für Home/hobby, 2,225 für Business/Professional, 0,320 Mio Technisch/wissentschaftlich, 0,472 für Erziehung.

Der Mikromarkt Deutschland wird für das Jahr 1982 wie folgt beziffert: 223.000 Auslieferungen, davon 71% an Haushalte, 29% an Betriebe, Schulen und Freiberufler. Die Marktsättigung wird beschrieben mit o,8% bei Haushalten mit mehr als 32.500 DM Einkommen, 1,7% bei Kleinbetrieben und Freiberuflern mit weniger als 10 Mitarbeitern, mit 11% in Mittelbetrieben mit 10-200 Mitarbeitern. 15.000 Großbetriebe mit mehr als 200 Mitarbeitern sollen 33.000 Mikrocomputer beschafft haben.

Marktanteile heute in Deutschland: Commodore 26%, Apple 11%, HP 10%, Tandy 7%, Kontron 6%, Triumph-Adler 5%, andere 35%. Wenn diese Zahlen in etwa zutreffen, dann hat die IBM hier noch nicht tief Luft geholt.

Kleinanzeigen

AIM 65, 16K RAM, BASIC, Assembler, mit HERO-System Video 31/64, Mini-DCR, 16/32K Epromkarte, Eprommer, VB 1300 DM. Tel.: 04151 - 53 64.

FORCE SYS68K/CPU-1 mit MC68000, 128 kbyte RAM, 3 x RS232C, Parallelschnittstelle, Uhr, VMEbus, beschrieben in Heft Nr. 30 MICRO MAG, incl. Editor/Assembler; 15% unter Neupreis. Rott, Tel.: 0821 - 71 29 23.

AIM 65 zu verkaufen: Expansionserweiterung mit Busadaption, 44 KB RAM, Videoplatine, Echtzeituhr, Floppycontroller + DOS und Doppelfloppystation (bestückt mit 1*SA400), Netzteil, 19"-Einschub. Erweiterung für Applikationsbus-- EPROMMER, VIA-LED, Relaisplatine, RS-232 Druckerinterface. Software: REMON, FAST-ASSEMBLER, Editorerweiterung + SCREEN-Editor, Textformatierer, alle Sprachen. Bestimmte Baugruppen auch einzeln zu verkaufen: VB 3100,- DM. Dipl.-Phys. A. Redder, c/o Schild, Goldstr, 16, 4400 Münster, Tel.: 0251/83 49 59 (tagsüber).

Textverarbeitung mit AIM 65: Groß- und Kleinschreibung umschaltbar mit CNTL A, deutsche Tastaturbelegung, automatisches Tastenrepeat, Schnelles REPLACE, Delete über DILINK, Treibersoftware für serielle Druckerschnittstelle, Editorerweiterungen einfach über Vektoren u.v.a.m.. Zwei geänderte Monitor-EPROMs REMON 3.5 für 98,- DM inkl. ausführlicher Dokumentation.

Textformatierer ROFF 3.2 für jeden AIM 65. 19 Formatierkommandos für Blocksatz, Seitenumbruch, automatische Seitennumerierung, Zentrierung u.v.a.m.. Software im 2532-EPROM für Steckplatz $C000 auf der AIM-Hauptplatine für 98,- DM inkl. ausführlicher Dokumentation.

Paketangebot: REMON 3.5. + ROFF 3.2 + EXEDIT 5.3. Erweiterter Editor mit Blockmove, -copy, -erase, Change to End, Help-Funktion, zweiter Editor aktiv u.a.. Drei 2532 EPROMs in den C, E und F-Sockeln der AIM-Hauptplatine für komplett 178,- DM inkl. ausführl. Dokumentation. **Bestellungen bei Dipl.-Ing Rüdiger Wollenberg, Stockumer Str. 234, 4600 Dortmund 50, Tel. 0231 - 75 34 77.**

Assembler unter FORTH

CROSS–09

Cross-Assembler für MC6809 (Motorola). Tonbandcassette (6,5K), Diskette (CBM 3040/4040) oder 2 EPROMs, deren Inhalt nach hex 200-1B00 kopiert werden muß. DM 380,-- + MWSt.

CROSS–05

Cross-Assembler für MC6805/68705xx (Motorola), auf Cassette, Diskette oder EPROM für D000: DM 320,-- + MWSt.

Alle Assembler sind komfortabel ausgelegt, benutzen die Mnemonics des Herstellers und laufen unter FORTH des AIM 65 (für CBM-FORTH von Lowinski oder PHS ggfs. auf Anfrage). Sie erzeugen EPROM-fähigen Code für beliebige Speicherräume und enthalten alle üblichen Kontrollstrukturen wie BEGIN, WHILE, REPEAT, IF, ELSE, THEN usw. aber auch alle Branches (6809: Longbranches). Es kann mit externen Symbolen und Labels im Programmablauf gearbeitet werden. Zusätzlich sind viele Assembler-Anweisungen implementiert: Byte- Word- und Stringablage, .OPTLIS, Reserve Memory Bytes und binäre Argumente.

Mathe–ROM für 6502

Implementierung von P. Rix (s. Textteil) für Sockel D000. Fließkommaarithmetik und höhere mathem. Funktionen (wie in Microsoft-BASIC) für AIM 65-FORTH(komfortabler eigener Fließkommastapel) und für jedes 6502-Assemblerprogramm (dokumentierte Einsprungspunkte und Argumente). EPROM DM 110,- + MWSt.

Commodore Serie 700 und 600 ROM-Listing DM 74,--

Ca. 300 Seiten, gebunden, ausführliche Dokumentation mit original CBM-Labeln, Belegung der Zero Page, des RAM und des I/O, Cross-Referenzliste. Assemblierfähiger Quelltext. Die Benutzung der Banks/Segmente wird erklärt.

LASER 110

PERSONAL-COMPUTER

Prompt lieferbar: Computer-Konsole mit Microsoft-BASIC und Betriebssystem in 16 KB für Anschluß an TV-Gerät oder Daten-Monitor (TV- und Video-Ausgang), BASIC mit 6 signifikanten Ziffern und PRINT USING.

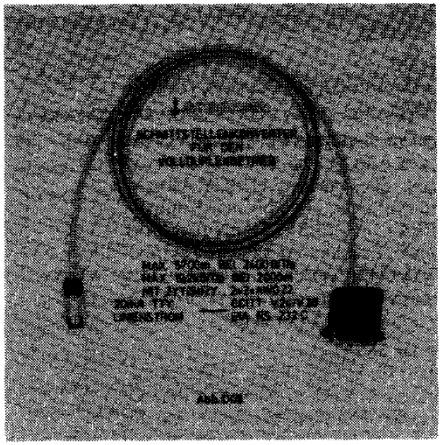
CPU Z80A, 4KB RAM. Siehe auch die Besprechung in Heft 32. Mit externem Netzteil, TV-Anschlußkabel, Kabel für Cassetten-Rekorder, Demo-Cassette. Deutsches Anwenderhandbuch.

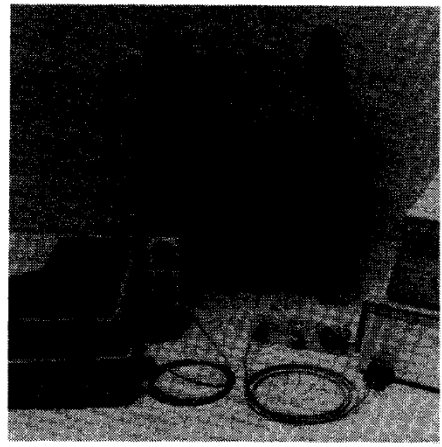
Preis inkl. MWSt, Verp. und Versand DM 215,--

Floppy-Schnittstelle

Umsetzung des parallelen IEEE 488 Busses auf die preiswerte CBM-Floppy 1541. Gestattet den Betrieb von CBM und PET mit der Floppy ebenso aller anderen Rechner mit IEEE-Bus Routinen (s. Heft 33). Einfacher Anschluß des Schnittstellenprozessors (28 Pin). Lieferbar ab Dezember 1983. DM 198,- inkl. MWSt und Zusendung.

Roland Löhr, Hansdorfer Str. 4, 2070 Ahrensburg
Tel.: 04 102 - 55 816





V.24/V.28 oder RS 232 C — 20 mA Konverter eingebaut in einem serienmäßigen Subminiatur „D" Steckergehäuse
Gebrauchsmuster angemeldet!

Haben Sie weit auseinanderliegende Datengeräte mit Standard V.24/V.28 oder RS 232 C Schnittstelle? ... dann ist 20 mA Current Loop die Lösung für Ihr lokales Netzwerk.

Den Konverter liefern wir:

Technische Daten:

— V.24/V.28 oder RS 232 C — 20 mA Current Loop.

— Galvanische Trennung

— Umschaltbar aktiv/passiv

— Betriebsanzeige: Senden, grüne LED
Empfangen, rote LED

— Maximale Leitungslänge 5700 Meter, Kabel 2YY (ST) 2Y2 x 2 x AWG22 (0,64); 58nF/km; 114 Ohm/km, 2,4 k Bit/sec sind dann noch möglich.

— Maximale Übertragungsrate bei gleichem Kabel: 19,2 k Bit/s bis 2000 m aktiv senden
und bis 290 m passiv senden.

— Einspeisung: +/— 12 V aus angeschlossenem Datengerät oder über Schalterdosennetzgerät SKNT 20, das zusätzlich in die Wandsteckdose der fest verlegten 20 mA Datenleitung eingebaut wird.

— Stromverbrauch: + 12 V ca. 60 mA max.
—12 V ca. 30 mA max.

— Abmessungen: 49 x 54 x 15 mm
AMP Subminiatur D Stecker 25polig, lieferbar mit Stiften oder Buchsen.

— Auf Wunsch mit flexibler Leitung 7 x 0,14 und Rundstecker Amphenol Tuchel 7polig.
Vier Adern Daten, drei für externe Stromversorgung.

Preise incl. MWSt

Konverter ohne Kabel ..... DM 342,00
Konverter mit 2,5 m
7 x 0,14 wie Abb. DO9: ..... DM 438,90
Netzgerät SKNT 20 für
Einbau in Installationsdose: ..... DM 171,00

Wir beraten und planen für Sie,
Ihr lokales Netzwerk!
Fordern Sie bitte weiteres Informationsmaterial an.

STECKER
INGENIEURBÜRO

5000 Köln 60 (Niehl)
Postfach 60 07 66
Delmenhorster Straße 20
Telefon (02 21) 7 12 40 18

Neue Telefon-Nummer

DUO Plott Interface

für MX80 F/T und
ITOH 8510
und COMMODORE-Rechner 30/40/80

Paralleles IEEE 488 - Interface, genormter IEEE-Stecker und Kabel
kompletter Zeichensatz des CBM-Computers
zwei Geräteadressen für Groß-/Grafikmodus und Textmodus
alle Funktionen der Drucker bleiben erhalten, Floppy-Kompatibilität
Deutsche Umlaute, ß und Paragraph
Problemloser Einbau, inklusive deutsches EPSON-Handbuch
Komplettpreis einschl. Kabel, CBM-Grafiksatz und Handbuch incl. MWSt
für EPSON DM 298,--
für ITOH DM 298,--

Umrüstsatz MX80 F/T auf MX80 Typ 3

Aus Ihrem EPSON MX-80 F/T wird der MX-80 Typ 3
Einzelnadelansteuerung, erweiterter Befehlssatz, Elite-Schrift
Preis inkl. MWSt DM 98,--
Komplettpreis Interface, Kabel, Handbuch und Umrüstsatz inkl. MWSt DM 450,--

Für COMMODORE

Deutsche Tastatur mit Original-Tastensätzen (keine Aufkleber)
inkl. deutschem Zeichengenerator
für CBM 8032 DM 98,--
für CBM 8032 und 8096 DM 98,--
nur 8096, ohne Tasten DM 98,--

Speichererweiterung auf 96 K

LOS-kompatibel, inkl. MWSt DM 798,--

ISAM-Routinen
Datenhaltungsprogramm, Indexed Sequential Access Method
siehe Besprechung in Heft 23 des 65xx MICRO MAG, **DM 298,--**

Drucker:

RX 80 DM 998,-- + MWSt

ITOH 8510 DM 1495,-- + MWSt

EPSON-Druck-Computer FX 80 DM 1295,-- + MWSt

Stellberg Computer-Systeme

COMMODORE EPSON C. ITOH SOFTWARE INTERFACE

Blindenaaf 36 5063 Overath Tel.: 022 06 - 66 44

65xx MICRO MAG

COMPUTING · SOFTWARE · HOBBY

Herausgeber:
Dipl.-Volkswirt Roland Löhr
Hansdorfer Straße 4
D–2070 Ahrensburg
Tel.: 04 102 - 55 816

65xx MICRO MAG erscheint zweimonatlich, jeweils Mitte Februar, April usw..COPYRIGHT 1983 by Roland Löhr. Alle Rechte vorbehalten, auch die des auszugsweisen Nachdruckes, der Übersetzung, der fotomechanischen Wiedergabe und die der Verbreitung auf magnetischen und sonstigen Trägern. Beiträge, die nicht besonders gekennzeichnet sind, stammen vom Herausgeber. - Offsetdruck: Druckartist Gerhard M. Meier, Hamburg 70.

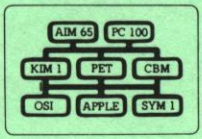
Bezugsbedingungen: Abonnement ab laufender Ausgabe für 6 Hefte DM 54,-- (Inlandsendpreis). Ausland/foreign via surface mail DM 59,--, USA air 26 Dollar. Abonnements laufen bis auf Widerruf mit Kündigungsmöglichkeit bis zu zwei Wochen vor deren Ablauf.

Nachliefermöglichkeiten: Hefte 1-6 und 7-13 sind als Buch nachlieferbar (s. Anzeige unten). Hefte 14-33 können alle als Einzelhefte zu DM 7,80/St. + DM 2,50 je Sendung geliefert werden. Einzelpreis bei 10 und mehr Heften je Sendung DM 6,--.

Private Besteller werden um Überweisung/Scheck (auch Auslandsschecks) zusammen mit der Bestellung gebeten. Konto Roland Löhr, Nr. 654 70-202 Postscheckamt Hamburg, BLZ 200 100 20.

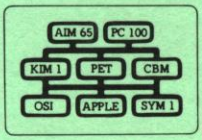
Leser-Service des Herausgebers

Das Buch 1-6 des 65.. MICRO MAG



230 Seiten, DM 26,--

Das Buch 7-13 des 65.. MICRO MAG



340 Seiten, DM 42,--

FORTH User's Manual
Rockwell-Handbuch für das FIG-FORTH des AIM 65. Mit der Erläuterung des Befehlssatzes auch für andere FORTH-Betreiber geeignet. Ca. 300 S., engl. DM 30,--

Mathe-ROM nach P. Rix für FORTH des AIM + Assemblerprog. m. Doku DM 124,30

Vorstehende Preise inkl. MWSt, zuzüglich DM 2,50/Sendung + ggfs. NN + DM 2,--

Cross-Assembler für Motorola 6809 DM 380,-- und für 6805 DM 320,-- + MWSt, inkl. Dokumentation (DM 10,--), beide unter FORTH des AIM laufend.