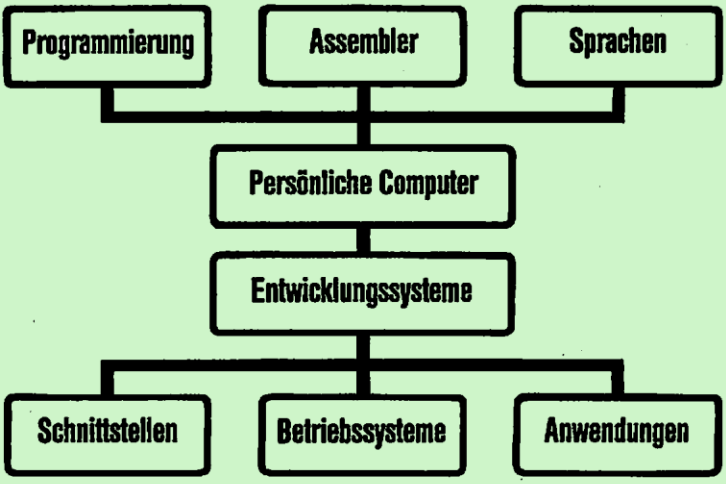
MICRO MAG

DM 9,50 Nr.45 November 1985



Inhaltsverzeichnis

Texteditor CBM 720 3
Commodore AMIGA 23
Atari ST260 und ST520+ 24
Multitech Popular 500, IBM-Kompatibel 25
MELPS 40, Einchipper mit 65xx-Befehlssatz 26
PL/65, Anpassung und Erweitertung 28
Eproms ab Typ 2764 43
Commodore PC-128 56
Bücher 59
Editorial 60
Aus der Branche 62

dBASE III: Einführung und Referenz
Die deutsche dBASE III-Version – an Hunderten von Beispielen illustriert, in Moduln lexikalisch dargestellt, mit schnell orientierenden Befehlserklärungen.
Ein Mehrzwecktext:
Als Einführung in kurserprobten Lernsequenzen lesbar.
Als Referenz in alphabetsicher Befehlsfolge lesbar.

Als dBASE III-Kurstext entwickelt, nutzt dieses Buch die beiden Anwendungsvarianten von dBASE: über **Direktbefehle** und über **Befehlsfolgen in Programmform** benutzbar zu sein.
Ideal als Kursunterlage durch 2. Inhaltsverzeichnis (guided tour) mit Führung durch Lernmoduln und Erfolgsabfrage. Geschrieben für Erfordernisse der Praxis.
Russel Stultz, ca. 464 Seiten, Softcover, reiches Bild-/Beispielmaterial, DM 79,–

NEU
BASIC Programmierung: PC-10/PC-20
Ein amerikanisch-lockerer BASIC-Kurs von dem kalifornischen Professor Lien. Durch seine Systematik ideal als Kursunterlage für PC-10/PC-20 und Kompatible. Mit Einführung in das PC-10-System und Tastendarstellung im Text; BASIC-Befehlsschreibung mit Aufgaben und Antworten; Sammlung von Beispielprogrammen; BASIC-Befehle für Ton/Graphik/Text; SAVE/MERGE/CHAIN-Anwendungen. Ein leicht lesbares Buch für Erstanwender, systematisch und vollständig für Erfahrene.
David A. Lien, 488 Seiten, Softcover, DM 59,–

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40
te-wi

Roland Löhr

Texteditor CBM 720

Für die Commodore-Rechner 710/720 wurde bisher nur wenig Software veröffentlicht. Das Modell fand trotz guter Leistungsmerkmale nur wenig Verbreitung und wird inzwischen auch nicht mehr produziert. Aus den noch vorhandenen Lagerbeständen kann man derzeit sehr günstig einkaufen.

Der Computer wurde in Heft 32 beschrieben: Die CPU 6509 arbeitet mit 2 MHz Takt, der Computer hat eine professionell derbe Tastatur mit Funktionstasten und Zehnerblock, deutsche Zeichen auf der Tastatur und auf demsehr klaren Schirmbild. Dem erweiterten BASIC 4 und dem Anwender stehen neben der Systembank 15 mit Kernel, BASIC und I/O vier Bänke mit jeweils 64 KB für BASIC-Text und Variablen zur Verfügung. Wir haben den üblichen IEEE 488-Anschluß (parallel) für Geräte (Floppy, Drucker), einen Userport und eine RS 232-Schnittstelle mit bis zu 9600 Baud. Mit dieser Ausstattung ist dieses Gerät nach wie vor attraktiv und leistungsfähig, insbesondere, wenn man ein hervorragendes Textsystem wie GMA-Text oder die Datenbank Superbase darauf betreibt. - In vielen Fällen kann man den Rechner auch als intelligentes Terminal einsetzen.

Der nachstehende Texteditor wurde für dieses System (720) entworfen, um ein Textprogramm mittlerer Leistungsfähigkeit stets auf der Maschine resident zu haben. Es wird also nicht umständlich geladen, sondern es ist auf einer Erweiterungsplatine in einem EPROM ab Adresse hex 4003 enthalten. Gleichwohl kann es auch in anderen Bänken zur Ausführung gebracht werden, wenn man die Datenbänke konsequent disponiert und wenn man die in Heft 33 veröffentlichten Banking-Routinen benutzt. - Die mit dem Programm geschaffene Umgebung ist speziell für Assemblierungen bequem. Bei solchen Arbeiten möchte man große Quelltexte (hier bis 128 KB) oder Kommandofiles auf der Maschine resident halten, Textstellen finden und Abänderungen vornehmen können. Solche sind typisch Berichtigungen in einer Zeile, Einfügen oder Auslassen von Zeilen. Beim Lesen möchte man Zugriff wahlweise auf die Floppy oder die Tastatur haben..

Die Benutzeroberfläche ähnelt der des Texteditors im AIM 65. Auch die Labels haben oft eine ähnliche Bezeichnung, gleichwohl ist das Programm davon unabhängig. Es wurden mehr und leistungsfähigere Funktionen eingebaut. Insbesondere kann man während der Texteingabe oder während des Veränderns einer Zeile in dieser korrigieren (Cursor-Tasten, Insert und Delete). Eine Zeile wird erst mit 'CR' aus dem Zeilenpuffer INBUFF in den eigentlichen Textspeicher übernommen.

Während der Texteingabe/Korrektur müssen alle Tasten totgelegt werden, die ein Entweichen des Cursors in eine andere Zeile oder sonst Störungen bewirken würden. Natürlich könnte man mit Cursor Up oder Down in die Vorzeile springen oder in die folgende. Das würde logisch aber die Verwaltung von Screens erfordern, was hier nicht beabsichtigt war. — Gleichwohl kann dieser Editor für vergleichbare Umgebungen angepaßt werden, wo Zeichen z.B. von einem Terminal her über eine ACIA seriell auf irgendeine Computerplatine als Text eingegeben werden. Da CBM-Systeme inzwischen eine vereinheitlichte Sprungleiste am Ende des Festwertspeichers in den hohen Adressenhaben, sollte die Anpassung auch an andere CBM-Computer recht leicht möglich sein.

In dieser Zeit schnell vorbeiziehender Computermodelle ist die Darlegung der Programm-Logik wichtiger als die Fixierung auf ein einzelnes Gerät. Zur Erklärung gleichwohl das Folgende: Die CPU hat in den sonst als Zero Page benutzten Zellen 00 und 01 zwei Register für die gültige Programm- und die Datenbank. Letzteres Register wird i6509 genannt. Sie hat über entsprechend ausgesandte Signale an den Pins Zugriff auf 16 Programm- und Datenbänke. Letztere können im 'Durchgriff' über das Register i6509 nur mit den Befehlen LDA (POINTER),y und STA (POINTER),y erreicht werden, nicht jedoch, was immer einmal wieder zu fehlerhafter Programmierung führt, mit den übrigen arithmetischen oder logischen Befehlen, wie z.B. CMP (POINTER),y. Dies muß man halt hinnehmen. Maximal hat man dann jedoch auf dem CBM 720 4 Datenbänke mit den Nummern 1,2,3 und 4.

Etliche Routinen des Programms dienen daher dem Durchgriff durch die Datenbanken, wobei der Eingabe- und Korrekturpuffer INBUFF in Bank 15 (hex 0F) gehalten wird. Dieses Programm erlaubt es, zwei Bänke als Textbänke zu benutzen, nämlich 2 und 1, so daß man 128 KB Text in gedrängter Form (ohne redundante Spaces an den Zeilenenden) auf der Maschine resident halten kann. Die Bankumschaltung erfolgt mit dem Befehl '/' (Slash). In der Routine BANK-SWIT könnte man natürlich auch eine dritte oder vierte Bank unter Bewahrung der jeweiligen Parameter zulassen.

Beim Assemblieren entsprechen 128 KB kommentierten Quelltextes etwa 12 KB Maschinencode. Dieser wird hier ersatzweise (on default) in Bank 3 an beliebiger Stelle abgelegt oder mit Offsetvorgabe an beliebiger Stelle in der Maschine. Beim Assembler dient dann Bank 4 der Aufnahme der Symboltafel mit maximal etwa 4600 Symbolen. Insgesamt hat man dann auf dem CBM 720 als Entwicklungssystem viele Freiheiten.

Im Laufe der Zeit sind dem Autor viele Textsysteme und Editoren über den Tisch gekommen. Für CBM-Maschinen waren es auch solche, die das Aufzeichnungsformat für BASIC-Texte nebst Zeilennummern benutzten oder auch solche im WORDPRO-Format. Manche benutzen für die Aufzeichnung auf der Diskette das Format PGM oder USR. Das ist unglücklich, und man darf das heute nicht mehr machen, denn es kommt heute auf Kompatibiltität und den Austausch rein sequentieller Files von Rechner zu Rechner an. Zu fordern sind also reine ASCII-Dateien, in denen eine Zeile mit dem ersten Buchstaben beginnt und mit einem 'CR' abgeschlossen wird, der gesamte Text wird mit 00 oder ggfs. CTRL-Z begrenzt. Und: Commodore hat eine eigene interne Verschlüsselung der Zeichen, die von ASCII abweicht. Hier sollte man von den auf dem CBM 720 vorhandenen Umwandlungsroutinen TOASCI (Senden) und TOCBM (Empfang) Gebrauch machen.

Im Textsystem sind die folgenden Befehle und Funktionen implementiert, der Anwender kann sie nach seinen Wünschen erweitern:

Taste	Routine	Funktion
E	TEXT–EIN	Laufende Texteingabe an das Ende des Textspeichers
T	TOP	Einstellen der obersten Zeile, Textbeginn
B	BOTTOM	Einstellen hinter die letzte Zeile
U	UP	1 Zeile aufwärts gehen
D	DOWN	1 Zeile weitergehen
'CR'	RETURN	Anzeige der laufenden Zeile
Q	QUIT	Rückkehr zu BASIC, wenn mit SYS aufgerufen
K	KILL	1 Zeile entfernen
I	NSERT	1 Zeile zwischenschießen
R	READ	Von Floppy an das Textende hinzulesen
L	LIST	Listen der Zeilen auf den Bildschirm, Anhalten mit der C=-Taste, Zeilen-Nummer mit # anzeigen, Listung verlassen mit Q
F	FIND	Sehr schnelles Finden eines Strings mit dem von M. Albrecht in Heft 42 veröffentlichten Algorithmus
W	WEIT–SUCH	Fortsetzung der FIND-Funktion, Wiederholung mit der Stringvorgabe
C	CHANGE	Zeile wird wieder zur Korrektur geöffnet
A	ASSEMLER	Sprung auf ein Assemblerprogramm
M	MONITOR	Sprung in das Monitorprogramm des CBM 720
/	BANKSWIT	Bankumschaltung, Nummer ist anzugeben
S	LOESCH	Gesamten Text in einer Bank löschen
N	NUMMER	Ansteuern einer Zeile mit Nummer (vierstellig eingeben!) ab Textbeginn zu zählen, virtuelle Zeilennummer
P	PRINT	Ausgabe auf Drucker, Device 4
O	READNSERT	Lesen eines Files von Floppy vor die Einfügestelle
<	LIN–NSERT	Fortlaufendes Einlesen mehrerer Zeilen von der Tastatur, Einfügung vor Einfügestelle. Abbruch wie bei 'E' mit ESC und dann §
&	WRIT–FLOP	Schreiben des gesamten Textes auf Floppy, Device 8

Zum Programmablauf folgende Übersicht. Die ersten Befehle dienen der Initialisierung des Cursors und der Zeiger auf die Textbänke. EDITOR zeigt in einer Zeile die möglichen Befehle und die Nummer der Textbank an. BEFPAR entschlüsselt die Kommandotasten, mit denen auf eine der vorstehenden Routinen verzweigt wird, die im ersten Teil des Programmes stehen. Daran schließen sich viele der wiederholt benutzten Unterprogramme an und schließlich in Tabellen die Meldungen sowie die verschlüsselten Befehlstasten und ihre Sprungadressen. Bemerkeswert ist, daß die schon zitierte FIND-Routine einen am Ende eines 64K-Textes befindlichen String im Bruchteil einer Sekunde findet.

```
0001                                ;SKO"EDITORSA"
0002                                ;FUER CBM 720, COPYRIGHT ROLAND LOEHR, 10/85
0003                                ;BEFEHLSTASTEN SIND:
0004                                ;E=TEXTEINGABE TASTATUR
0005                                ;T=TOPZEILE EINSTELLEN
0006                                ;B=BOTTOM, HINTER LETZTE ZEILE GEHEN
0007                                ;U=UP, 1 ZEILE AUFWAERTS GEHEN
0008                                ;D=DOWN, 1 ZL. ABWAERTS GEHEN
0009                                ;CR=LFD. ZEILE ANZEIGEN
0010                                ;Q=QUIT, ZU BASIC ZURUECK
0011                                ;K=KILL, 1 ZL. ENTFERNEN
0012                                ;I=INSERT, 1 ZL. ZWISCHENSCHIEBEN
0013                                ;<=LAUFENDER INSERT MEHRERER ZEILEN
0014                                ;R=READ V. FLOPPY HINTER VORH. TEXT, DEVICE 8
0015                                ;O=READ VON FLOPPY MITTEN IN DEN VORH. TEXT
0016                                ;L=LIST, UNTERBRECHBAR MIT C= UND #=ZL-NR
0017                                ;F=FIND, INSTRING-FUKTION, STRINGSUCHE
0018                                ;W=WEITER SUCHEN, FORTSETZUNG VON FIND
0019                                ;C=CHANGE, ZEILE ZUM NEUEN EDITIEREN AUFMACHEN
0020                                ;A=ASSEMBLER AUFRUFEN
0021                                ;M=SPRUNG ZUM MONITOR
0022                                ;/=BANKWECHSEL
0023                                ;S=SCRATCH ALL TEXT IN THIS BANK
0024                                ;N=GEHE AUF ZEILEN-NR., 4 ZIFFERN !!! EINGEBEN
0025                                ;P=PRINT AUF DEVICE 4
0026                                ;&=SEQU. FILE AUF FLOPPY SCHREIBEN, DEVICE 8
0027                                ;
0028                                ;********** HARDWARE-UMGEBUNG ***
0029 000000         I6509     =$01              ;DATENBANK-REGISTER
0030 000000         CRTC      =$DB00            ;B5-CONTROLLER 6545
0031                                ;VERKEHRSZEICHEN ********
0032 000000         NULL      =0
0033 000000         CR        =$0D              ;CARRIAGE RETURN
0034                                ;***********************
0035 000000         ESCAPE    =$1B              ;ESCAPE-ZEICHEN
0036 000000         DELETE    =$14
0037 000000         INSERT    =$94              ;TASTENZEICHEN
0038 000000         SPACE     =$20
0039 000000         REVERS_ON =$12              ;REVERSE SCHRIFT EIN
0040 000000         CURSOR_UP =$91              ;CURSOR UP CHARACTER
0041 000000         REVRS_OFF =$92              ;BZW. AUS
0042
0043                                ;********** SYSTEMVARIABLEN CBM 720
0044 000000         CR_POS    =$D5              ;POSITION LETZTES ZEICHEN IN
                                    DER ZEILE
0045 000000         TXTPTR    =$85              ;=BUFFPT FUER UEBERGABE AN BA
                                    SIC
0046 000000         BUFFPT    =$88              ;ZEIGER NEBST BANK AUF EINGAB
                                    EPUFFER TASTATUR
0047 000000         INDEX1    =$22              ;ZEIGER AUF EINGABEPUFFER
0048 000000         COUNT     =$0E              ;ZEILENLAENGE NACH JSR GETSTR
                                    ING
0049 000000         TMP1      =$B9              ;SPEICHER F. BIN ZAHLENEINGAB
                                    EN
0050 000000         PNTR      =$CB              ;CURSORSPALTE
0051 000000         TBLX      =$CA              ;CURSORZEILE
0052 000000         QTSW      =$D2              ;QUOTE-FLAG, ANFUEHRUNGSZEICH
                                    EN
0053 000000         INSFLG    =$039A            ;INSERT-FLAG ESC/A/C
0054 000000         INDX      =$D5              ;POSITION LETZTES ZEICHEN IN
                                    ZEILE
```

```
0055 000000          LSTCHR   =$0260          ;LAST CHARACTER ENTERED, FUER
                                  'ESC'
0056 000000          RVSFLG   =$0397          ;REVERS-FLAG
0057 000000          PAGSAV   =$03A0          ;ZEITWEISE RETTUNG EINER BANK
0058 000000          LA       =$9E            ;AKTIVES LOG. FILE
0059 000000          SA       =$A0            ;AKTIVE SEKUNDARADRESSE
0060 000000          DFLTO    =$A2            ;AKTIVE AUSGABE-GERAETE-NR.
0061 000000          FA       =$9F            ;AKTIVE GERAETE-NUMMER
0062 000000          FNLEN    =$9D            ;LAENGE FILE-NAME
0063 000000          FNADR    =$90            ;POINTER AUF DEN FILE-NAME
0064 000000          STATUS   =$9C            ;I/O-STATUS
0065
0066                          ;******** VARIABLEN DES BENUTZERS, ZERO-PAGE
0067                          ;ABGESTIMMT MIT 65816-ASSEMBLER
0068 000000          NOWLN    =$F0            ;POINTER AUF AKTUELLE ZEILE I
                                  M EDITOR-BUFFER
0069 000000          BOTLN    =$F2            ;POINTER AUF ENDE DES EDITOR-
                                  BUFFERS
0070                          ;
0071 000000          MOVPTR   =$F8            ;POINTER FUER BLOCKMOVE
0072 000000          SYMDES   =MOVPTR
0073                          ;VARIABLEN DES ANWENDERS
0074 000000          INBUFF   =$400           ;ALLG. EINGABEPUFFER
0075 000000          FINDBUFF =$500           ;SUCHSTRING AUFBEWAHREN
0076 000000          SUCH_TAB =$580           ;TABELLE BEIM STRINGSUCHEN
0077
0078 000000                   *=$600
0079 000600          ZL_LEN   *=*+1           ;LAENGE EINGABEZEILE
0080 000601          OLD_LEN  *=*+1           ;LAENGE NACH JSR DISLIN
0081 000602          SUCH_LEN *=*+1           ;LAENGE SUCHSTRING AKTUELLE S
                                  UCHE
0082 000603          SUCH_LENW *=*+1          ;VORHALTUNG BEIM WEITERSUCHEN
0083 000604          SUCH_PNTR *=*+2          ;EINSETZSTELLE BEIM FORTGESET
                                  ZTEN SUCHEN
0084 000606          SAV_LEN  *=*+1           ;ALTE LAENGE GESICHERT
0085 000607          END      *=*+2           ;POINTER BEIM STRINGSUCHEN
0086 000609          SYMSAV   *=*+2           ;TEMP ABLAGE
0087 00060B          SYMPAG   *=*+1           ;ZAHL ZU MOVENDER PAGES
0088 00060C          TEMP     *=*+1           ;TEMPORARY STORAGE
0089 00060D          TEXTBANK *=*+1           ;AKTIVE TEXTBANK
0090 00060E          MODE     *=*+1           ;00=NORMALE TEXTEINGABE, $FF=
                                  CHANGE-MODE
0091 00060F          NOWLN1   *=*+2           ;POINTER-RETTUNG FUER BANKUMS
                                  CHALTUNG
0092 000611          BOTLN1   *=*+2           ;DITO
0093 000613          NOWLN2   *=*+2           ;POINTER-RETTUNG FUER BANKUMS
                                  CHALTUNG
0094 000615          BOTLN2   *=*+2           ;DITO
0095 000617          NFELD    *=*+2           ;FELD F. KUENSTL. ZEILEN-NUMM
                                  ERN
0096 000619          INS_FLAG *=*+1           ;00=APPEND-, $FF=INSERT-MODUS
0097 00061A          FLOP_VEKT *=*+2          ;INDIREKT VEKTOR, FLOPPY AUSG
                                  ABE
0098
0099                          ;******** KONSTANTEN DES BENUTZERS
0100 00061C          ANFANG   =$0003          ;BEGINN DES EDITOR-BUFFERS IN
                                  BANK 1, 2
0101 00061C          BANK15   =15             ;BANK DES ZEILEN-PUFFERS
0102 00061C          BANK2    =2              ;BANK DES HAUPT-TEXTSPEICHERS
0103
0104                          ;******** SYSTEMROUTINEN ****
0105 00061C          KSTOP    =$FFE1          ;RUN/STOP-TASTE LESEN
0106 00061C          KUNLSN   =$FFAE          ;UNLISTEN AUF IEEE AUSGEBEN
0107 00061C          KLISTN   =$FFB1          ;DITO LISTEN
0108 00061C          KSECND   =$FF93          ;SECONDAERADRESSE AUSG.
0109 00061C          KCKOUT   =$FFC9          ;AUSGABEKANAL OEFFNEN, LA IN
                                  X UEBERGEBEN
0110 00061C          KBSOUT   =$FFD2          ;ZEICHEN AUF AKTIVEN KANAL AU
                                  SGEBEN
0111 00061C          KC1OUT   =$FFA8          ;EIN BYTE AUS AKKU AUF IEEE-B
                                  US AUSG.
0112 00061C          ERROR4   =$F942          ;FILE NOT FOUND ERROR
0113 00061C          WRDA     =$0OF6          ;X, DANN Y ALS 4 ASCII-ZAHLEN
                                  AUSGEBEN
```

```
0114 00061C                RDOA      =$F123          ;4 ZAHLEN EINLESEN, ZU HEX MA
                                       CHEN NACH TEMP1
0115 00061C                FNDEND    =$E54B          ;SETZT CURSOR AUF LETZTES ZEI
                                       CHEN PER POINTER
0116 00061C                RVSF      =$E34B          ;STELLT VERVERS-FLAG UM
0117 00061C                KGETIN    =$FFE4          ;HOLE 1 ZEICHEN AUS TASTATUR-
                                       BUFFER
0118 00061C                KBASIN    =$FFCF          ;ZEICHEN VON AKTIVEM KANAL HO
                                       LEN
0119 00061C                OCRLF     =$AEC8          ;AUSGABE CR/LF AUF AKTIVE EIN
                                       HEIT
0120 00061C                BSOUT     =$FFD2          ;ZEICHEN ANZEIGEN, AKTIVER KA
                                       NAL
0121 00061C                PRT       =$E29B          ;ZEICHEN AUF BS AUSGEBEN
0122 00061C                SEQUEN    =$E66B          ;AUFRUF ESCAPE-FUNKTIONEN
0123 00061C                GETOUT    =$E30A          ;ENDE DER ZEICHENAUSGABE
0124 00061C                PAGSUB    =$E285          ;RETTET LAUFENDE BANK
0125                                 ;BISHERIGE BANK IN PAGSAV, HOLT ZEICHEN VOM STACK
0126 00061C                PAGRES    =$E290          ;RESTORE BANK FROM PAGSAV, AC
                                       CU UNAFFECTED
0127 00061C                KOPEN     =$FFC0
0128 00061C                KCLOSE    =$FFC3
0129 00061C                KCLRCH    =$FFCC          ;ALLE KANAELE SCHLIESSEN
0130 00061C                KBSOUT    =$FFD2
0131 00061C                KCHIN     =$FFC6
0132 00061C                ASSEMBLER =$6003          ;EIGENER 65816-ASSEMBLER
0133                                 ;**** MAIN PROGRAM
0134
0135 00061C                          *=$2003         ;IN BANK 15
0136 002003 A501           MAIN      LDA I6509       ;ZEIGER AUF BISHERIGE BANK
0137 002005 8DA003                   STA PAGSAV
0138 002008 A900                     LDA #NULL       ;NULL VOR DEN ANFANG BRINGEN
0139 00200A 8D0004                   STA INBUFF      ;NULL AN DEN ANFANG
0140 00200D 201A25                   JSR INITBANK1   ;BANKPOINTER EINRICHTEN
0141 002010 204125                   JSR INITBANK2
0142
0143                                 ;****** KOMMANDO-EBENE DES EDITORS
0144 002013                EDITOR
0145 002013 A00A                     LDY #$0A        ;ANSTEUERUNG DES CRTC-REGISTE
                                       RS
0146 002015 A960                     LDA #$60        ;BLINKING SOLID CURSOR
0147 002017 8C00D8                   STY CRTC        ;IN DIE REGISTER DES CRTC
0148 00201A 8D01D8                   STA CRTC+1
0149 00201D A902                     LDA #2          ;BANK 2 AKTIVIEREN
0150 00201F 8D0D06                   STA TEXTBANK
0151
0152 002022 A000           EDITOR1   LDY #MES0-MES0  ;ERSTEN TEXT AUSGEBEN
0153 002024 206426                   JSR MESOUT
0154 002027 A0B6                     LDY #MES11-MES0 ;BANKANZEIGE
0155 002029 206426                   JSR MESOUT
0156 00202C AD0D06                   LDA TEXTBANK
0157 00202F 0930                     ORA #'0'        ;ZU ASCII MACHEN
0158 002031 20D2FF                   JSR BSOUT       ;AUSGABE
0159 002034 20C8BE                   JSR OCRLF       ;NEUE ZEILE
0160
0161                                 ;KOMMANDOENTSCHLUESSELUNG
0162 002037 20C8BE         EDIT1     JSR OCRLF       ;NEWLINE
0163 00203A 206825         GETBEF    JSR GETCHR      ;ZEICHEN VOM KEYBOARD HOLEN
0164 00203D A216                     LDX #22         ;ZAHL-1 DER BEFEHLE
0165 00203F DD1D29         BEFPAR    CMP TASTEN,X    ;ERLAUBTE BEFEHLE
0166 002042 F006                     BEQ ONGOTO
0167 002044 CA                       DEX
0168 002045 10F8                     BPL BEFPAR
0169 002047 4C3A20                   JMP GETBEF      ;KEINE GUELTIGE TASTE
0170
0171 00204A 8A             ONGOTO    TXA             ;MULTIPLIZIERE X MIT 2
0172 00204B 0A                       ASL A
0173 00204C AA                       TAX
0174 00204D BD3529                   LDA JMPTBL+1,X
0175 002050 48                       PHA
0176 002051 BD3429                   LDA JMPTBL,X
0177 002054 48                       PHA
0178 002055 60                       RTS             ;EXECUTE COMMAND VIA STACK
0179
```

```
0180                            ;****** HAUPTSCHLEIFE DER TEXTEINGABE
0181 002056 A03E      TEXT_EIN  LDY #MES1-MESO      ;2. TEXT AUSGEBEN
0182 002058 206426              JSR MESOUT
0183
0184 00205B 206420    INLOOP    JSR GETSTRING       ;EINE ZEILE EINGEBEN
0185 00205E 20A025              JSR APPEND
0186 002061 4C5B20              JMP INLOOP
0187
0188                            ;**** UNTERPROGRAMM ZUR EINGABE EINER ZEILE
0189 002064 A900      GETSTRING LDA #00             ;NORMALER EINGABEMODUS
0190 002066 F002                BEQ GETSTR_1
0191 002068 A9FF      GETSTR_FF LDA #$FF            ;EINGABEMODUS BEI CHANGE
0192 00206A 8D0E06    GETSTR_1  STA MODE
0193 00206D AD0006              LDA ZL_LEN          ;ALTE ZEILENLAENGE AUTOMATISC
                                H SICHERN
0194 002070 8D0606              STA SAV_LEN
0195 002073 206825    GETSTR_2  JSR GETCHR          ;ZEICHEN VON TASTATUR HOLEN
0196 002076 A4CB      AUSGAB    LDY PNTR            ;CURSOR AM ZEILENENDE?
0197 002078 C04F                CPY #79
0198 00207A 9006                BCC INLIN5          ;NEIN
0199 00207C C90D                CMP #CR             ;KOMMT ZEILENABSCHLUSS?
0200 00207E F002                BEQ INLIN5          ;JA
0201 002080 A99D                LDA #$9D            ;CURSOR LEFT, MACH PLATZ FUER
                                  'CR'
0202 002082 48        INLIN5    PHA
0203 002083 C90D                CMP #CR             ;INVERS DARZUSTELLEN?
0204 002085 D011                BNE INLIN6
0205 002087 2C0E06              BIT MODE            ;NORMAL ODER CHANGE?
0206 00208A 300A                BMI INLIN5A         ;CHANGE
0207 00208C A912                LDA #REVERS_ON
0208 00208E 20D2FF              JSR BSOUT
0209 002091 A92F                LDA #'/'
0210 002093 20D2FF              JSR BSOUT
0211 002096 A90D      INLIN5A   LDA #CR
0212 002098 2098E2    INLIN6    JSR PRT             ;ZEICHEN AN BILDSCHIRM
0213 00209B A900                LDA #NULL           ;CANCEL QUOTE MODE
0214 00209D 85D2                STA QTSW            ;INS FLAG
0215 00209F 68        INLIN6A   PLA
0216 0020A0 C90D                CMP #CR             ;ZEILE JETZT UEBERNEHMEN?
0217 0020A2 F06F                BEQ FINI            ;JA
0218 0020A4 48                  PHA
0219 0020A5 297F                AND #$7F            ;STRIP OFF
0220 0020A7 C920                CMP #$20            ;FILTER FUER STEUERZEICHEN, A
                                UCH GESHIFTET
0221 0020A9 B031                BCS DRUCKB
0222
0223 0020AB 68                  PLA                 ;STEUERZEICHENBEHANDLUNG
0224 0020AC 8D6002              STA LSTCHR          ;ALS VORHERIGES ZEICHEN
0225 0020AF C914                CMP #DELETE
0226 0020B1 D01F                BNE NDRU1
0227 0020B3 2C0E06              BIT MODE            ;CHANGE?
0228 0020B6 3001                BMI DELCHR1         ;JA
0229 0020B8 C8        DELCHR    INY                 ;LINKSVERSCHIEBUNG IM BUFFER
0230 0020B9 C04F      DELCHR1   CPY #79             ;ZEILENENDE?
0231 0020BB F00D                BEQ ASPACE          ;JA, NOCH SPACE AN DAS ENDE
0232 0020BD B90004              LDA INBUFF,Y
0233 0020C0 88                  DEY
0234 0020C1 990004              STA INBUFF,Y
0235 0020C4 C6D5                DEC INDX            ;ZAHL BENUTZTE ZEICHEN
0236 0020C6 C8                  INY
0237 0020C7 4CB820              JMP DELCHR
0238 0020CA A920      ASPACE    LDA #SPACE
0239 0020CC 990004              STA INBUFF,Y
0240 0020CF 4C7320              JMP GETSTR_2
0241
0242 0020D2 C994      NDRU1     CMP #INSERT
0243 0020D4 D003                BNE NDRU2
0244 0020D6 208225              JSR INSCHR          ;AB CURSOR RECHTS RUECKEN
0245 0020D9 4C7320    NDRU2     JMP GETSTR_2
0246
0247 0020DC AD6002    DRUCKB    LDA LSTCHR          ;GING ESC VORAUS?
0248 0020DF C91B                CMP #ESCAPE
0249 0020E1 D010                BNE STORE           ; ALSO ASPEICHERN
```

```
0250 0020E3 68              PLA
0251 0020E4 8D6002          STA LSTCHR
0252 0020E7 C940            CMP #'$'                ;QUIT-BEFEHL? ESCAPE/KLAMMERA
                              FFE
0253 0020E9 F003            BEQ EXIT
0254 0020EB 4C7320          JMP GETSTR_2
0255
0256 0020EE 68        EXIT  PLA                     ;GOTO MONITOR PGM
0257 0020EF 68              PLA
0258 0020F0 4C2D21          JMP BOTTOM
0259
0260 0020F3 2C9A03    STORE BIT INSFLG              ;INSERT-MODE ON?
0261 0020F6 1003            BPL STORE1              ;NEIN
0262 0020F8 208225          JSR INSCHR              ;PLATZ MACHEN
0263 0020FB 68        STORE1 PLA
0264 0020FC A4CB            LDY PNTR                ;CURSORPOSITION
0265 0020FE 88              DEY
0266 0020FF 990004          STA INBUFF,Y            ;IN DEN EINGABEPUFFER
0267 002102 8D6002          STA LSTCHR
0268 002105 E6D5            INC INDX                ;ZAHL UEBERNOMMENE ZEICHEN
0269 002107 C4D5            CPY INDX
0270 002109 B003            BCS INLH
0271 00210B 4C7320          JMP GETSTR_2
0272
0273 00210E 84D5      INLH  STY INDX                ;JEWEILS HOECHSTER WERT
0274 002110 4C7320          JMP GETSTR_2
0275 002113 A4D5      FINI  LDY INDX                ;ROUTINE AB ZEILENABSCHLUSS
0276 002115 990004    FINI1 STA INBUFF,Y            ;CR AN DEN SCHLUSS
0277 002118 8C0006          STY ZL_LEN
0278 00211B 60              RTS                     ;ZEILE KANN JETZT UEBERNOMMEN
                              WERDEN
0279
0280
0281 00211C           TOP                           ;TOPZEILE ANSTEUERN, TASTE T
0282 00211C A057            LDY #MES2-MES0          ;TEXT ANZEIGEN
0283 00211E 206426          JSR MESOUT
0284 002121 209726          JSR TOPNO
0285 002124 20F125          JSR TOBUFR              ;TOPZEILE HOLEN
0286 002127 200E26          JSR DISLIN              ;UND ANZEIGEN
0287 00212A 4C3A20          JMP GETBEF
0288
0289 00212D           BOTTOM                        ;AN DAS TEXTENDE GEHEN, TASTE
                              B
0290 00212D 20A826          JSR SETBOT              ;NOWLN=BOTLN
0291 002130 20C88E    BOTTM1 JSR OCRLF              ;NEUE ZEILE
0292 002133 A05E      BOTTOM2 LDY #MES3-MES0
0293 002135 206426          JSR MESOUT
0294 002138 4C2220          JMP EDITOR1
0295
0296 00213B           UP                            ;1 ZEILE AUFWAERTS IM TEXT, T
                              ASTE U
0297 00213B 207126          JSR SUBNFELD            ;ZEILENZAHL-1
0298 00213E 204721          JSR PREV_LIN
0299 002141 200E26          JSR DISLIN              ;ZEIGE ZEILE
0300 002144 4C3A20          JMP GETBEF
0301 002147 A000      PREV_LIN LDY #0               ;POSITIONIERE AUF VORZEILE
0302 002149 20BF26          JSR PREVCH              ;VORHERGEHENDES ZEICHEN
0303 00214C D005            BNE PREVL0
0304 00214E 68        PREVL1 PLA                    ;ADJUST STACK
0305 00214F 68              PLA
0306 002150 4C1C21          JMP TOP                 ;NULL VOR TEXTBEGINN GEFUNDEN
0307 002153 20BF26    PREVL0 JSR PREVCH
0308 002156 F0F6            BEQ PREVL1              ;BEGRENZENDE NULL VOR DEM ANF
                              ANG
0309 002158 C90D      PREVL2 CMP #CR                ;ENDE DER VORZEILE?
0310 00215A D0F7            BNE PREVL0
0311 00215C C8              INY                     ;Y=1
0312 00215D 20B226          JSR ADD_YREG            ;NOWLN WIEDER +1
0313 002160 20F125          JSR TOBUFR              ;HOLE VORZEILE
0314 002163 20C88E          JSR OCRLF               ;IN NEUE ZEILE
0315 002166 60              RTS
0316
0317 002167           DOWN                          ;FOLGEZEILE SUCHEN/ANZEIGEN,
                              TASTE D
```

```
0318 002167 A000            LDY #NULL
0319 002169 20E725          JSR GET_FROM2       ;SIND WIR AM ENDE?
0320 00216C F00B            BEQ NEXTL1          ;JA, NICHT WEITER!
0321 00216E E6F0  DOWN2     INC NOWLN
0322 002170 D002            BNE DOWN3
0323 002172 E6F1            INC NOWLN+1
0324 002174 20E725  DOWN3   JSR GET_FROM2       ;HOLE/PRUEFE ZEICHEN
0325 002177 D003            BNE NEXTL2          ;ZEILE <> BOTTOM
0326 002179 4C2D21  NEXTL1  JMP BOTTOM
0327 00217C C90D    NEXTL2  CMP #CR             ;ZEILENENDE?
0328 00217E D0EE            BNE DOWN2           ;WEITER SUCHEN
0329 002180 20B526          JSR ADDNFELD        ;ZEILENZAHL
0330 002183 C8              INY                 ;Y=1
0331 002184 20E725          JSR GET_FROM2       ;WAS FOLGT AUF CR?
0332 002187 08              PHP                 ;NULLSTATUS?
0333 002188 20B226  NEXTL9  JSR ADD_YREG
0334 00218B 28              PLP                 ;WAR ES NULL
0335 00218C D003            BNE NEXTLA
0336 00218E 4C3021          JMP BOTTM1          ;DISPLAY BOTTOM
0337 002191 20C88E  NEXTLA  JSR OCRLF           ;CR AUSGEBEN
0338 002194 20F125  NEXTLA1 JSR TOBUFR          ;IN NEUE ZEILE
0339
0340 002197         RETURN                      ;LAUFENDE ZEILE ANZEIGEN, TAS
                            TE CR
0341 002197 200E26          JSR DISLIN          ;ZEILE AUS BUFFER ZEIGEN
0342 00219A 4C3A20  RETURN1 JMP GETBEF
0343
0344 00219D 60      QUIT    RTS                 ;EDITOR VERLASSEN, BASIC ZURU
                            ECK, TASTE Q
0345
0346 00219E A08F    LOESCH  LDY #MES12-MESO     ;TEXT EINER BANK LOESCHEN, TA
                            STE S
0347 0021A0 206426          JSR MESOUT
0348 0021A3 206825          JSR GETCHR          ;WAS WIRD GETIPPT?
0349 0021A6 48              PHA
0350 0021A7 20D2FF          JSR BSOUT           ;ECHO
0351 0021AA 20C88E          JSR OCRLF           ;ZEILENVORSCHUB
0352 0021AD 68              PLA                 ;ANTWORT-ZEICHEN
0353 0021AE C94A            CMP #'J'            ;JA?
0354 0021B0 F003            BEQ LOESCH1
0355 0021B2 4C3720          JMP EDIT1
0356 0021B5 AD0D06  LOESCH1 LDA TEXTBANK
0357 0021B8 C902            CMP #2
0358 0021BA F006            BEQ LOESCH2
0359 0021BC 201A25          JSR INITBANK1
0360 0021BF 4C3720          JMP EDIT1
0361 0021C2 204125  LOESCH2 JSR INITBANK2
0362 0021C5 4C2220          JMP EDITOR1
0363
0364 0021C8 A0A8    BANKSWIT LDY #MES10-MESO   ;BANKUMSCHALTUNG, TASTE /
0365 0021CA 206426          JSR MESOUT
0366 0021CD 206825          JSR GETCHR          ,ZEICHEN VON TASTATUR HOLEN
0367 0021D0 48              PHA
0368 0021D1 20D2FF          JSR BSOUT           ;ECHO
0369 0021D4 20C88E          JSR OCRLF           ;ZEILENVORSCHUB
0370 0021D7 68              PLA                 ;BANK-# FILTERN AUF 1-2
0371 0021D8 C931            CMP #'1'
0372 0021DA 90BE            BCC RETURN1         ;FEHLER
0373 0021DC C933            CMP #'3'
0374 0021DE B0BA            BCS RETURN1         ;FEHLER
0375 0021E0 290F            AND #$0F            ;JETZT BINAER
0376 0021E2 8D0C06          STA TEMP            ;ZUNAECHST POINTER UMSTELLEN
0377 0021E5 A6F0            LDX NOWLN           ,DATEN SCHON SICHERN
0378 0021E7 A4F1            LDY NOWLN+1
0379 0021E9 AD0D06          LDA TEXTBANK        ;WO WAREN WIR?
0380 0021EC CD0C06          CMP TEMP            ;IRRTUM?
0381 0021EF F0A9            BEQ RETURN1
0382 0021F1 C902            CMP #2
0383 0021F3 F02D            BEQ BANKSWIT1       ;NACH 1 UMSCHALTEN
0384 0021F5 8E0F06          STX NOWLN1          ;WIR WAREN IN BANK 1
0385 0021F8 8C1006          STY NOWLN1+1        ,UND SICHERN DEREN PARAMETER
0386 0021FB A6F2            LDX BOTLN
0387 0021FD A4F3            LDY BOTLN+1
0388 0021FF 8E1106          STX BOTLN1
```

```
0389 002202 8C1206            STY BOTLN1+1
0390 002205 AE1306            LDX NOWLN2          ;JETZT DIE PARAMETER DER ZIEL
                                  BANK EINSETZEN
0391 002208 AC1306            LDY NOWLN2
0392 00220B 86F0              STX NOWLN
0393 00220D 84F1              STY NOWLN+1
0394 00220F AE1506            LDX BOTLN2
0395 002212 AC1606            LDY BOTLN2+1
0396 002215 86F2    BANKSWIT2 STX BOTLN
0397 002217 84F3              STY BOTLN+1
0398 002219 AD0C06            LDA TEMP
0399 00221C 8D0D06            STA TEXTBANK        ;BANK NUN AKTIVERT
0400 00221F 4C2220            JMP EDITOR1         ;MENUE ANZEIGEN
0401
0402 002222 8E1306  BANKSWIT1 STX NOWLN2          ;WIR WAREN IN BANK 2
0403 002225 8C1406            STY NOWLN2+1        ,UND SICHERN DEREN PARAMETER
0404 002228 A6F2              LDX BOTLN
0405 00222A A4F3              LDY BOTLN+1
0406 00222C 8E1506            STX BOTLN2
0407 00222F 8C1606            STY BOTLN2+1
0408 002232 AE0F06            LDX NOWLN1          ;JETZT DIE PARAMETER DER ZIEL
                                  BANK EINSETZEN
0409 002235 AC0F06            LDY NOWLN1
0410 002238 86F0              STX NOWLN
0411 00223A 84F1              STY NOWLN+1
0412 00223C AE1106            LDX BOTLN1
0413 00223F AC1206            LDY BOTLN1+1
0414 002242 4C1522            JMP BANKSWIT2
0415
0416 002245 A0D5    NUMMER    LDY #MES13-MES0     ;SUCHEN ZEILE MIT NUMMER, TAB
                                  TE N
0417 002247 206426            JSR MESOUT
0418 00224A 20C88E            JSR OCRLF
0419 00224D A900              LDA #00
0420 00224F 85B9              STA TMP1            ;INITIALISIERUNG
0421 002251 85BA              STA TMP1+1
0422 002253 2023F1            JSR RDOA            ;4 ZAHLEN EINLESEN UND HEX-WA
                                  NDELN NACH TMP
0423 002256 20C88E            JSR OCRLF           ;NEUE ZEILE
0424 002259 209726  NUMMER2   JSR TOPND           ;AUF 1. ZEILE EINSTELLEN
0425 00225C A5B9              LDA TMP1
0426 00225E A6BA              LDX TMP1+1
0427 002260 8D1706            STA NFELD
0428 002263 8E1806            STX NFELD+1         ;PARMS UEBERTRAGEN
0429 002266 AD0D06            LDA TEXTBANK
0430 002269 8501              STA I6509
0431 00226B AD1706  NUMMER0   LDA NFELD
0432 00226E 0D1806            ORA NFELD+1
0433 002271 F016              BEQ NUMMERZ         ;ZEILE ERREICHT
0434 002273 A000              LDY #0
0435 002275 B1F0    NUMMER1   LDA (NOWLN),Y
0436 002277 F013              BEQ NUMMERB         ;TEXTENDE
0437 002279 C8                INY                 ;LAUFZEIGER DER ZEILE
0438 00227A C90D              CMP #CR
0439 00227C F002              BEQ NUMMER4
0440 00227E D0F5              BNE NUMMER1
0441
0442 002280 20B226  NUMMER4   JSR ADD_YREG        ;POINTER ERHOEHEN
0443 002283 207126            JSR SUBNFELD        ;ELSE -1 ZEILEN SCANNEN
0444 002286 4C6B22            JMP NUMMER0         ;WEITER SUCHEN
0445 002289 4C9421  NUMMERZ   JMP NEXTLA1         ;ANZEIGEN, FERTIG
0446
0447 00228C 2090E2  NUMMERB   JSR PAGRES
0448 00228F 4C2D21            JMP BOTTOM          ;TEXTENDE ANZEIGEN
0449
0450 002292         KILL                          ;ANGEZEIGTE ZEILE ENTFERNEN,
                                  TASTE K
0451 002292 200E26            JSR DISLIN          ;ZEILE NOCHMALS ZEIGEN, LAENG
                                  E IN INDX
0452 002295 A09E              LDY #MES9-MES0
0453 002297 206426            JSR MESOUT
0454 00229A A000              LDY #0
0455 00229C 20E725            JSR GET_FROM2
0456 00229F D003              BNE KILL1           ;SKIP
```

```
0457 0022A1 4C2D21          JMP BOTTOM          ;SCHON AM ENDE
0458 0022A4 206627  KILL1   JSR KILSUB
0459 0022A7 4C9421          JMP NEXTLA1         ;FOLGEZEILE HOLEN/ANZEIGEN
0460
0461 0022AA A900    NSERT   LDA #0              ;MODUS NORMAL
0462 0022AC 8D1906  NSERT1  STA INS_FLAG
0463 0022AF 20C88E          JSR CRLF            ;1 ZEILE EINSCHIEBEN, TASTE I
0464 0022B2 A08A            LDY #MES7-MES0      ;INTERAKTIVE AUSGABE
0465 0022B4 206426          JSR MESOUT
0466 0022B7 206420  NSERT2  JSR GETSTRING       ;EINGABEZEILE VON TASTATUR HO
                            LEN
0467 0022BA 20C822          JSR MOV_PACK        ;ROUTINEN ZUM EINGLIEDERN
0468 0022BD 2C1906          BIT INS_FLAG        ;MODUS?
0469 0022C0 30F5            BMI NSERT2          ;LAUFENDE EINGABE
0470 0022C2 20F125          JSR TOBUFR          ;FOLGEZEILE HOLEN
0471 0022C5 4C9721          JMP RETURN          ;UND ANZEIGEN
0472
0473 0022C8 20CF26  MOV_PACK JSR MOVBACK        ;FUEGE EINGABEZEILE IN DEN TE
                            XTBUFFER E
0474 0022CB 20C925          JSR TONOWL          ;ZEILE BEI NOWLN EINFUEGEN
0475 0022CE 204E27          JSR KORRBOTLN       ;ZEIGER BERICHTIGEN
0476 0022D1 4CAF26          JMP ADDZL_LEN       ;SETZE NOWLN AUF FOLGENDE Z.,
                             RTS DORT
0477
0478 0022D4 A9FF    READNSERT LDA #$FF          ;READ FILE, INSERT AB NOWLN,
                            TASTE O
0479 0022D6 4C7623          JMP READ_IN         ;WEITER WIE BEI READ
0480
0481 0022D9 A9FF    LIN_NSERT LDA #$FF          ;SIGNALISIERE INSERTMODUS
0482 0022DB D0CF            BNE NSERT1          ;LAUFENDE ZEILEN-EINSCHIEBUNG
                            , TASTE <
0483
0484 0022DD         WRIT_FLOP                   ;WRITE FILE TO FLOPPY, TASTE
                            &
0485 0022DD 209726          JSR TOPNO           ;AUF 1. ZEILE EINSTELLEN
0486 0022E0 20CCFF          JSR KCLRCH          ;ALLES ERST SCHLIESSEN
0487 0022E3 203D26          JSR OPEN_FLOP       ;OPEN FLOPPY
0488 0022E6 20AEFF          JSR KUNLSN
0489 0022E9 A69E            LDX LA              ;LOGISCHE ADRESSE
0490 0022EB 20C9FF          JSR KCKOUT          ;LOGISCHES FILE OEFFNEN
0491 0022EE A908            LDA #8              ;DEVICE
0492 0022F0 85A2            STA DFLTO
0493 0022F2 A903            LDA #3
0494 0022F4 2093FF          JSR KSECND          ;SEKUNDAERADRESSE
0495 0022F7 A900            LDA #0
0496 0022F9 859C            STA STATUS
0497 0022FB A000    WRIT_F1 LDY #0
0498 0022FD 20E725          JSR GET_FROM2       ;ENDE ODER LEERER BUFFER?
0499 002300 F030            BEQ WRIT_END        ;JA
0500 002302 20F125          JSR TOBUFR
0501
0502 002305 A000            LDY #0
0503 002307 8C0C06  WRIT_F3 STY TEMP
0504 00230A A59F            LDA FA
0505 00230C 20B1FF          JSR KLISTN
0506 00230F A5A0            LDA SA
0507 002311 2093FF          JSR KSECND
0508 002314 AC0C06          LDY TEMP
0509 002317 B90004          LDA INBUFF,Y        ;HOLE ZEICHEN
0510 00231A F016            BEQ WRIT_END        ;TEXTENDE
0511 00231C 48              PHA
0512 00231D 20A8FF          JSR KCIOUT          ;AUSGABE AUF DEVICE IEEE
0513 002320 20AEFF          JSR KUNLSN
0514 002323 68              PLA
0515 002324 AC0C06          LDY TEMP
0516 002327 C8              INY
0517 002328 C90D            CMP #CR
0518 00232A D0DB            BNE WRIT_F3
0519 00232C 20B226          JSR ADD_YREG
0520 00232F 4CFB22          JMP WRIT_F1
0521
0522 002332 2090E2  WRIT_END JSR PAGRES         ;BANK HERSTELLEN
0523 002335 4CBE23          JMP READEND         ;FILE SCHLIESSEN
0524
```

```
0525 002338 20CCFF    PRINT     JSR KCLRCH          ;PRINTER-AUSGABE PER 'P', CLO
                                  SE CHANNELS
0526 00233B A904                LDA #4              ;DEVICE 4
0527 00233D 20B1FF              JSR KLISTN          ;LISTEN
0528 002340 A9FF                LDA #255
0529 002342 2093FF              JSR KSECND          ;OHNE SEKUNDARADRESSE
0530 002345 20E1FF    PRINT_1   JSR KSTOP           ;STOP-TASTE GEDRUECKT?
0531 002348 F01F                BEQ PRINT_END       ;JA
0532 00234A A000                LDY #0
0533 00234C 20E725              JSR GET_FROM2       ;TEXTENDE?
0534 00234F F018                BEQ PRINT_END
0535 002351 20F125              JSR TOBUFR          ;ZEILE HOLEN
0536 002354 A000                LDY #0
0537 002356 B90004    PRINT_2   LDA INBUFF,Y
0538 002359 48                  PHA
0539 00235A 20A8FF              JSR KCIOUT          ;AUF DEN BUS
0540 00235D 68                  PLA                 ;ZEICHEN
0541 00235E C8                  INY
0542 00235F C90D                CMP #CR
0543 002361 D0F3                BNE PRINT_2
0544 002363 20B226              JSR ADD_YREG        ;NOWLN ERHOEHEN
0545 002366 4C4523              JMP PRINT_1
0546 002369 A90D      PRINT_END LDA #CR
0547 00236B 20A8FF              JSR KCIOUT
0548 00236E 20AEFF              JSR KUNLSN          ;GERAET ABSCHALTEN
0549 002371 4C9721              JMP RETURN          ;ZEILE NOCHMALS ANZEIGEN
0550
0551 002374           READ                          ;TEXT EINZIEHEN VOM SEQ FILE,
                                  TASTE R
0552 002374 A900                LDA #0              ;MODUS MIT ANHAENGEN, SONST I
                                  NSERT-MODUS MIT #FF
0553 002376 8D1906    READ_IN   STA INS_FLAG
0554 002379 203D26              JSR OPEN_FLOP
0555
0556 00237C A69E                LDX LA
0557 00237E 20C6FF              JSR KCHIN
0558
0559 002381 A000      READF0    LDY #0
0560 002383 8C0006              STY ZL_LEN
0561 002386 20E4FF    READ1     JSR KGETIN
0562 002389 A8                  TAY                 ZEICHEN SICHERN
0563 00238A A59C                LDA STATUS          ;GGFS. FILE UNBEKANNT
0564 00238C 4A                  LSR A
0565 00238D 4A                  LSR A
0566 00238E B04B                BCS READERR         ;FEHLER
0567 002390 98                  TYA
0568 002391 4C9923              JMP READ3
0569 002394 20E4FF    READFILE  JSR KGETIN
0570 002397 F0FB      READ2     BEQ READFILE
0571 002399 AC0006    READ3     LDY ZL_LEN
0572 00239C 990004              STA INBUFF,Y
0573 00239F EE0006              INC ZL_LEN
0574 0023A2 249C                BIT STATUS
0575 0023A4 7018                BVS READEND
0576 0023A6 C90D                CMP #CR
0577 0023A8 D0EA                BNE READFILE
0578 0023AA 2C1906              BIT INS_FLAG        ;INSERT ODER APPEND?
0579 0023AD 1009                BPL READ_APP
0580 0023AF CE0006              DEC ZL_LEN
0581 0023B2 20CB22              JSR MOV_PACK        ;MOVE-PACKAGE AUFRUFEN
0582 0023B5 4CB123              JMP READF0
0583 0023B8 20A025    READ_APP  JSR APPEND          ;ZEILENTRANSPORT UND POINTER
                                  ERHOEHEN
0584 0023BB 4CB123              JMP READF0
0585
0586 0023BE 2C1906    READEND   BIT INS_FLAG
0587 0023C1 1009                BPL READENDA
0588 0023C3 CE0006              DEC ZL_LEN
0589 0023C6 20CB22              JSR MOV_PACK        ;INSERT-MODE
0590 0023C9 4CCF23              JMP READEND1
0591 0023CC 20A025    READENDA  JSR APPEND          ;EOF ERKANNT, ZEILE NOCH ANHA
                                  ENGEN
0592 0023CF A901      READEND1  LDA #1              ;LOG. FILE-#
```

```
0593 0023D1 38       READENDC SEC
0594 0023D2 20C3FF            JSR KCLOSE          ;CLOSE EINGABEKANAL
0595 0023D5 20CCFF            JSR KCLRCH
0596 0023D8 4C2220            JMP EDITOR1
0597
0598 0023DB 2042F9   READERR  JSR ERROR4          ;FILE NOT FOUND
0599 0023DE 4CCF23            JMP READEND1        ;CLOSE + BEFEHLSEBENE
0600
0601 0023E1          LIST                         ;TEXT LISTEN, TASTE L
0602 0023E1 A094              LDY #MES8-MES0
0603 0023E3 206426            JSR MESOUT
0604 0023E6 20F12S   LIST_1   JSR TOBUFR          ;ZEILE HOLEN
0605 0023E9 C900              CMP #NULL           ;END OF TEXT?
0606 0023EB F026              BEQ LIST_2
0607 0023ED 200E26            JSR DISLIN
0608 0023F0 20AF26            JSR ADDZL_LEN
0609 0023F3 208526            JSR ADDNFELD        ;ZEILENZAHL +1
0610 0023F6 20C88E            JSR OCRLF
0611 0023F9 20E4FF            JSR KGETIN          ;IRGENDWAS GEDRUECKT?
0612 0023FC C900              CMP #0
0613 0023FE F0E6              BEQ LIST_1
0614 002400 C923              CMP #'#'            ;ZEILEN-NR ANZEIGEN?
0615 002402 F012              BEQ LIST_N
0616 002404 C951              CMP #'Q'            ;QUIT?
0617 002406 F008              BEQ LIST_12
0618 00240B 20E4FF   LIST_11  JSR KGETIN          ;WIRD TASTE LOSGELASSEN
0619 00240B D0FB              BNE LIST_11         ;WARTESCHLEIFE
0620 00240D 4CE623            JMP LIST_1          ;WEITER LISTEN
0621 002410 4C3A20   LIST_12  JMP GETBEF
0622 002413 4C3321   LIST_2   JMP BOTTOM2         ;ANZEIGE BOTTOM
0623
0624 002416 20C88E   LIST_N   JSR OCRLF
0625 002419 A912              LDA #REVERS_ON
0626 00241B 20D2FF            JSR BSOUT
0627 00241E AE1806            LDX NFELD+1         ;AUSGABE DER ZEILENZAHL IM FL
                                  UGE
0628 002421 AC1706            LDY NFELD
0629 002424 20F6F0            JSR WROA            ;AUSGABE MIT KERNEL
0630 002427 20C88E            JSR OCRLF
0631 00242A 20E4FF   LIST_N1  JSR KGETIN
0632 00242D D0FB              BNE LIST_N1         ,WARTEN BIS TASTE LOSGELASSEN
0633 00242F 4C3A20            JMP GETBEF
0634
0635 002432          FIND                         ;ZEICHENKETTE FINDEN, TASTE F
0636 002432 A064              LDY #MES4-MES0
0637 002434 206426            JSR MESOUT
0638 002437 206420            JSR GETSTRING       ;SUCHSTRING ANFORDERN
0639 00243A AD0006            LDA ZL_LEN
0640 00243D 8D0206            STA SUCH_LEN        ;LAENGE MERKEN
0641 002440 8D0306            STA SUCH_LENW       ;VORMERKUNG F. WEIT_SUCH
0642 002443 A000              LDY #0              ;DEN SUCHSTRING MERKEN
0643 002445 B90004   FIND1    LDA INBUFF,Y        ;EINGABEPUFFER
0644 002448 990005            STA FINDBUFF,Y
0645 00244B C8                INY
0646 00244C C90D              CMP #CR             ;ZEILENENDE?
0647 00244E D0F5              BNE FIND1
0648 002450 20AC27   FIND2    JSR SUCH            ;SUCHE DURCHFUEHREN
0649 002453 20F125   FIND2A   JSR TOBUFR          ;ZEILE HOLEN
0650 002456 A912     FIND21   LDA #REVERS_ON
0651 002458 20D2FF            JSR BSOUT
0652 00245B 200E26            JSR DISLIN          ;ZEIGT DEN TEXT AB FUNDSTELLE
0653 00245E A000              LDY #0
0654 002460 20BF26            JSR PREVCH          ;SIND WIR AM ZEILENANFANG?
0655 002463 48                PHA
0656 002464 A001              LDY #1
0657 002466 20B226            JSR ADD_YREG        ;NOWLN WIEDER EINRICHTEN
0658 002469 68                PLA                 ;VOR-ZEICHEN?
0659 00246A F004              BEQ FIND2B
0660 00246C C90D              CMP #CR
0661 00246E D006              BNE FIND3           ;NEIN
0662 002470 200E26   FIND2B   JSR DISLIN          ;SICHERT AUCH NACH SAV_LEN
0663 002473 4C3A20            JMP GETBEF
0664 002476 204721   FIND3    JSR PREV_LIN        ;UND ANZEIGEN AB BEGINN
0665 002479 4C7024            JMP FIND2B
0666
```

```
0667 00247C 20C88E   WEIT_SUCH JSR OCRLF
0668 00247F 20F125             JSR TOBUFR
0669 002482 20AF26             JSR ADDZL_LEN
0670 002485 20F125             JSR TOBUFR
0671 002488 20AF26             JSR ADDZL_LEN      ;NAECHSTE ZEILE
0672 00248B 20E327             JSR SUCH3
0673 00248E 4C5324             JMP FIND2A
0674
0675 002491          CHANGE                       ;STRING ERSETZEN, TASTE C
0676 002491 A912               LDA #REVERS_ON
0677 002493 20D2FF             JSR BSOUT
0678 002496 200E26             JSR DISLIN
0679 002499 206820             JSR GETSTR_FF      ;EINGABEMODUS FUER CHANGE
0680 00249C CE0106             DEC OLD_LEN
0681 00249F AE0D06             LDX TEXTBANK       ;AKTIVE BANK
0682 0024A2 8601               STX I6509
0683 0024A4 38                 SEC
0684 0024A5 AD0006             LDA ZL_LEN
0685 0024A8 ED0106             SBC OLD_LEN
0686 0024AB D006               BNE CHANGE1
0687 0024AD 20C925             JSR TONOWL         ;ZEILE GLEICHER LAENGE UEBERN
                               EHMEN
0688 0024B0 4C9421             JMP NEXTLA1
0689
0690 0024B3 A000     CHANGE1   LDY #0
0691 0024B5 20E725             JSR GET_FROM2      ;LAUFENDES ZEICHEN
0692 0024B8 D003               BNE CHANGE2
0693 0024BA 4C2D21             JMP BOTTOM         ;WAR SCHON TEXTENDE, ANZEIGEN
0694
0695 0024BD CE0006   CHANGE2   DEC ZL_LEN         ;WEGEN DES BEGRENZER-ZEICHENS
0696 0024C0 CE0106             DEC OLD_LEN
0697 0024C3 AD0006             LDA ZL_LEN
0698 0024C6 8D0C06             STA TEMP           ;DATENSICHERUNG
0699 0024C9 38                 SEC
0700 0024CA AD0006             LDA ZL_LEN
0701 0024CD ED0106             SBC OLD_LEN
0702 0024D0 901D               BCC RAFFEN         ;NEUER STRING IST KUERZER
0703 0024D2 8D0006             STA ZL_LEN         ;NUR UM DIE DIFFERENZ NACH RE
                               CHTS RUECKEN
0704 0024D5 CE0006             DEC ZL_LEN
0705 0024D8 20CF26             JSR MOVBACK        ;ES WIRD GERUECKT
0706 0024DB 204E27             JSR KORRBOTLN      ;ZEIGER BOTLN KORRIGIEREN
0707 0024DE AD0C06   CHANGE21  LDA TEMP           ;LAENGE EINGABESTRING
0708 0024E1 8D0006             STA ZL_LEN         ;WIEDERHERSTELLEN
0709 0024E4 20C925             JSR TONOWL         ;TRANSPORT IN DEN SPEICHER
0710 0024E7 A991               LDA #CURSOR_UP
0711 0024E9 20D2FF             JSR BSOUT
0712 0024EC 4C3A20             JMP GETBEF
0713
0714 0024EF 38       RAFFEN    SEC
0715 0024F0 AD0106             LDA OLD_LEN        ;VERSATZMENGE FUER KILSUB
0716 0024F3 ED0C06             SBC TEMP
0717 0024F6 8D0006             STA ZL_LEN
0718 0024F9 A5F1               LDA NOWLN+1        ;KORRIGIERTE ZEILE IST KUERZE
                               R
0719 0024FB 48                 PHA                SICHERN
0720 0024FC A5F0               LDA NOWLN
0721 0024FE 48                 PHA                SICHERN
0722 0024FF 18                 CLC
0723 002500 6D0C06             ADC TEMP           ;LAENGE EINGABEZEILE
0724 002503 6902               ADC #2
0725 002505 85F0               STA NOWLN          ;ZIELADRESSE B. VERSCHIEBUNG
0726 002507 A5F1               LDA NOWLN+1
0727 002509 6900               ADC #0
0728 00250B 85F1               STA NOWLN+1
0729 00250D 207027             JSR KILSUB1        ;BLOCK MOVE AUSFUEHREN
0730 002510 68                 PLA
0731 002511 85F0               STA NOWLN
0732 002513 68                 PLA
0733 002514 85F1               STA NOWLN+1        ;ALTER POINTER W. HERGESTELLT
0734 002516 4CDE24             JMP CHANGE21
0735
0736 002519 00       MONITOR   BRK                ;EXIT FO MONITOR WITH 'M'
0737
```

```
0738                            ;***** DIVERSE UNTERPROGRAMME
0739 00251A A000     INITBANK1  LDY #00              BANK 1 INITIALISIEREN
0740 00251C A901                LDA #1               ;BANK ANSTEUERN
0741 00251E 8D0D06              STA TEXTBANK
0742 002521 209726              JSR TOPNO            ;STELLE TOPZEILE EIN
0743 002524 85F2                STA BOTLN            ;POINTER AUF ENDE TEXTSPEICHE
                                R
0744 002526 86F3                STX BOTLN+1
0745 002528 8D0F06              STA NOWLN1           ;RESERVEZEIGER
0746 00252B 8E1006              STX NOWLN1+1
0747 00252E 8D1106              STA BOTLN1
0748 002531 8E1206              STX BOTLN1+1
0749 002534 C6F2                DEC BOTLN            ;=2
0750 002536 A900                LDA #00
0751 002538 20DD25              JSR PUT_TO_2         ;ABSPEICHERN NACH 020002
0752 00253B E6F2                INC BOTLN            ;=WIEDER ANFANG
0753 00253D 20DD25              JSR PUT_TO_2         ;NULL ALS ERSTES ZEICHEN
0754 002540 60                  RTS
0755
0756 002541 A000     INITBANK2  LDY #00              BANK 2 INITIALISIEREN
0757 002543 A902                LDA #2               ;BANK ANSTEUERN
0758 002545 8D0D06              STA TEXTBANK
0759 002548 209726              JSR TOPNO            ;STELLE TOPZEILE EIN
0760 00254B 85F2                STA BOTLN            ;POINTER AUF ENDE TEXTSPEICHE
                                R
0761 00254D 86F3                STX BOTLN+1
0762 00254F 8D1306              STA NOWLN2           ;RESERVEZEIGER
0763 002552 8E1406              STX NOWLN2+1
0764 002555 8D1506              STA BOTLN2
0765 002558 8E1606              STX BOTLN2+1
0766 00255B C6F2                DEC BOTLN            ;=2
0767 00255D A900                LDA #0
0768 00255F 20DD25              JSR PUT_TO_2         ;ABSPEICHERN NACH 020002
0769 002562 E6F2                INC BOTLN            ;=WIEDER ANFANG
0770 002564 20DD25              JSR PUT_TO_2         ;NULL ALS ERSTES ZEICHEN
0771 002567 60                  RTS
0772
0773 002568 20E4FF   GETCHR     JSR KGETIN           ;ZEICHEN AUS TASTATURPUFFER H
                                OLEN
0774 00256B C900                CMP #NULL            ;LEER?
0775 00256D F0F9                BEQ GETCHR
0776 00256F C991                CMP #$91             ;CURSOR UP?
0777 002571 F0F5                BEQ GETCHR           ;STEUERZEICHEN ABFILTERN
0778 002573 C913                CMP #$13             ;HOME?
0779 002575 F0F1                BEQ GETCHR
0780 002577 C993                CMP #$93             ;CLEAR/HOME?
0781 002579 F0ED                BEQ GETCHR
0782
0783 00257B C911                CMP #$11             ;CURSOR DOWN ZU CR MACHEN
0784 00257D D002                BNE GETEND
0785 00257F A90D                LDA #CR
0786 002581 60       GETEND     RTS
0787
0788                            ;******* VERSCHIEBUNGEN BEI INSERT
0789 002582 A04F     INSCHR     LDY #79              ;MAX-SPALTE
0790 002584 C4D5                CPY INDX             ;SCHON ERREICHT?
0791 002586 F017                BEQ INSEND           ;JA
0792 002588 88       INSLOP     DEY                  ;ZEIGE AUF ZEICHEN DAVOR
0793 002589 B90004              LDA INBUFF,Y
0794 00258C C8                  INY                  ;ZEIGE AUF STELLE DAHINTER
0795 00258D 990004              STA INBUFF,Y
0796 002590 88                  DEY
0797 002591 C4CB                CPY PNTR             ;INSERT-STELLE ERREICHT?
0798 002593 D0F3                BNE INSLOP
0799
0800 002595 2C9A03              BIT INSFLG           ;IM INSERT-MODE KEINEN SPACE!
0801 002598 3005                BMI INSEND
0802 00259A A920                LDA #SPACE           ;DORT SPACE EINSETZEN
0803 00259C 990004              STA INBUFF,Y
0804 00259F 60       INSEND     RTS
0805
0806 0025A0 A000     APPEND     LDY #NULL            ;TRANSPORT DER EINGEGEBENEN Z
                                EILE
0807 0025A2 208725              JSR APPENDBOT
0808 0025A5 A900                LDA #NULL            ;NULL AN DAS TEXTENDE
```

```
0809 0025A7 20DD25          JSR PUT_TO_2
0810
0811 0025AA 98              TYA                 ;LAENGE DER ZEICHENKETTE
0812 0025AB 18              CLC                 ;ZUM POINTER BOTLN ADDIEREN
0813 0025AC 65F2            ADC BOTLN
0814 0025AE 85F2            STA BOTLN
0815 0025B0 A5F3            LDA BOTLN+1
0816 0025B2 6900            ADC #NULL
0817 0025B4 85F3            STA BOTLN+1
0818 0025B6 60              RTS
0819
0820 0025B7          APPENDBOT                  ;HOLE ZEILE AUS EINGABE-BUFFE
                                                R
0821 0025B7 B90004          LDA INBUFF,Y
0822 0025BA AE0D06          LDX TEXTBANK        ;SPEICHERE EIN ZEICHEN IN DIE
                              TEXTBANK
0823 0025BD 8601            STX I6509
0824 0025BF 91F2            STA (BOTLN),Y       ;IM AKKU UEBERGEBENES ZEICHEN
0825 0025C1 C8              INY
0826 0025C2 C90D            CMP #CR             ;ZEILENENDE?
0827 0025C4 D0F1            BNE APPENDBOT       ;NEIN
0828 0025C6 4C90E2          JMP PAGRES          ;ALTE BANK WIEDER EINRICHTEN
                            + RTS
0829
0830 0025C9 A000     TONOWL LDY #NULL           ;EINGABEZEILE NACH NOWLN BRIN
                            GEN
0831 0025CB AE0D06          LDX TEXTBANK
0832 0025CE 8601            STX I6509
0833 0025D0 B90004   TONOW1 LDA INBUFF,Y        ;VOM BUFFER HOLEN
0834 0025D3 91F0            STA (NOWLN),Y
0835 0025D5 C8              INY
0836 0025D6 C90D            CMP #CR             ;ZEILENENDE?
0837 0025D8 D0F6            BNE TONOW1          ;NEIN
0838 0025DA 4C90E2          JMP PAGRES
0839
0840 0025DD AE0D06 PUT_TO_2 LDX TEXTBANK        ;BRINGE 1 ZEICHEN NACH BANK 2
0841 0025E0 8601            STX I6509
0842 0025E2 91F2            STA (BOTLN),Y
0843 0025E4 4C90E2          JMP PAGRES
0844
0845 0025E7 AE0D06 GET_FROM2 LDX TEXTBANK       ;HOLE EIN ZEICHEN VON BANK 2
0846 0025EA 8601            STX I6509           ;BANKUMSCHALTUNG
0847 0025EC B1F0            LDA (NOWLN),Y       ;ZEICHEN
0848 0025EE 4C90E2          JMP PAGRES          ;BANK UMSCHALTEN
0849
0850 0025F1 A0FF     TOBUFR LDY #$FF           ;HOLE 1 ZEILE AUS BANK 2 IN Z
                            EILENBUFFER
0851 0025F3 AE0D06   TOBUF1 LDX TEXTBANK
0852 0025F6 8601            STX I6509
0853 0025F8 C8              INY
0854 0025F9 B1F0            LDA (NOWLN),Y       ;ZEICHEN
0855 0025FB 990004          STA INBUFF,Y        ;ZEICHEN
0856 0025FE C900            CMP #NULL           ;ZEILE LEER?
0857 002600 F004            BEQ TOBUF2
0858 002602 C90D            CMP #CR             ;ZEILENENDE?
0859 002604 D0ED            BNE TOBUF1
0860 002606 84D5     TOBUF2 STY INDX            ;ZEILENLAENGE FESTHALTEN
0861 002608 8C0006          STY ZL_LEN
0862 00260B 4C90E2          JMP PAGRES          ;ALTE BANK UND RTS
0863
0864                        ;***** DISPLAY LINE IN LINE-BUFFER
0865 00260E A000     DISLIN LDY #0              ;ZEICHENZEIGER
0866 002610 B90004   DISLN1 LDA INBUFF,Y        ;ZEICHEN
0867 002613 F025            BEQ NOLIN           ;LEER
0868 002615 48              PHA
0869 002616 C90D            CMP #CR             ;INVERS DARZUSTELLEN?
0870 002618 D00C            BNE DISLIN2
0871 00261A A912            LDA #REVERS_ON
0872 00261C 20D2FF          JSR BSOUT
0873 00261F A92F            LDA #'/'
0874 002621 20D2FF          JSR BSOUT
0875 002624 A90D            LDA #CR
0876 002626 20D2FF   DISLIN2 JSR BSOUT
0877 002629 C8              INY
```

```
0878 00262A 68              PLA
0879 00262B C90D            CMP #CR             ;ZEILENENDE?
0880 00262D D0E1            BNE DISLN1
0881 00262F 8C0006          STY ZL_LEN
0882 002632 8C0106          STY OLD_LEN
0883 002635 A991            LDA #CURSOR_UP
0884 002637 20D2FF          JSR BSOUT
0885 00263A 4C90E2  NOLIN   JMP PAGRES          ;ALTE BANK UND RTS
0886
0887 00263D A07C    OPEN_FLOP LDY #MES6-MES0    ;OPEN FLOPPY CHANNEL 1,3,8
0888 00263F 206426          JSR MESOUT
0889 002642 206420          JSR GETSTRING       ;ZEILE HOLEN
0890 002645 849D            STY FNLEN           ;LAENGE FILENAME
0891 002647 A900            LDA #<INBUFF
0892 002649 8590            STA FNADR
0893 00264B A904            LDA #>INBUFF
0894 00264D 8591            STA FNADR+1         ;ZEIGE AUF FILENAME
0895 00264F A901            LDA #1              ;LOG FILE
0896 002651 859E            STA LA
0897 002653 A903            LDA #3
0898 002655 85A0            STA SA
0899 002657 A908            LDA #8
0900 002659 859F            STA FA
0901 00265B A900            LDA #0
0902 00265D 859C            STA STATUS
0903 00265F 18              CLC
0904 002660 20C0FF          JSR KOPEN           ;OPEN 1,3,8, FILENAME
0905 002663 60              RTS
0906                        ;***** INTERAKTIVE AUSGABEN
0907 002664 B92928  MESOUT  LDA MES0,Y
0908 002667 F007            BEQ TXTEND
0909 002669 20D2FF          JSR BSOUT
0910 00266C C8              INY
0911 00266D 4C6426          JMP MESOUT
0912 002670 60      TXTEND  RTS
0913
0914 002671 F8      SUBNFELD SED                ;DEZIMAL ARBEITEN, ZEILENZAHL
                                -1
0915 002672 38              SEC
0916 002673 AD1706          LDA NFELD           ;-1 ZEILE ZU SUCHEN
0917 002676 E901            SBC #1
0918 002678 8D1706          STA NFELD
0919 00267B AD1806          LDA NFELD+1
0920 00267E E900            SBC #0
0921 002680 8D1806  SUBNFELD1 STA NFELD+1
0922 002683 D8              CLD                 ;WIEDER BINAER
0923 002684 60              RTS
0924 002685 F8      ADDNFELD SED                ;DEZIMAL ARBEITEN, ZEILENZAHL
                                +1
0925 002686 18              CLC
0926 002687 AD1706          LDA NFELD           ;-1 ZEILE ZU SUCHEN
0927 00268A 6901            ADC #1
0928 00268C 8D1706          STA NFELD
0929 00268F AD1806          LDA NFELD+1
0930 002692 6900            ADC #0
0931 002694 4C8026          JMP SUBNFELD1
0932
0933 002697 A900    TOPNO   LDA #0
0934 002699 8D1706          STA NFELD           ;ZEILENZAEHLER=0
0935 00269C 8D1806          STA NFELD+1
0936 00269F A903            LDA #<ANFANG        ;RICHTE TEXTSPEICHER EIN
0937 0026A1 A200            LDX #>ANFANG
0938 0026A3 85F0    TOPNO1  STA NOWLN           ;ZEIGER AUF TOPZEILE
0939 0026A5 86F1            STX NOWLN+1
0940 0026A7 60              RTS
0941
0942 0026A8 A5F2    SETBOT  LDA BOTLN           ;SETZE NOWLN AUF TEXTENDE
0943 0026AA A6F3            LDX BOTLN+1
0944 0026AC 4CA326          JMP TOPNO1
0945
0946 0026AF AC0006  ADDZL_LEN LDY ZL_LEN
0947 0026B2 18      ADD_YREG CLC                ;ADDIERE Y ZU NOWLN
0948 0026B3 98              TYA
0949 0026B4 65F0            ADC NOWLN
```

```
0950 0026B6 85F0          STA NOWLN
0951 0026B8 A5F1          LDA NOWLN+1
0952 0026BA 6900          ADC #0
0953 0026BC 85F1          STA NOWLN+1
0954 0026BE 60            RTS
0955
0956 0026BF 38      PREVCH SEC                ;ERNIEDRIGE NOWLN
0957 0026C0 A5F0          LDA NOWLN           ;SUBTRAHIERE JEWEILS 1
0958 0026C2 E901          SBC #1
0959 0026C4 85F0          STA NOWLN
0960 0026C6 A5F1          LDA NOWLN+1
0961 0026C8 E900          SBC #0
0962 0026CA 85F1          STA NOWLN+1         ;POINTER ERNIEDRIGT
0963 0026CC 4CE725        JMP GET_FROM2       ;HOLE 1 ZEICHEN UND RTS
0964
0965                      ;***** 1 ZEILE IN TEXTSPEICHER EINFUEGEN
0966 0026CF         MOVBACK                   ;EINE ZEILE IN DEN TEXTSPEICH
                            ER EINFUEGEN
0967 0026CF EE0006        INC ZL_LEN          ;=WIRKLICHE INSERT-MENGE
0968 0026D2 A5F2          LDA BOTLN           ;BEGRENZUNG BEI $FE00
0969 0026D4 C900          CMP #00
0970 0026D6 A5F3          LDA BOTLN+1
0971 0026D8 E9FE          SBC #$FE
0972 0026DA 9008          BCC MOVBACK1
0973 0026DC A0E3          LDY #MES14-MES0     ;SPEICHER VOLL
0974 0026DE 206426        JSR MESOUT
0975 0026E1 4CCF23        JMP READEND1        ;CLOSE FILE
0976
0977 0026E4 A5F3    MOVBACK1 LDA BOTLN+1      ;VORLAEUFIGES TAFELENDE RETTE
                            N
0978 0026E6 8D0A06        STA SYMSAV+1
0979 0026E9 A5F2          LDA BOTLN
0980 0026EB 8D0906        STA SYMSAV
0981 0026EE AE0D06        LDX TEXTBANK        ;MOVE IN BANK 2
0982 0026F1 8601          STX 1650?
0983
0984 0026F3 38      MOVBACK2 SEC
0985 0026F4 A5F2          LDA BOTLN
0986 0026F6 E5F0          SBC NOWLN
0987 0026F8 AB            TAY                 ;ZAEHLER IN DER PAGE
0988 0026F9 A5F3          LDA BOTLN+1         ;BESTIMME ZAHL PAGES
0989 0026FB E5F1          SBC NOWLN+1
0990 0026FD D021          BNE MOVBACK7        ;VERSCHIEBUNG VOLLER SEITEN
0991 0026FF 98            TYA                 ;MENGE IN DER SEITE=0?
0992 002700 D003          BNE MOVBCK2
0993 002702 4CC925        JMP TONOWL          ; JA PUT BEHIND
0994
0995 002705 88      MOVBCK2  DEY              ;Y=$FF
0996 002706 18            CLC
0997 002707 A5F0          LDA NOWLN           ;AB QUELLE ZL_LEN STELLEN REC
                            HTS
0998 002709 6D0006        ADC ZL_LEN          ;LAENGE EINGABEZEILE
0999 00270C 85F2          STA BOTLN           ;TEMP. ZIELPOINTER
1000 00270E A5F1          LDA NOWLN+1
1001 002710 6900          ADC #0
1002 002712 85F3          STA BOTLN+1
1003 002714 B1F0    MOVBACK3 LDA (NOWLN),Y    ;TRANSPORT NUN IN DER PAGE
1004 002716 91F2          STA (BOTLN),Y
1005 002718 88            DEY
1006 002719 D0F9          BNE MOVBACK3
1007
1008 00271B B1F0          LDA (NOWLN),Y       ;NUN DAS BYTE MIT Y=0
1009 00271D 91F2          STA (BOTLN),Y
1010 00271F 60            RTS                 ;UND WEITER NACH TONOWL
1011
1012 002720 8D0B06  MOVBACK7 STA SYMPAG       ;VOLLE PAGES FUER MOVE
1013 002723 C6F3          DEC BOTLN+1
1014 002725 18            CLC
1015 002726 A5F2          LDA BOTLN           ;EMPFANGSPOINTER EINRICHTEN F
                            UER VERSCHIEBUNG
1016 002728 6D0006        ADC ZL_LEN
1017 00272B 85F8          STA SYMDES
1018 00272D A5F3          LDA BOTLN+1
1019 00272F 6900          ADC #0
```

```
1020 002731 85F9            STA SYMDES+1
1021
1022 002733 A0FF      MOVBACK8  LDY #$FF          ;1 VOLLE PAGE MOVEN
1023 002735 B1F2      MOVBACK9  LDA (BOTLN),Y
1024 002737 91F8            STA (SYMDES),Y
1025 002739 88              DEY
1026 00273A D0F9            BNE MOVBACK9
1027
1028 00273C B1F2            LDA (BOTLN),Y       ;DAS UNTERSTE BYTE ZULETZT
1029 00273E 91F8            STA (SYMDES),Y
1030 002740 C6F3            DEC BOTLN+1
1031 002742 C6F9            DEC SYMDES+1
1032 002744 CE0B06          DEC SYMPAG          ;VOLLE PAGES FUER MOVE
1033 002747 D0EA            BNE MOVBACK8        ;WEITERE PAGES MOVEN
1034
1035 002749 E6F3            INC BOTLN+1         ;KORREKTUR, UNTERLAUF
1036 00274B 4CF326          JMP MOVBACK2
1037
1038 00274E 18        KORRBOTLN CLC             ;BOTLN UM ZL_LEN KORRIGIEREN
1039 00274F AD0906          LDA SYMSAV
1040 002752 6D0006          ADC ZL_LEN
1041 002755 85F2            STA BOTLN
1042 002757 AD0A06          LDA SYMSAV+1
1043 00275A 6900            ADC #0
1044 00275C 85F3            STA BOTLN+1
1045 00275E A900            LDA #00
1046 002760 AB              TAY
1047 002761 91F2            STA (BOTLN),Y       ;00-BEGRENZER AN DAS TEXTENDE
1048 002763 4C90E2          JMP PAGRES          ;NORMALER ABSCHLUSS
1049
1050                        ;***** 1 ZEILE IM TEXTSPEICHER ENTFERNEN
1051 002766 A5F1      KILSUB    LDA NOWLN+1       ;ALTEN ZEILENZEIGER RETTEN
1052 002768 8D0A06          STA SYMSAV+1
1053 00276B A5F0            LDA NOWLN
1054 00276D 8D0906          STA SYMSAV
1055
1056 002770 18        KILSUB1   CLC               ;TRANSPORTPOINTER EINRICHTEN
                                MIT
1057 002771 AD0006          LDA ZL_LEN          ;AUFBEWAHRTE ZEILENLAENGE
1058 002774 65F0            ADC NOWLN
1059 002776 85F8            STA MOVPTR
1060 002778 A5F1            LDA NOWLN+1
1061 00277A 6900            ADC #0
1062 00277C 85F9            STA MOVPTR+1
1063 00277E AE0D06          LDX TEXTBANK        ;IN BANK2 ARBEITEN
1064 002781 8601            STX I6509
1065 002783 A000            LDY #0
1066
1067 002785 B1F8      KILS1     LDA (MOVPTR),Y    ;VON RECHTS NACH LINKS
1068 002787 91F0            STA (NOWLN),Y
1069 002789 F00A            BEQ KILEND          ;BEGRENZER GEFUNDEN
1070 00278B C8              INY
1071 00278C D0F7            BNE KILS1
1072 00278E E6F9      KILS2     INC MOVPTR+1      ;ZEIGE AUF NAECHSTE PAGES
1073 002790 E6F1            INC NOWLN+1
1074 002792 4C8527          JMP KILS1
1075
1076 002795 98        KILEND    TYA               ;DERZEITIGER OFFSET
1077 002796 18              CLC                 ;ADRESSE TEXTBEGRENZER NACH B
                                OTLN
1078 002797 65F0            ADC NOWLN
1079 002799 85F2            STA BOTLN
1080 00279B A5F1            LDA NOWLN+1
1081 00279D 6900            ADC #0
1082 00279F 85F3            STA BOTLN+1
1083 0027A1 AD0906          LDA SYMSAV          ;ALTE NOWLN ZURUECK
1084 0027A4 85F0            STA NOWLN
1085 0027A6 AD0A06          LDA SYMSAV+1
1086 0027A9 85F1            STA NOWLN+1
1087 0027AB 60              RTS
1088
1089                        ;SUCHE SUCHSTRING IM BEREICH NOWLN BIS END
1090                        ;NACH BOYER & MOORE, MICRO MAG NR. 42-85
```

```
1095 0027AC AD0306    SUCH      LDA SUCH_LENW
1096                            ;LAENGENEINTRAG IN SUCH_TAB ZUNAECHST M. STRINGLA
ENGE
1097 0027AF 8D0206              STA SUCH_LEN        ;DIESER PARM WIRD SPAETER ZER
                                    STOERT
1098 0027B2 48                  PHA                 ;RESTORE LATER
1099 0027B3 A000                LDY #0
1100 0027B5 998005    SUCH1     STA SUCH_TAB,Y
1101 0027B8 C8                  INY
1102 0027B9 10FA                BPL SUCH1           ;BIS $7F
1103
1104                            ;VERSCHIEBEDISTANZ IN SUCH_TAB
1105 0027BB A200                LDX #0              ;BEGINNE BEI SUCHSTRING
1106 0027BD CE0206              DEC SUCH_LEN
1107 0027C0 AD0206    SUCH2     LDA SUCH_LEN        ;EINTRAG VON STRINGLAENGE-1 B
                                    IS 0
1108 0027C3 BC0005              LDY FINDBUFF,X      ;BUCHSTABE ALS INDEX IN SUCH_
                                    TAB
1109 0027C6 998005              STA SUCH_TAB,Y
1110 0027C9 E8                  INX
1111 0027CA CE0206              DEC SUCH_LEN
1112 0027CD 10F1                BPL SUCH2
1113 0027CF 68                  PLA
1114 0027D0 8D0206              STA SUCH_LEN
1116                            ;BERECHNE SUCH-ENDE
1117 0027D3 A5F2                LDA BOTLN
1118 0027D5 38                  SEC
1119 0027D6 ED0206              SBC SUCH_LEN
1120 0027D9 8D0706              STA END
1121 0027DC A5F3                LDA BOTLN+1
1122 0027DE E900                SBC #0
1123 0027E0 8D0806              STA END+1
1124 0027E3 AC0306    SUCH3     LDY SUCH_LENW       ;BEGINNE VERGLEICH VOM STRING
                                    ENDE
1125 0027E6 88                  DEY
1126 0027E7 AD0D06              LDA TEXTBANK        ;BANKUMSCHALTUNG
1127 0027EA 8501                STA I6509
1128 0027EC A200                LDX #0              ;SCHLUESSELPOSITION
1129 0027EE B1F0      SUCH4     LDA (NOWLN),Y       ;TEXTZEICHEN
1130 0027F0 D90005              CMP FINDBUFF,Y      ;SUCHSTRING
1131 0027F3 D007                BNE SUCH5
1132 0027F5 E8                  INX
1133 0027F6 88                  DEY
1134 0027F7 10F5                BPL SUCH4           ;NOCH NICHT ALLES BYTES
1135 0027F9 4C90E2              JMP PAGRES          ;GEFUNDEN, CARRY=1, BANKUMSCH
                                    ALTUNG, RTS
1136
1137                            ;SUCHSTRING NOCH NICHT GEFUNDEN, NUN SCHIEBEN
1138 0027FC AB        SUCH5     TAY                 ;BUCHSTABE IST INDEX F. SUCH_
                                    TAB
1139 0027FD 8A                  TXA                 ;SCHLUESSELSTELLE
1140 0027FE 49FF                EOR #$FF            ;KOMPLEMENT
1141 002800 38                  SEC                 ;2ER-KOMPLEMENT
1142 002801 798005              ADC SUCH_TAB,Y      ;VERSCHIEBEDISTANZ - SCHLUESS
                                    ELSTELLE
1143 002804 F002                BEQ SUCH6
1144 002806 1002                BPL SUCH7           ;DISTANZ>0
1145 002808 A901      SUCH6     LDA #1
1146 00280A 18        SUCH7     CLC
1147 00280B 65F0                ADC NOWLN
1148 00280D 85F0                STA NOWLN
1149 00280F A5F1                LDA NOWLN+1
1150 002811 6900                ADC #0
1151 002813 85F1                STA NOWLN+1
1152                            ;TESTE AUF ENDE
1153 002815 CD0806              CMP END+1
1154 002818 90C9                BCC SUCH3
1155 00281A D009                BNE SUCH9
1156 00281C A5F0                LDA NOWLN           ;LOW
1157 00281E CD0706              CMP END
1158 002821 90C0                BCC SUCH3
1159 002823 F0BE                BEQ SUCH3
1160 002825 18        SUCH9     CLC                 ;NICHT GEFUNDEN, CARRY=0
1161 002826 4C90E2              JMP PAGRES          ;BANKUMSCHALTUNG UND RTS DORT
```

```
1163                                    ;***** INTERAKTIVE MELDUNGEN
1164 002829 0D12        MES0            .BYT CR,REVERS_ON,'BEFEHLE: '
1164 00282B 42454645
1165 002834 45205420                    .BYT 'E T B U D "CR" Q K I R L F W C A M / S N P
                                             O < & '
1166 002865 9200                        .BYT REVRS_OFF,NULL
1167 002867 12          MES1            .BYT REVERS_ON,'TEXTEINGABE QUIT=ESC/$'
1167 002868 54455854
1168 00287E 0D00                        .BYT CR,NULL
1169 002880 12          MES2            .BYT REVERS_ON,'TOP',CR,CR,NULL
1169 002881 544F50
1169 002884 0D0D00
1170 002887 12          MES3            .BYT REVERS_ON,'END',CR,NULL
1170 002888 454E44
1170 00288B 0D00
1171 00288D 0D12        MES4            .BYT CR,REVERS_ON,'FINDEN',CR,NULL
1171 00288F 46494E44
1171 002895 0D00
1172 002897 12          MES5            .BYT REVERS_ON,'VERAENDERN',CR,CR,NULL
1172 002898 56455241
1172 0028A2 0D0D00
1173 0028A5 12          MES6            .BYT REVERS_ON,'FILENAME=?',CR,CR,NULL
1173 0028A6 46494C45
1173 0028B0 0D0D00
1174 0028B3 12          MES7            .BYT REVERS_ON,'INSERT',CR,CR,NULL
1174 0028B4 494E5345
1174 0028BA 0D0D00
1175 0028BD 0D12        MES8            .BYT CR,REVERS_ON,'LISTEN',CR,NULL
1175 0028BF 4C495354
1175 0028C5 0D00
1176 0028C7 0D12        MES9            .BYT CR,REVERS_ON,'KILLED',CR,NULL
1176 0028C9 4B494C4C
1176 0028CF 0D00
1177 0028D1 0D12        MES10           .BYT CR,REVERS_ON,'BANKSWITCH',CR,NULL
1177 0028D3 42414E4B
1177 0028DD 0D00
1178 0028DF 12          MES11           .BYT REVERS_ON,'BANK = ',NULL
1178 0028E0 42414E4B
1178 0028E7 00
1179 0028E8 12          MES12           .BYT REVERS_ON,'ALLES LOESCHEN? J/CR',00
1179 0028E9 414C4C45
1179 0028FD 00
1180 0028FE 0D12        MES13           .BYT CR,REVERS_ON,'ZEILE NR. ',CR,00
1180 002900 5A45494C
1180 00290A 0D00
1181 00290C 0D12        MES14           .BYT CR,REVERS_ON,'SPEICHER VOLL',CR,00
1181 00290E 53504549
1181 00291B 0D00
1182
1183                                    ;***** ERLAUBTE BEFEHLSTASTEN
1184 00291D 45544255 TASTEN             .BYT 'ETBUD',CR,'QKIRLFWCMA/SNPO<&'
1184 002922 0D
1184 002923 514B4952
1185
1186                                    ;***** SPRUNGVERTEILER FUER BEFEHLE
1187 002934 55201B21 JMPTBL             .WOR TEXT_EIN-1,TOP-1,BOTTOM-1,UP-1,DOWN-1,RETUR
                                             N-1
1187 002938 2C213A21
1187 00293C 66219621
1188 002940 9C219122                    .WOR QUIT-1,KILL-1
1189 002944 A9227323                    .WOR NSERT-1,READ-1,LIST-1,FIND-1,WEIT_SUCH,CHAN
                                             GE-1
1189 002948 E0233124
1189 00294C 7C249024
1190 002950 18250260                    .WOR MONITOR-1,ASSEMBLER-1,BANKSWIT-1,LOESCH-1,N
                                             UMMER-1
1190 002954 C7219D21
1190 002958 4422
1191 00295A 3723D322                    .WOR PRINT-1,READNSERT-1,LIN_NSERT-1,WRIT_FLOP-1
1191 00295E DB22DC22
1192 002962                             .END
FEHLERZAHL INSGESAMT: 0000
```

Roland Löhr

Commodore AMIGA

Dieser neue Rechner soll ab März 1986 auch in Deutschland lieferbar sein. In den USA kostet er etwas unter 2000 Dollar und soll in den ersten drei Wochen bereits 20.000mal verkauft worden sein. Er fußt auf der CPU 68000 mit 7.15909 MHz Taktfrequenz und drei Custom-ICs für Grafik, Animation und Sound. Die Grundausstattung hat 256 KB Arbeitsspeicher, die intern auf 512 und mit externen Modulen auf 8 MB ausgebaut werden können. Die Schnittstellen werden weiter unten besprochen.

Auf der SYSTEMS wurde der neue Computer, dem in der Fachpresse schon höchst lobende Worte vorauseilten, erstmals auch ausgewählten Messebesuchern in Deutschland vorgestellt. Das Lob ist berechtigt. Der AMIGA setzt in der Klasse der gehobenen Personal-Computer deutlich neue Maßstäbe, die sich von den älteren Strukturen in der recht uniformen IBM-Welt unterscheiden und auch von denjenigen, die beim MacIntosh und bei Atari verwirklicht wurden. Bei letzteren finden wir von Anfang an bereits ein Desk Top-Menue, das der Benutzer mit der Maus bedient. So natürlich auch hier. Beim Amiga ist der zentrale Prozessor jedoch durch die Spezialchips vom Bildaufbau entlastet und kann vornehmlich fürRechenleistung, auch im Multi-Tasking, benutzt werden. Bei den verschiedenen eindrucksvollen Demo-Programmen mit bewegter Grafik und stereophonischer Tonausgabe wurde angemerkt, die CPU sei damit nur zu etwa 8% ausgelastet. Wie dem auch sei, es scheinen erhebliche Leistungsreserven im Rechner vorgesehen zu sein.

Bei den in diesem Jahre neuen Personal Computern wird weniger herausgestellt, was sie an Rechenleistung bieten, als das, was mit ihnen zusätzlich möglich ist, vor allem in der Grafik und bei der Ton- und Musikerzeugung. Ein Datenblatt liegt dem Autor hierzu auch noch nicht vor, so daß er nur erwähnen kann, daß während der Demonstration auch Disketten mit BASIC und der Sprache C nebst Assembler auf dem Tisch lagen, ferner auch ein Textsystem, das kurz in Betrieb gebracht wurde und Ähnlichkeiten mit anderen aufweist, die z.B. auf dem IBM-PC zum Einsatz kommen, mit Formatierungsmöglichkeiten, Lineal usw. Das Textsystem, die Tastaturbelegung und die Zeichendarstellung sollen bis zur Markteinführung auf deutsche Bedürfnisse zugeschnitten werden.

Die graphischen Fähigkeiten des AMIGA sind in dieser Klasse beachtlich. Sie werden von zwei Chips besorgt. Der Graphik-Chip erzeugt das Monitorbild mit einem oder mehreren Bildern (Screens), die in verschiedenen Maßstäben und Ausschnitten zugleich auf dem Bildschirm ausgegeben werden, so daß man in verschiedene laufende Prozesse oder Files Einblick nehmen kann. Dieser Grafik-Chip und auch derjenige für den Ton werden von einem Animations-Chip gesteuert, der wiederum zwei Funktionsgruppen hat, den 'Copper' (wahrscheinlich ist das ein Druckfehler, der zu 'Chopper' berichtigt werden müßte) und den Blitter. Ersterer steuert den Graphik-Chip je nach der Lage des erreichten Bildpunktes, der Blitter hingegen zeichnet Linien und Kästen, füllt Figuren mit Farbe aus und verschiebt Bildteile bzw. Daten im Speicher. Und das Ganze natürlich in Farbe mit der wirklich hohen und ruhig stehenden Auflösung von bis zu 640x400 Bildpunkten. Je nach Auflösung sind bis zu 4096 Farbtöne bzw. 16 Farben möglich. - Im hochauflösenden Modus wurden Tierbilder auf dem Monitor erzeugt, die in der Qualität an ein Farbfoto heranreichten.

In den vorläufig verteilten Informationen zum Computer tauchen Begriffe auf, die noch nicht im einzelnen erklärt worden sind, wie Hold-and-Modify-Modus, Dual-Playfield-Mode (Blitter). Ferner wird gesprochen von einem problemlosen Handling der Sprites und Shapes. Wie in der Demonstration z.T. gezeigt und erklärt wurde: Man kann verschiedene Bildebenen wählen (Vorder-, Hinter- und wohl auch Mittelgrund). Es können digitalisierte Bilder einer Video-Kamera mit der Graphik und den Ausgaben aus dem Computer gemischt werden. Ein digitalisiertes Bild läßt sich graphisch nachbearbeiten in den Konturen und in der Farbgebung, so daß bildliche Produkte entstehen, wie man sie in der Werbung findet.

Die graphischen Fähigkeiten des Amiga zielen damit auf folgende typischen Anwendergruppen: Freiberufler im graphischen Fach, Planungs- und Entwurfsbüros, ggfs. den CAD-Sektor, Künstler Schulen und Bildungssektor. Daneben ist eine graphische Aufbereitung und Präsentation von Rechenergebnissen im kaufmännischen und technischen Sektor möglich, wie schon mit der vorhergehenden Generation der Personal Computer.

Auch die Fähigkeiten bei Geräusch und Musik sind auf eine neue Basis gestellt worden. Es ist eine gut verständliche Sprachsynthese in mehreren Stimmlagen (Mann, Frau usw.) möglich. Für den Sound steht ein vierstimmiger Synthesizer zur Verfügung, der stereophon ausgibt und der nach dem Sampling-Verfahren arbeitet. Frühere Sound-Chips arbeiteten im wesentlichen mit dem reinen Ton, der Wellenform, der Hüllkurve und mit den An- und Abschwellen, natürlich auch in der Mischung mehrerer Generatoren. Beim AMIGA ist offensichtlich zusätzlich noch das typische Klangspektrum von etwa 10 Musikinstrumenten gespeichert, das neben den Tönen und den eingeschalteten Filtern und Formern abgetastet wird, so daß sich die Ausgabe sehr instrumenten-nah anhört. Auch hier kann mit externen Schallquellen gemischt werden, so daß man im Laufe der Zeit beeindruckende Schöpfungen erwarten darf, auch im Zusammenhang mit bewegter Farbgraphik.

Schnittstellen: Ein 3,5-Zoll-Laufwerk ist integriert. Es gibt einen Port für RAM-Erweiterung und den Anschluß z.B. einer Harddisk oder andere Peripherie, eine Schnittstelle für externe Floppy und Drucker (Centronics), sowie für RS 232C.

Wo es nun interessant wird: Neben dem Amiga-Betriebssystem soll fast jedes beliebige Betriebssystem geladen werden können, z.B. MS-DOS oder UNIX (das große Speicher braucht). Wenn diese Aussage zutrifft, dann kann der Amiga auch als IBM-PC mit seinen vielen Programmen gefahren werden und öffnet die Welt des UNIX, das im professionellen Bereich immer größere Bedeutung erlangt. Vorgeführt wurde jedenfalls schon das Multitasking, der Ablauf verschiedener Prozesse im zeitlichen Versatz, versehen auch mit Prioritäten.

Die vorerwähnten Merkmale des AMIGA klingen sehr gut und vielversprechend. Man sollte sie kritisch prüfen, sobald auf den ersten ausgelieferten Systemen Demonstrationen und Versuche gefahren werden können und natürlich auch den Bedienungskomfort zur Erzielung solcher Leistungen, unterstützt von Menues und der Maus. Sicher geht man in der Aussage, daß der AMIGA in seiner Preisklasse viele neue Möglichkeiten bietet und daß diese Möglichkeiten viele interessante Programmangebote der Softwarehäuser nach sich ziehen werden, die dann dem Publikum zur Verfügung stehen.

Roland Löhr

Atari 260 und 520+

Im Lieferangebot der Firma Atari sind mit Beginn der SYSTEMS 85 Erweiterungen und Veränderungen eingetreten. Neu ist der ST520+ mit 1 MB Arbeitsspeicher zum Preise von DM 2.998,-. Er wird vom Hersteller 'Mega-Atari' genannt. Zum Lieferumfang gehört der Monochrom-Monitor SM 124 und die Floppy SF 354. Er hat folgende Schnittstellen: V24/RS232 für Modem oder Drucker, parallel für Drucker, Midi für Synthesizer, DMA für Festplatte. Anschließbar sind außerdem zwei Floppys 3 1/2 Zoll mit 320 oder sogar 720 KB. - Dieses Modell ist damit in erster Linie ein Angebot für professionelle Anwender. Es ist ab sofort beim Handel verfügbar, die ersten 1000 Geräte wurden in der 44: Woche (Messebeginn) bereits in den Handel geleitet. - Dem Leser wird auffallen, daß dieses Gerät bei doppeltem Arbeitsspeicher den Preis des bisherigen Modells 520 ST hat. Damit wurde der günstigere Dollar-Kurs und die Verbilligung der Bauteile an den Endverbraucher weitergegeben.

Neu ist das Modell 260ST, das an die Stelle des bisherigen 520ST treten dürfte. Es hat wie dieser

512 KB Arbeitsspeicher und kann 'entbündelt' gekauft werden: Die Computer-Konsole kostet DM 1.298,-. Man kann nun einen vorhandenen Monitor anschließen oder Floppy (DM 598,-) und Maus (DM 148,-) nach Bedarf nachrüsten und erhält damit ab etwa DM 2.000,- ein funktionsfähiges System. - Atari nennt den 260ST damit ein Modell für 'Aufsteiger'.

Alle Modelle sollen untereinander software-kompatibel sein. Ab November 1985 soll für diese ferner ein CP/M-Z80 Emulator verfügbar sein, der die ST in einen mit 2 MHz getakteten Z80-Recher verwandelt, auf dem die gesamte CP/M 2.2-Software lauffähig ist. Damit hat man Zugriff auf viele bekannte Software, allerdings in der Emulation eben nur mit entsprechend 2 MHZ Takt.

Interessant waren Ausführungen, die Atari anläßlich einer Pressekonferenz im Zuge der SYSTEMS machte. Man hatte die deutsche Geschäftsführung (A. Stumpf) und die internationale vor sich: S. Tramiel, S. Hartmann (Software) und S. Shivji (Entwicklung). Es wurde deutlich, daß man an noch leistungsstärkeren Maschinen arbeitet: Echte 32 Bit, mit denen man viermal schneller werden und software-kompatibel bleiben möchte. Anläßlich der Hannover-Messe wird man dazu wohl schon konkreteres hören können.

Zur Software berichtete Sigmund Hartmann, daß bis Jahresende etwa 200 Software-Produkte benutzbar sein sollen, Productivity-Pakete, Textverarbeitung, Spreadsheets, Data Bases, Ausbildung, Networking. Multitasking für Computer mit 1-2 MB soll in Hannover vorgestellt werden. Insgesamt hofft man auf einen Zustrom weiterer Software, nachdem in den USA 1000, in England 250 und in Deutschland 300 Software Development Kits verkauft wurden. - Bei den Ankündigungen ist sicher manches noch Hoffnung, und die Termine mögen sich verspäten, da die Entwicklung guter Software doch immer sehr arbeitsaufwendig ist. - Im laufenden Firmenprospekt sind folgende Software-Produkte enthalten: ST Manager für Kunden, Rechnung, Lager, Text und PASCAL.

Für die ST-Computer sollen durch die englische Firma Electric Software Ltd., 91 High Street, Longstanton, Cambridge, ab November folgende Software-Pakete lieferbar sein: GSTC C-Compiler mit GM Window Support, 68000 Macro Assembler und GST Editit GEM Screen Editor. Die Prese sind mit 60, 40 bzw. 25 engl. Pfund durchaus mäßig.

Roland Löhr

Multitech Popular 500

Der IBM-PC ist auch für diesen preiswerten Rechner Vorbild gewesen. Er wird in Taiwan hergestellt, von der Ce-Tec in Hamburg importiert und über Händler ausgeliefert.

Der Popular 500 enthält die CPU 8088, hat 128 KB Arbeitsspeicher und ist bereits im Grundausbau mit einer Farb-Graphikkarte sowie mit paralleler und RS232-Schnittstelle ausgerüstet. Wenn auch die bei IBM übliche Aufrüstbarkeit und die Kompatiblität mit dem zum Lieferumfang gehörenden MS-DOS 2.11 gegeben ist, so sollte man beachten, daß im Endverbraucherpreis von DM 2798,- wohl ein monochromer Bildschirm-Monitor (Sanyo 4112) enthalten ist, jedoch nur ein Laufwerk. Allerdings kann man für DM 548,- ein zweites Laufwerk kaufen und problemlos anschließen, so daß man für erforderliche Sicherheitskopien ein bequemeres Arbeiten hat. Je Laufwerk werden die üblichen 360 KB Daten gespeichert.

Der Herausgeber hatte die Gegenheit, den Popular 500 während eines kurzen Wochenendes zu testen und mit monochromer Bildausgabe zu betreiben. Angenehm fiel die in vier Felder gegliederte große Tastatur auf: Links die üblichen 10 Funktionstasten, dann das alphabetische Schreibmaschinenfeld, bei dem die Tasten doppelt mit ihrer ASCII- und mit ihrer deutschen Belegung beschriftet sind, dann ein Cursor-Kontrollblock zum Editieren und rechts schließlich das Feld für Zahleneingaben. Die Tastatur ist 55 cm breit, so daß die Tasten bequem zu erreichen sind. Gewöhnungsbedürftig ist (wie beim IBM) die Lage der ALT-Taste (Zeichensatzumschaltung). Man drückt sie oft versehentlich, wenn man die Shift-Taste erreichen wollte.

Bei der Bildschirmausgabe hat man ein klares Bild und hat von Anfang an den deutschen Zeichensatz. Beim Lieferanten konnte auch die Ausgabe auf einen Farbbildschirm beobachtet werden, die ebenfalls ein klares und ruhiges Bild brachte. - Der Rechner teilt mit dem Original von IBM das Merkmal, daß der Lüfter lauter ist, als technisch machbar, wenn man nur einige Mark mehr ausgeben würde.

Von den vielen Programmen, die auf Diskette zur Verfügung standen, konnten in der Kürze nur wenige erprobt werden. Das Booten des Betriebssystems und das Laden von Programmen erfolgte ohne Störungen. In Betrieb genommen wurden der wirklich schnell arbeitende Turbo PASCAL-Compiler, BASIC, verschiedene graphische Demonstrationsprogramme, der Debugger sowie das mit einer 131-seitigen deutschen Betriebsanleitung versehene Textprogramm PRACTIWORD, das ähnliche Merkmale wie z.B. Wordstar hat.

Zum Rechner gehört eine kleine englische Dokumentation (User's Guide) sowie eine ausführliche von ca. 180 Seiten. Als Lizenznehmer des DOS hat der Hersteller Multitech ferner das englische 'Microsoft MS-DOS User's Guide' nachgedruckt. In diesem Buch findet man alle Befehle des Betriebssystems und die von ihnen geforderten Parameter.

Als Ergebnis der Erprobung ist zu sagen, daß mit diesem Rechner ein sehr preiswerter Einstieg in die Welt des IBM-PC möglich ist. Dem Benutzer stehen hier viele Erweiterungsmöglichkeiten und eine Unzahl von Programmen zur Verfügung. - Info: CE-TEC International, Lange Reihe 29, 2000 Hamburg 1, T. 040-280 10 45.

Roland Löhr

MELPS 40

Einchipper mit 65xx-Befehlssatz

Weniger bekannt als das breite Spektrum von Einchippern bei Motorola dürfte die MELPS 740-Serie von Mitsubishi sein. Es handelt sich um Ein-Chip-Mikrocomputer, die den Standard-Befehlssatz des 6502 als Teilmenge enthalten. Die Zahl der Befehle und Adressierungsarten wurde bei ihnen in einer interessanten Weise erweitert, die einerseits sparsamen Code und andererseits weitaus mehr Möglichkeiten der Bitmanipulation und der Verzweigung in Abhängigkeit von Bit-Bedingungen haben, als z.B. die CMOS-CPU 65C02 von Rockwell. Die verschiedenen Zentraleinheiten sind typisch für das Interfacing ausgelegt und haben 32 bis 56 Interface-Pins für Ports zur Umwelt sowie 3 Timer von je 8 Bits. Die Chips sind in CMOS hergestellt und können meistens bis 4 MHz betrieben werden. Der Typ M50747 ist sogar mit 8 MHz spezifiziert, womit sich sehr hohe Verarbeitungsgeschwindigkeiten ergeben.

Die einzelnen Typen, deren Bezeichnung M50xxx ist, unterscheiden sich durch die Zahl der Gehäusepins (52-64), durch die Größe des Festwertspeichers, des RAM On Chip und durch Attribute wie A/D-Konverter und 35V Treiberfähigkeit für Fluoreszenz-Displays. Für die meisten Typen gibt es Evaluation Chips, die auf dem CPU-Gehäuse ein EPROM tragen können (Piggy Pack).

Festwertspeicher heißt hier meistens: Maskenprogrammiertes ROM. Der Entwickler einer kleinen Serie mag bei diesem Wort zuerst erschreckt sein, weil sowas für ihn nicht in Betracht kommt. Beim Studium der Datenblätter stellt man jedoch fest, daß sich die Japaner drei Betriebsweisen haben einfallen lassen, die elektrisch durch 0, 5 oder 10 Volt am Pin CNV erzwungen und nach dem Reset auch noch durch Beschreiben des Registers MODE verändert werden können, so daß sich auch dynamisch eine veränderliche Konfiguration ergibt.

Im Single Chip Mode sind alle Ports wie üblich nur Ports. Im EVA Mode wird das interne ROM ausgeblendet und es erscheinen an den Portpins Adreß- und Datenbus-Signale, und zwar in der Prozessorphase 1. In Phase 2 sind die Pins dann wieder Ports. Dieses Multiplexing verlangt dann natürlich externe Fanggatter (Latches), um einen störungsfreien Betrieb zu gewährleisten. Aber immerhin stehen dann je nach Typ 8 KB bis 64 KB externer Speicherraum zur Verfügung. Im dritten und vierten Modus (memory Expanding und Microprocessor Mode) verlieren die meisten

Portpins ihre I/O-Eigenschaft und dienen nur als Daten- und Adreßbus bzw. für die Signale R/W und SYNC. So sollte es möglich sein, trotz eines vorhandenen ROMs, dieses auszublenden und die Prozessoren mit einem Betriebsprogramm des Anwenders zu versehen. Angekündigt ist auch ein Typ M50734 ohne ROM und mit 32 Port-Pins sowie 16 Bit Adreßbus, der in manchen Konfigurationen eine einfachere externe Beschaltung erlaubt.

Bei den meisten Vertretern der Famile finden wir mit R0 bis R3 bezeichnete Pins (Register Select). An sie kann direkt mit dem auch vorhandenen Signal CE ein Port-Expander vom Typ M50791 angeschlossen werden, der um weitere 10 Portpins zu je 4 Bit erweitert, womit dann ein noch umfangreicheres Interfacing möglich wird, bzw. man kann die Multiplex-Beschaltung an den Portpins des Prozessors geringer auslegen.

Die Timer nebst Prescalern können auch als Ereigniszähler benutzt werden. Auch eine Ausgabe von Rechteckimpulsen ist möglich. Insgesamt sind fünf Interrupt-Quellen zugelassen, von denen drei den Timern zugeordnet sind.

Die Einzelheiten der Auslegung und des möglichen Betriebes entnehme man den Datenblättern und Unterlagen des Herstellers. Nach deren Durchsicht ist klar zu erkennen, daß Mitsubishi viele neue Möglichkeiten in die 65xx-Familie trägt, die geradezu zu Experimenten ermutigen, zumal für etliche Typen auch der serielle Datenverkehr mit einem Terminal oder Host leicht zu implementieren ist. Interessant sind auch die neuen Befehle und Adressierungsarten, auf die wir jetzt eingehen:

Die Erweiterung des Befehlssatzes hat bei MELPS 740 Ähnlichkeiten mit derjenigen bei der Rockwell-CPU 65C02, geht aber darüber hinaus. Auch ist zu betonen, daß gleichlautende und gleichwirkende Befehle einen anderen hexadezimalen Opcode haben, was bei Assemblierungen zu berücksichtigen ist.

BRA	Hexcode 80, verzweige immer, 2 Byte langer Befehl wie bei 65C02 und 65816.
BBS	Branch On Bit Set und
BBC	Branch on Bit Clear (Reset), Wirkung wie bei 65C02, Programmverzweigung, wenn ein Bit im Bereich der Zero Page 0 oder 1 ist. Erweiterung bei MELPS 740: BBS und BBC, wenn ein Bit im Akku 0 oder 1 ist. Diese neue Adressierungsart wirkt wie zusammengesetzt BIT # mit anschließendem Verzweigungsbefehl. Elegant für viele Entscheidungen.
SEB	Setze Bit Nr. xx in einer Zero Page-Adresse (Interface/Register).
CLB	Lösche ein Bit ebendort. Erweiterung bei MELPS 740: SEB und CLB auf den Akku wirkend.
DEC A	Dekrementiere Akku. Wie bei 65C02 und 65816.
INC A	Inkrementiere Akku, wie vor.
TST	Abfrage einer Zelle in der Zero Page, Setzen der Flags N und Z im Status, abhängig vom Inhalt der Zelle.
COM	Bildung des Einer-Komplements in einer Zelle der Zero Page. Der Inhalt des Akkus bleibt unverändert.
LDM #	Laden einer Zelle in der Zero Page mit einem Direktwert, der im Befehl steht. Der Akku wird nicht verändert. Wie schon der Befehl STZ, der eine Null in den Speicher schreibt, so schreibt LDM jeden beliebigen Wert in den Speicher und kürzt bei einer Länge von 3 Bytes die übliche Sequenz LDA Direktwert, STA Speicher etwas ab.
SLW	und FST (Slow, Fast) sind nicht überall implementiert. Sie wirken auf die Ausgabe der Clock am Pin XoutF.
STP	(Stop) hält den Oszillator an (Sleep), der durch Reset oder Interrupt wieder gestartet wird.
JMP	(zp). Ein verkürzter JMP-Befehl von 2 Bytes Länge. Die indirekte Adresse steht in der Zero Page.
JSR	(zp). Indirekter Sprung auf ein Unterprogramm, sonst wie bei JMP (zp).

JSR ¥ Sprung auf Unterprogramm, das in einer 'Special Page' steht, die bei $1F00 beginnt. Auch hier ein Befehl mit 2 Bytes und damit verkürzter Adressierung.
SET Setzt das T-Bit im Status.
CLT Löscht das nämliche Bit. Auswirkung auf Register X.

Im Zusammenhang mit den Befehlen SET und CLT ist eine Sonderrolle zu sehen, in die das Indexregister X hereingerufen werden kann. Zunächst: Das T-Bit steht an Bitposition 5 im Status, die ja beim 6502 nicht benutzt worden ist. Wenn T=0, dann haben wir das übliche Funktionieren des Indexregisters X. Wenn T=1, dann wird bei den arithmetischen und logischen Operationen der Akku nicht berührt. Die Befehle wirken vom Speicher aus auf den Speicher, sind in ihrer Wirkung also z.B. mit ROL Adresse usw. zu vergleichen. Das Register X zeigt dann auf eine Zelle in der Zero Page, in der ein 1. Operand steht. Die Adresse im Befehl bezeichnet einen 2. Operanden M2, in dem auch das Ergebnis der arithmetischen oder logischen Operation abgelegt wird. In Worten: Nimm den Inhalt von M2, addiere den Inhalt der Zelle in der Zero Page, auf die X weist, und lege das Ergebnis bei M2 ab. Addition Speicher zu Speicher, der Akku wird nicht verändert.

Zusammenfassend: Die vorstehenden Beschreibungen zeigen, daß auch die 8-Bit Prozessoren der Familie 65xx weiterentwickelt werden. Es sind neue Konfigurations- und Lösungsmöglichkeiten in der Hardware und effiziente Befehlsfolgen in der Software angeboten. Hinzu kommt ein schneller Prozessortakt von 4 bis 8 MHz, der auf vielen Anwendungen einmal wieder das sehr gute Abschneiden der 65xx in Benchmarks untermauern dürfte. Als Designer sollte man bedenken, daß die üblichen Interfacebausteine der Familie nicht für die hohen Geschwindigkeiten ausgelegt sind. Daher wird sich aus Gründen des Timings empfehlen, für die Ein- und Ausgaben die Ports der Zentraleinheit zu benutzen - und nicht externe Bausteine.

□○

Wolfgang Radeloff, 2080 Pinneberg

PL/65

Anpassung und Erweiterung

Das SPS von Rockwell für den AIM 65 (Software Preparation System) wurde bereits in Heft 39 des MICRO MAG vorgestellt. Heft 40 enthielt dann eine Anpassung der Sprache PL/65 an dieses Softwarepaket. Nachfolgend wird in drei Abschnitten nun eine verbesserte Anpassung dieser Sprache dargestellt, es werden Erfahrungen mitgeteilt und schließlich werden Utility-Programme erklärt und abgedruckt, die das Arbeiten mit PL/65 wesentlich erleichtern.

1. Anpassung und Erweiterung

Es wird vorausgesetzt, daß der Anwender PL/65 in das RAM laden kann. Das nachfolgende Programm überlagert und verändert dann Teile des PL/65. Man assembliert also für oder in den Bereich ab Adresse $B000, um ein neues PL/65 zu erhalten.

Bei der Durchsicht dieses Softwarepaketes stellt man interessanterweise fest, daß PL/65 selbst mit einem PL/65-Compiler geschrieben worden ist. Dadurch ergibt sich die Möglichkeit, verschiedene Routinen der Sprache neu und platzsparender zu formulieren. Wenn dabei dann der erweiterte Befehlssatz der 65Cxx-Prozessoren eingesetzt wird, kann ausreichend Platz für Erweiterungen geschaffen werden. Der Speicherbereich von hex B000-B0c8 wurde so bearbeitet, und es wurden dabei folgende Änderungen und Erweiterungen vorgenommen (siehe auch Programm-Liste):

1. Ein Bufferbereich wurde verlegt, um Konflikte mit dem CRT-Controller im RM-65-Modul zu vermeiden. Es wurde der Bereich ab $0366 gewählt (Zeilenpuffer des CRT-Controllers).
2. Jede Ausgabe in den Speicher wird mit EOF (00) abgeschlossen, um Files im Format des Text-Editors zu erzeugen. Mit OUT=U/DEVICE NO.=4 (Befehl des SPS) wird der von PL/65 erzeugte Assembler-Quelltext in den Textspeicher gelenkt und mit dem Befehl (U)/FUNCTION=7 wird der Texteditor mitgeschaltet. Dadurch ist ein beliebiges Umschalten zwischen den Quelltexten des PL/65 und des Assemblers möglich.

3. Es wurde ein 3. Pass eingerichtet. Er wird mit der Taste (5) und PASS=2 erreicht. Es erfolgt dann eine Compilierung ohne die Ausgabe von PL-Statements. Mit (5) PASS=1 und mit (6) PASS=1 wird weiterhin Pass 1 eingeschaltet. (6) PASS=2 erzeugt als Befehl gegeben Compilertext mit PL/65-Statements, die man zur Fehlersuche und zur Dokumentation braucht.

```
(5) PASS=1          (5) PASS=2          (6) PASS=2
(6) PASS=1

;DCL ALPHA,BETA;    ALPHA *=*+1         ;DCL ALPHA,BETA;
;ALPHA = BETA+5;    BETA *=*+1          ALPHA *=*+1
                    LDA BETA            BETA *=*+1
                    CLC                 ;ALPHA = BETA+5;
                    ADC #5              LDA BETA
                    STA ALPHA           CLC
                                        ADC #5
                                        STA ALPHA
```

In Pass 3 müssen Kommentare in Assembler-Form gegeben werden: ';Kommentar';.

4. PL/65 meldet sich nach den Änderungen mit einem neuen Text: SPS – PL/65 V2.1.
5. Die Darstellung der Fehlermeldungen wurde an die Art des SPS-Assemblers angepaßt. Statt '???????? ?ERROR' wird nun '--------?ERROR' ausgegeben.
6. Fehler führten bisher zu einem Abbruch des Compilierens über einen Interrupt mit BRK. Fehler werden jetzt mit einem entsprechenden Text gemeldet. Danach erfolgt die Rückkehr in das Monitorprogramm über COMIN. Fehler wurden bisher mit den Zahlen 0 bis 3 gemeldet. Stattdessen haben wir jetzt:

0 DO/END? Eine ungerade Anzahl von DO/BEGIN und END wurde festgestellt. Das PL/65-Programm ist nicht mit EXIT; abgeschlossen.

1 INDEX? Es wurde eine unzulässige Index-Schreibweise gefunden. Es werden mehr als drei FOR-TO-Schleifen verschachtelt.

2 BRACKET? Ein Ausdruck kann wegen falscher Klammersetzung nicht aufgelöst werden.

3 LINE? Die Länge einer Ein- oder Ausgabezeile überschreitet 80 Zeichen. (TAB prüfen).

7. Mit .DFILE können Folgedateien mit Quelltext in PL/65 aufgerufen werden. Dazu ist ein Vektor auf die Disk-Inputfunktion des DOS unter F1 zu hinterlegen. Dies wurde geändert, da das SPS diesen Vektor unter 'User Input 0' initialisiert. Der Eintrag unter F1 entfällt jetzt.
8. Mit ENTRY wird als Defaultwert der Start des generierten Programmes bei $0200 festgelegt. Entsprechend der Speichereinteilung im SPS ist der Wert jetzt auf hex 0800 festgesetzt.

```
;ENTRY;                 ;ENTRY;
 *=$0200                *=$0800
R0=0                    R0=0
R1=R0+2                 R1=R0+2
...                     ...
```

```
LINE  ADDR  OBJECT        SOURCE

0001  0000                .OPT NOC,NOS
0002  0000               ;****************************************
0003  0000               ;** ANPASSUNG PL65 AN DAS SPS SYSTEM    **
0004  0000               ;****************************************
0005  0000               ;
```

```
0006  0000              VAR1   =$0366          ;PL85, NEU
0007  0000              BUFF5  =$0367          ;INS/DEL BUFFER CRTC, PL65, NEU
0008  0000              OUTFLG =$A413          ;MON
0009  0000              IRQV2  =$A404          ;MON
0010  0000              TEMP   =$05F3          ;SPS DEVICE NR., SPS
0011  0000              PASS   =$1D            ;PL65
0012  0000              PASS3  =$21            ;PL65, NEU
0013  0000              SEMFLG =$22            ;PL65, NEU
0014  0000              ;
0015  0000              WHEREI =$E848
0016  0000              WHEREO =$E871
0017  0000              BLANK2 =$E83B
0018  0000              DU11   =$E50A
0019  0000              NUMA   =$EA46
0020  0000              TOPNO  =$F8CE          ;SPS
0021  0000              OUTPUT =$E97A
0022  0000              OUTALL =$E9BC
0023  0000              REDOUT =$E973
0024  0000              HEX    =$EA7D
0025  0000              CRLOW  =$EA13
0026  0000              COMIN  =$E1A1
0027  0000              ;
0028  0000              CR     =$0D
0029  0000              LF     =$0A
0030  0000              ;
0031  0000              ;1. BUFFER AB $02A0 VERLEGEN
0032  0000              ;
0033  0000                     *=$B2ED
0034  B2ED  8D 66 03           STA VAR1
0035  B2F0                     *=$B6F1
0036  B6F1  AD 66 03           LDA VAR1
0037  B6F4              ;
0038  B6F4                     *=$B4F4
0039  B4F4  99 67 03           STA BUFF5,Y
0040  B4F7                     *=$B52B
0041  B52B  99 67 03           STA BUFF5,Y
0042  B52E                     *=$B54F
0043  B54F  99 67 03           STA BUFF5,Y
0044  B552                     *=$B571
0045  B571  99 67 03           STA BUFF5,Y
0046  B574                     *=$B590
0047  B590  BD 67 03           LDA BUFF5,X
0048  B593                     *=$B5A4
0049  B5A4  BD 67 03           LDA BUFF5,X
0050  B5A7              ;
0051  B5A7              ;2. PL SOURCE WIRD IM PASS3 UNTERDRUECKT
0052  B5A7              ;
0053  B5A7                     *=$B76A
0054  B76A  4C B0 B0           JMP PTCH30
0055  B76D                     *=$B190
0056  B190  20 E9 CF           JSR PTCH31
0057  B193              ;
0058  B193              ;3. PL/65 MELDET SICH NUN MIT:
0059  B193              ;
0060  B193                     *=$CFB0
0061  CFB0  03          INIMES .BYT 3,CR,LF,'SPS - PL/65 V2.1
0062  CFC5  20 20 50           .BYT '  PASS(1.2) =',0
0064  CFD3              ;4. IM ANSCHLUSS: DIE KORREKTUR-ROUTINEN:
0065  CFD3              ;
```

```
0066  CFD3  AD 13 A4   PATCH2 LDA OUTFLG
0067  CFD6  C9 55             CMP #'U'
0068  CFD8  D0 0A             BNE CRT
0069  CFDA  AD F3 05          LDA TEMP
0070  CFDD  C9 04             CMP #$04
0071  CFDF  D0 03             BNE CRT
0072  CFE1  A9 00             LDA #0
0073  CFE3  2C                .BYT $2C
0074  CFE4  A9 0D       CRT   LDA #CR
0075  CFE6  4C B0 B0    OUT   JMP PTCH30
0076  CFE9                    ;
0077  CFE9  24 21       PTCH31 BIT PASS3
0078  CFEB  10 08             BPL P31E1
0079  CFED  48                PHA
0080  CFEE  A5 22             LDA SEMFLG
0081  CFF0  C9 3B             CMP #';'
0082  CFF2  F0 04             BEQ P31E2
0083  CFF4  68                PLA
0084  CFF5  4C BC E9    P31E1 JMP OUTALL
0085  CFF8  68          P31E2 PLA
0086  CFF9  60                RTS
0087  CFFA                    ;
0088  CFFA                    ;5. MAIN ROUTINE NEU
0089  CFFA                    ;(5) ENTRY PASS2 MIT PL-SOURCE
0090  CFFA                    ;(6) ENTRY PASS2 OHNE PL-SOURCE
0091  CFFA                    ;
0092  CFFA                          *=$B000
0093  B000                    ;
0094  B000  A9 FF       FUENF LDA #$FF
0095  B002  2C                .BYT $2C
0096  B003  A9 00       SECHS LDA #$00
0097  B005  85 21             STA PASS3
0098  B007  A2 FF             LDX #$FF
0099  B009  9A                TXS
0100  B00A  A9 B0             LDA #<INIMES
0101  B00C  A2 CF             LDX #>INIMES
0102  B00E  20 35 B0          JSR PRINT
0103  B011  20 73 E9          JSR REDOUT
0104  B014  C9 0D             CMP #$0D
0105  B016  D0 02             BNE ENTR1
0106  B018  A9 32             LDA #'2'
0107  B01A  20 09 B9    ENTR1 JSR SETPA
0108  B01D  20 3B E8          JSR BLANK2
0109  B020  A9 00             LDA #0
0110  B022  85 1E             STA $1E
0111  B024  20 CE F8          JSR TOPNO
0112  B027  20 48 E8          JSR WHEREI
0113  B02A  20 3B E8          JSR BLANK2
0114  B02D  20 71 E8          JSR WHERED
0115  B030  20 13 EA          JSR CRLOW
0116  B033  80 0F             BRA MAIN
0117  B035                    ;
0118  B035  85 06       PRINT STA $06
0119  B037  86 07             STX $07
0120  B039  B2 06       LP    LDA ($06)
0121  B03B  F0 2C             BEQ END
0122  B03D  20 7A E9          JSR OUTPUT
0123  B040  E6 06             INC $06
0124  B042  80 F5             BRA LP
```

```
0125  B044                  ;
0126  B044  20 72 B0   MAIN  JSR $B072
0128  B047  20 D3 CF   EXIT  JSR PATCH2
0129  B04A  20 0A E5         JSR DUI1
0130  B04D  A9 54            LDA #$54
0131  B04F  8D 04 A4         STA IRQV2
0132  B052  A9 E1            LDA #$E1
0133  B054  8D 05 A4         STA IRQV2+1
0134  B057  A9 46            LDA #<EXMES
0135  B059  A2 B8            LDX #>EXMES
0136  B05B  20 35 B0         JSR PRINT
0137  B05E  A5 0D            LDA $0D
0138  B060  20 46 EA         JSR NUMA
0139  B063  20 13 EA         JSR CRLOW
0140  B066  4C A1 E1         JMP COMIN
0141  B069  60           END  RTS
0142  B06A                  ;
0143  B06A  35           ERMES .BYT <M1,<M2,<M3,<M4
0144  B06E  FF                 .BYT $FF,$FF,$FF,$FF
0145  B072                  ;
0146  B072  B8                CLV
0147  B073  A9 FF             LDA #$FF
0148  B075  85 04             STA $04
0149  B077  85 0C             STA $0C
0150  B079  64 1F             STZ $1F
0151  B07B  64 02             STZ $02
0152  B07D  64 03             STZ $03
0153  B07F  64 0D             STZ $0D
0154  B081  A9 4F             LDA #$4F
0155  B083  85 01             STA $01
0156  B085  A8                TAY
0157  B086  A9 E1             LDA #$E1
0158  B088  8D 04 A4         STA IRQV2
0159  B08B  A9 B6             LDA #$B6
0160  B08D  8D 05 A4         STA IRQV2+1
0161  B090  A9 99             LDA #$99
0162  B092  85 0A             STA $0A
0163  B094  85 0B             STA $0B
0164  B096  A9 5A             LDA #$5A
0165  B098  85 12             STA $12
0166  B09A  85 13             STA $13
0167  B09C  A9 20             LDA #$20
0168  B09E  85 18             STA $18
0169  B0A0  99 00 04   CLRBUF STA $0400,Y
0170  B0A3  99 50 04         STA $0450,Y
0171  B0A6  88                DEY
0172  B0A7  10 F7             BPL CLRBUF
0173  B0A9  20 B3 B6         JSR $B6B3
0174  B0AC  20 55 B9         JSR $B955
0175  B0AF  60               RTS
0176  B0B0                  ;
0177  B0B0  24 21      PTCH30 BIT PASS3
0178  B0B2  10 0C             BPL PT03
0179  B0B4  C9 00             CMP #0          ;EOF AUSGEBEN
0180  B0B6  F0 08             BEQ PT03
0181  B0B8  C9 3B             CMP #';'        ;SEMIC NUR IN FLG, NICHT AUSGEBEN.
0182  B0BA  F0 07             BEQ PT04                        .
0183  B0BC  C5 22             CMP SEMFLG      ;A=CR, WENN FLG =CR, DANN AUSGEBEN
```

```
0184  B0BE  D0 03        BNE PT04       ;WENN NICHT, NUR IN SEMFLG ABLEGEN
0185  B0C0  4C BC E9  PT03  JMP OUTALL
0186  B0C3  85 22      PT04  STA SEMFLG
0187  B0C5  60                RTS
0188  B0C6  FF                .BYT $FF,$FF
0189  B0C8            ;
0190  B0C8            ;6. ERROR DARSTELLUNG AN ASMBL ANPASSEN
0191  B0C8            ;
0192  B0C8                    *=$B0E3
0193  B0E3  2D                .BYT '-'
0194  B0E4                    *=$B112
0195  B112  5E                .BYT '^'
0196  B113            ;
0197  B113            ;7. COMPILER ERROR OHNE BREAK, KLARTEXT
0198  B113            ;
0199  B113                    *=$B82D
0200  B82D            ;
0201  B82D  A0 00      EREXIT LDY #0
0202  B82F  80 0A            BRA EREX1
0203  B831  A0 01            LDY #1
0204  B833  80 06            BRA EREX1
0205  B835  A0 02            LDY #2
0206  B837  80 02            BRA EREX1
0207  B839  A0 03            LDY #3
0208  B83B  B9 6A B0  EREX1  LDA ERMES,Y
0209  B83E  A2 B9            LDX #$B9
0210  B840  20 35 B0         JSR PRINT
0211  B843  4C 47 B0         JMP EXIT
0212  B846            ;
0213  B846  45 52 52  EXMES  .BYT 'ERRORS =',0
0214  B84F            ;
0215  B84F                    *=$B909
0216  B909            ;
0217  B909  20 7D EA  SETPA  JSR HEX
0218  B90C  85 1D            STA PASS
0219  B90E  3A                DEC A
0220  B90F  D0 02            BNE SETPA1
0221  B911  85 21            STA PASS3
0222  B913  60        SETPA1 RTS
0223  B914  FF                .BYT $FF,$FF
0224  B916            ;
0225  B916                    *=$B932
0226  B932            ;
0227  B932  4C E4 CF         JMP CRT        ;CR AN OUTALL
0228  B935            ;
0229  B935  44 4F 2F  M1     .BYT 'DO/END?',0
0230  B93D  49 4E 44  M2     .BYT 'INDEX?',0
0231  B944  42 52 41  M3     .BYT 'BRACKET?',0
0232  B94D  4C 49 4E  M4     .BYT 'LINE?',0
0233  B953            ;
0234  B953            ;8. .DFILE POINTER AUF INPUT HANDLER RICHTEN
0235  B953            ;    JUMP TO USER INPUT HANDLER, IN=U, DEVICE NO.=0
0236  B953            ;
0237  B953                    *=$B64E
0238  B64E  4C B7 05         JMP $05B7      ;INPUT FROM DISK
0239  B651            ;
0240  B651            ;9. DEFAULTWERT DER ENTRY-ANWEISUNG AUF $800 AENDERN.
0241  B651            ;
```

```
Ø242  B651                    *=$CBB8
Ø243  CBB8  24 38 3Ø          .BYT '$8Ø',$BØ
Ø244  CBBC              ;
Ø245  CBBC                    .END PLSPS 34/84

 ERRORS= ØØØØ  (ØØ11)
```

2. Erfahrungen mit PL/65

Bei der Arbeit mit dem Compiler wurden viele Erfahrungen gesammelt. Einige sollen hier weitergegeben werden:

2.1 Eine nicht dokumentierte Anweisung

IM PL-Text erzeugt 'PAGE 'Überschrift'; den Text .PAG 'Überschrift'. Der Assembler des AIM 65 (Programm ab D000) konnte diese Anweisung nicht bearbeiten. Der SPS-Assembler generiert hieraus jedoch einen Seitenvorschub und gibt die Uberschrift auf jeder Seite aus. Die Zeichenkette darf maximal 20 Zeichen enthalten.

2.2 Zuweisungen – Was nicht im Handbuch steht

Es kann zwar mit negativen Zahlen gerechnet werden, diese Zahlen können aber nur als Ein–Byte-Variablen mit einem Wert belegt werden. Zulässig sind nur Konstanten mit dezimalen Werten im Bereich -1 bis -128.

```
Beispiel:       ;BETA = -12;
                LDA #($FFFF+1-12
                STA BETA
```

2.3 Zuweisung eines ASCII-Zeichens

Ein ASCII-Zeichen kann einer Variablen zugewiesen werden:

```
Beispiel:       ;BETA = 'Z';
                LDA #'Z
                STA BETA
```

2.4 Das Ansprechen des Akkumulators

In früheren Veröffentlichungen wurde vorgeschlagen, den Akku mit 'R4=expr' zu besetzen. PL/65 erzeugt daraus LDA expr, STA R4, und der Akku ist für den Aufruf einer System-Routine entsprechend besetzt. – Nun läßt sich der Akku durch die Verwendung des Symbols A auch aus PL/65 heraus ansprechen, der Compiler akzeptiert dieses Symbol:

```
;SHL A;     ;SHR A;     ;ROL A;     ;ROR A;
ASL A       LSR A       ROL A       ROR A
```

Für den erweiterten Befehlssatz der CPU werden auch die folgenden Anweisungen akzeptiert:

```
;DEC A;     ;INC A;     ;A+1;       ;A-1;
DEC A       INC A       INC A       DEC A
```

Diese Anweisungen können nur dann eingesetzt werden, wenn sichergestellt ist, daß der Akkumulator entsprechend besetzt ist. Keinesfalls können sie innerhalb von IF ... THEN, WHILE und FOR ... DO-Blocks benutzt werden.

2.5 Tabulator im Pass 1

Mit '!' erreicht man in der PL/65-Liste das Einrücken einer Zeile um 5 Spalten. Der Compiler akzeptiert den Kontroll-Code HT (hex 08) anstelle von '!' als Tabulator. Dies ist für den Betrieb mit einem Terminal eine sinnvolle Eigenschaft.

Der Compiler bereitet eine Ausgabezeile stets in einem Puffer für 80 Zeichen vor. Wird die Bufferlänge durch zuviele Tabulatorzeichen überschritten, so erfolgt ein Abbruch des Übersetzungsvorganges mit der Fehlermeldung 'LINE?'.

2.6 Linken von PL/65-Teilprogrammen

PL/65 überwacht konsequent die Parität von ENTRY/EXIT (wobei ENTRY entfallen kann und trotzdem gewertet wird) und von DO/BEGIN - END. Wird im Eingabestrom das EOF-Zeichen entdeckt, und ist die Parität dann nicht aufgelöst, so wird der Compiler-Lauf mit der Fehlermeldung 'DO/END?' abgeschlossen. EXIT kann entfallen, wenn vorher die Parität gestimmt hat. - Es können dann PL- und Assembler-Quellmodule im SPS aneinander gekettet werden.

Man ruft den Compiler auf und gibt zur Ausgabe OUT=U, DEVICE=4 an. Auf die Abfrage des Speicherbereiches gibt man FROM=BOTLN. Die Ausgabe des Compilers wird dann an das vorhandene File angehängt. Es werden Module ohne die ENTRY/EXIT-Statements benutzt. Die Fehlermeldung beachtet man nicht.

3. Utilities für die Arbeit mit dem PL/65-Compiler

Mit den zusätzlichen Dienstleistungen,, die das SPS bereithält und mit dem geänderten PL/65 läßt es sich durchaus arbeiten, auch wenn die Sprache 'gewöhnungsbedürftig' ist. Wenn man strukturiert unter Verwendung der DO-BEGIN- und END-Anweisungen arbeitet und diese konsequent einsetzt, wird man schnell lauffähige Programme erzeugen. Die folgenden Utility-Programme bringen dann weitere Verbesserungen in der Code-Erzeugung und in der Handhabung, insbesondere kann eine PL-Prozedur durch Direkteingabe von der Tastatur sofort geprüft werden, wenn das Utility-Programm geladen ist.

Da unter dem SPS von Rockwell gearbeitet wird, konnte das Menue der Dienstleistungen wie folgt in der Form der U-KEY-Menues angelegt werden:

```
PL-UTILITIES FUNCTION NO.=

 Ø  ASSEMBLER
 1  PL/65 PASS 1,2
 2  PL/65 PASS 1,3
 3  STA R4
 4  JMP ZZxxxx
 5  RECOVER EDITOR
 6  FIND PROCEDURE
 7  MONITOR
```

0-3 Umschalten der Sprachen

Der Aufruf des Menueprogrammes erfolgt mit F1 (Taste) oder U9. Im SPS nehmen der Assembler und der PL/65-Compiler den gleichen Adreßraum von hex B000-CFFF ein. Die Funktionen 1-3 schalten die ROM's um. Es wird dann ein Sprung zur jeweiligen Sprache vorgenommen. Hier sollte in das Programm ein eigener Umschalter aufgenommen werden, wenn die Sprachen nicht von der Diskette hereingerufen werden.

Pass 3 im PL/65-Aufruf erzeugt Assembler-Quelltext ohne die sonst eingefügten PL-Statements (siehe auch im ersten Teil dieser Ausführungen).

3 STA R4

Da unter PL/65 der Akkumulator nur eingeschränkt direkt angesprochen werden kann, viele Systemroutinen jedoch eine Vorbesetzung des Registers verlangen, wird vor dem Aufruf mit 'R4=(wert)' der Accu mit dem Werk geladen. PL erzeugt daraus die Folge 'LDA (wert), STA R4'. Diese Utility löscht nun alle Zeilen mit 'STA R4' im PL/65-Listing. - Nach Aufruf dieser Funktion wird zur Sicherheit mit 'SHURE (Y,N)=' die Ausführung eingeleitet.

4 JMP ZZxxxx

Auch mit dieser Funktion wird das PL-Programm nocheinmal nachträglich bearbeitet. Es werden alle Zeilen mit 'JMP ZZxxx' durch 'BRA ZZxxx' ersetzt/optimiert. Da diese Sprungmarken nur innerhalb einer Codefolge erscheinen, die von einem PL-Statement erzeugt werden oder die innerhalb einer mit BEGIN/END markierten Prozedur auftreten, ist praktisch die Sprungweite immer innerhalb der Grenzen zu erwarten, in deren Verzweigungsbefehle benutzt werden können.

Wird nun mit dem erweiterten Befehlssatz gearbeitet, so wird das generierte Programm kürzer und schneller. Die Utility verändert die Sprungweite relativer Verzweigungen der Form 'BXX *+nn' automatisch um -1, soweit die neuen Befehle BRA ZZxxxx übersprungen werden.

Auch hier wird zur Sicherheit mit 'SHURE (Y/N)' die Funktion erst ausgeführt, wenn mit 'Y' geantwortet wird.

Beispiel:

```
;FOR J=1 TO N
 LDA #1
 STA J
 ZZ0001 CMP N
 BEQ *+7
 BCC *+5
 JMP ZZ0002
;DO;
;END;
 INC J
 LDA J
 JMP ZZ0001
 ZZ0002
```

```
;FOR J=1 TO N
 LDA #1
 STA J
 ZZ0001 CMP N
 BEQ *+6
 BCC *+4
 BRA ZZ0002
;DO;
;END;
 INC J
 LDA J
 BRA ZZ0001
 ZZ0002
```

5 Recover Editor

Durch Änderung der PL-ROMs wurde erreicht, daß ein in den Speicher gelenkter Assembler-Quelltext stets mit der EOF-Markierung des Texteditors mit '0' abgeschlossen wird. Die Recover-Funktion des SPS (U7) kann dann eingesetzt werden,um dann zwischen dem PL-Text und dem generierten Assemblertext umzuschalten. Diese Menuefunktion führt nur einen Aufruf U7 durch.

6 Find Procedure

Fehlerfreier Compilertext wird nur dann erzeugt, wenn streng strukturiert unter PL/65 programmiert wird. Dazu werden Unterprogramme und Funktionen mit BEGIN/END und DO/END eingefaßt, während das gesamte Programm mit ENTRY/EXIT markiert wird (ENTRY ist optional). Während des RUN überwacht der Compiler ständig die Parität dieser Markierungen und bricht den Lauf ab, wenn Fehler auftreten.
Das Hilfsprogramm 'FIND PROCEDURE' listet eine mit LABEL:BEGIN; und END; eingefaßte Prozedur, wobei das Ende derselben aus der Parität erkannt wird. Das Listing kann mit OUT= auf jede Ausgabeeinheit gelenkt werden. Anschließend ist man im Texteditor in der ersten Zeile der Prozedur.

Beispiel:

```
LIST PROCEDURE =TEST:

TEST: BEGIN;
FOR I=1 TO J DO;
!R4 = FELD[I];
!CALL OUTALL;
!END;
GOTO COMIN;
END;

TEST: BEGIN;
```

Ist die Parität innerhalb der Prozedur nicht auflösbar, so wird solange gelistet, bis sie aufgelöst ist oder bis das Textende erreicht wird.

7 Monitor

Hier wird aus dem Utilityprogramm zum Monitor verzweigt.

Compiler Input im RUN-Modus von der Tastatur

Mit Initialisierung des Utility-Programmes wird der User-Input-Vektor 5 auf die Routine PLKBD gerichtet. Wird PL/65 aufgerufen, so ist auf den Prompt IN= mit 'U' und auf die folgende Frage DEVICE= mit 5 zu antworten. Mit OUT= CR wird die Compilerausgabe auf den Bildschirm gelelenkt. Es ist dann möglich, ein PL/65-Statement von der Tastatur einzugeben. Es wird dann sofort der generierte entsprechende Assemblerquelltext auf den Bildschirm ausgegeben. – Die RUBOUT-Funktion wurde nicht implementiert. Der Compilermodus wird wie üblich mit 'EXIT;' verlassen.

Erweiterung des Menues

In das Programm können neue Funktionen eingebunden werden. Dazu erweitert man den Menuetext. In DEST ist die entsprechende Anfangsadresse einzutragen und in PLUT1 ist die Anzahl der Funktionen mit CMP #n zu prüfen.

```
0002  0000            ;*****************************************************
0003  0000            ;** PL/65 UTILITIES  29.09.85  W.RADELOFF    'PLUTIL' **
0004  0000            ;*****************************************************
0005  0000            ;
0006  0000                    *=6
0007  0006            PRVEC   *=*+2
0008  0008            PARITY  *=*+1
0009  0009            ;
0010  0009            RUBOUT  =$7F
0011  0009            LF      =$0A
0012  0009            CR      =$0D
0013  0009            STRARG  =$EB            ;SUCHSTRING
0014  0009            LENGTH  =$EA
0015  0009            NOWLN   =$DF
0016  0009            CURPOZ  =$A415
0017  0009            COUNT   =$A419
0018  0009            PTR     =$A429          ;STIY+2
0019  0009            OLDLEN  =$E9
0020  0009            UIN5    =$05C6
0021  0009            PRTBIT  =$05FF
0022  0009            ;
0023  0009            ;SUBROUTINEN AUS DEM SPS-EDITOR
0024  0009            ;GEAENDERTE WERTE GEGENUEBER DEM AIM-EDITOR!
0025  0009            ;
0026  0009            TOPNO   =$F8CE
0027  0009            FC5     =$F85B
0028  0009            UPNO    =$F71B
0029  0009            SUB     =$F928
0030  0009            REPLAC  =$F94A
0031  0009            FCHAR   =$F81E
0032  0009            SAVNOW  =$F93F
0033  0009            RESNOW  =$F8E2
0034  0009            LST01   =$F802
0035  0009            ;
0036  0009            COMIN   =$E1A1
0037  0009            PIA     =$8100
0038  0009            REDOUT  =$E973
0039  0009            HEX     =$EA7D
0040  0009            OUTPUT  =$E97A
0041  0009            RECOVR  =$FCBA
0042  0009            CRLOW   =$EA13
0043  0009            KBIN    =$EF71
0044  0009            BLANK2  =$E83B
0045  0009            WHEREO  =$E871
0046  0009            PRPL    =$B035
```

```
0049  0009                ;************************************************
0050  0009                ;** PL/65 UTILITIES: MENUE, START MIT F1         **
0051  0009                ;************************************************
0052  0009                ;
0053  0009                      *=$010C
0054  010C  4C 00 08            JMP PLUTIL
0055  010F                ;
0056  010F                      *=$0800
0057  0800                ;
0058  0800  20 13 EA      PLUTIL JSR CRLOW
0059  0803  A9 20               LDA #%00100000
0060  0805  0C FF 05            TSB PRTBIT
0061  0808  A9 FD               LDA #<PLKBD
0062  080A  8D C7 05            STA UIN5+1
0063  080D  A9 08               LDA #>PLKBD
0064  080F  8D C8 05            STA UIN5+2
0065  0812  A9 D0               LDA #<PROMT
0066  0814  A2 08               LDX #>PROMT
0067  0816  20 3E 08      PLUT1  JSR PRINT
0068  0819  20 73 E9            JSR REDOUT
0069  081C  20 7D EA            JSR HEX
0070  081F  C9 08               CMP #$8
0071  0821  B0 15               BCS SHMENU
0072  0823  0A                  ASL A
0073  0824  AA                  TAX
0074  0825  7C 28 08            JMP (DEST,X)
0075  0828                ;
0076  0828  2B 09         DEST   .WOR ASMBON,PLON1,PLON2,DELR4,BRANCH
0077  0832  BA FC                .WOR RECOVR,FIND,COMIN
0078  0838                ;
0079  0838  A9 4F         SHMENU LDA #<MENU
0080  083A  A2 08               LDX #>MENU
0081  083C  80 D8               BRA PLUT1
0082  083E                ;
0083  083E  85 06         PRINT  STA PRVEC
0084  0840  86 07               STX PRVEC+1
0085  0842  A0 00               LDY #0
0086  0844  B1 06         PR00   LDA (PRVEC),Y
0087  0846  F0 06               BEQ PR01
0088  0848  20 7A E9            JSR OUTPUT
0089  084B  C8                  INY
0090  084C  80 F6               BRA PR00
0091  084E  60            PR01   RTS
0092  084F                ;
0093  084F  0D            MENU   .BYT CR,CR,' 0  ASSEMBLER'
0094  085E  0D                   .BYT CR,' 1  PL/65 PASS 1,2'
0095  0871  0D                   .BYT CR,' 2  PL/65 PASS 1,3'
0096  0884  0D                   .BYT CR,' 3  STA R4'
0097  088F  0D                   .BYT CR,' 4  JMP ZZxxxx'
0098  089E  0D                   .BYT CR,' 5  RECOVER EDITOR'
0099  08B1  0D                   .BYT CR,' 6  FIND PROCEDURE'
0100  08C4  0D                   .BYT CR,' 7  MONITOR'
0101  08D0  0D            PROMT  .BYT CR,CR,'PL-UTILITIES FUNCTION NO.=',0
0102  08ED  20 20 53      SHURE  .BYT '  SHURE (Y,N) =',0
0104  08FD                ;************************************************
0105  08FD                ;** PL/65 UTILITY: PLCALC. DIRECT COMPILATION FROM KBD *
0106  08FD                ;************************************************
0107  08FD                ;
0108  08FD                ;PROMT 'IN=', ENTER <U>,<S>
```

```
0110  08FD  90 1D      PLKBD  BCC PK03
0111  08FF  20 71 EF          JSR KBIN
0112  0902  C9 7F            CMP #RUBOUT
0113  0904  F0 09            BEQ PK01
0114  0906  C9 40            CMP #$40        ;CONVERT LOWERCASE
0115  0908  30 02            BMI PK00
0116  090A  29 DF            AND #$DF
0117  090C  4C 7A E9   PK00   JMP OUTPUT
0118  090F                   ;
0119  090F  A6 08      PK01   LDX $08         ;PL/65 IO BUFFER POINTER
0120  0911  F0 07            BEQ PK02
0121  0913  C6 08            DEC $08
0122  0915  A9 7F            LDA #RUBOUT
0123  0917  20 7A E9          JSR OUTPUT
0124  091A  A9 0A      PK02   LDA #LF
0125  091C  60        PK03   RTS
0126  091D                   ;
0127  091D                   ;################################################
0128  091D                   ;## SWITCH ROM BANKING PL/65 - ASSEMBLER        ##
0129  091D                   ;################################################
0130  091D                   ;
0131  091D                   ;ASMBON: SWITCH ASSEMBLER ROMS ON AND JUMP
0132  091D                   ;PLON  : SWITCH PL/65 ROMS ON AND JUMP (PASS1,3)
0133  091D                   ;
0134  091D  A9 07      SWITCH LDA #$07        ;INIT PIA
0135  091F  8D 00 B1         STA PIA
0136  0922  A9 04            LDA #$04
0137  0924  8D 01 B1         STA PIA+1
0138  0927  8E 00 B1         STX PIA
0139  092A  60               RTS
0140  092B                   ;
0141  092B  A2 D3      ASMBON LDX #$D3
0142  092D  20 1D 09         JSR SWITCH
0143  0930  4C 36 B0         JMP $B036      ;ASSEMBLER ENTRY
0144  0933                   ;
0145  0933  A2 D2      PLON1  LDX #$D2
0146  0935  20 1D 09         JSR SWITCH
0147  0938  4C 03 B0         JMP $B003
0148  093B                   ;
0149  093B  A2 D2      PLON2  LDX #$D2
0150  093D  20 1D 09         JSR SWITCH
0151  0940  4C 00 B0         JMP $B000
0154  0943                   ;################################################
0155  0943                   ;## CHANGE JMP ZZxxxx TO BRA ZZxxxx, DEC BXX #=X  ##
0156  0943                   ;################################################
0157  0943                   ;
0158  0943                   ;
0159  0943  4A 4D 50   STR1   .BYT 'JMP ZZ',0
0160  094A  53 54 41   STR2   .BYT 'STA R4 ',0
0161  0952  42 45 47   STR3   .BYT 'BEGIN    ',0
0162  0959  44 4F 3B   STR4   .BYT 'DO       ',0
0163  095D  45 4E 44   STR5   .BYT 'END      ',0
0164  0962  42 52 41   CHANGE .BYT 'BRA'
0165  0965                   ;
0166  0965                   ;SUCHWORT INITIALISIEREN
0167  0965                   ;
0168  0965  20 E0 09   BRANCH JSR START
0169  0968  A2 00             LDX #0
0170  096A  20 CE 09          JSR INIWRD
```

```
0172  096D                ;SUCHE ZEILE MIT 'JMP ZZXXXX'
0173  096D                ;
0174  096D  20 1B F7      FD04   JSR UPNO
0175  0970  20 5B F8             JSR FC5
0176  0973                ;
0177  0973                ;PRUEFT, OB 'JMP ZZ' AM ZEILENANFANG
0178  0973                ;
0179  0973  B2 DF                LDA (NOWLN)
0180  0975  C9 4A                CMP #'J'
0181  0977  D0 F4                BNE FD04
0182  0979                ;
0183  0979                ;ERSETZT 'JMP' DURCH 'BRA'.
0184  0979                ;
0185  0979  A0 02                LDY #2
0186  097B  B9 62 09      FD03   LDA CHANGE,Y
0187  097E  91 DF                STA (NOWLN),Y
0188  0980  88                   DEY
0189  0981  10 F8                BPL FD03
0190  0983                ;
0191  0983                ;RELATIVEN SPRUNG VORHER -1
0192  0983                ;
0193  0983  A2 03                LDX #3
0194  0985  20 C3 09             JSR RETURN
0195  0988  D0 28                BNE SR01
0196  098A  A0 01                LDY #1
0197  098C  B1 DF                LDA (NOWLN),Y
0198  098E  3A                   DEC A
0199  098F  91 DF                STA (NOWLN),Y
0201  0991                ;ZWEITEN RELATIVEN SPRUNG EBENFALLS -1
0202  0991                ;
0203  0991  A2 08                LDX #8
0204  0993  20 C3 09             JSR RETURN
0205  0996  D0 0B                BNE SR00
0206  0998  B1 DF                LDA (NOWLN),Y
0207  099A  C9 37                CMP #'7'
0208  099C  D0 05                BNE SR00
0209  099E  3A           FD05   DEC A
0210  099F  91 DF                STA (NOWLN),Y
0211  09A1  80 CA                BRA FD04
0212  09A3                ;
0213  09A3                ;EVENTUELL EIN ZEICHEN VOR SUCHEN
0214  09A3                ;
0215  09A3  A2 01        SR00   LDX #1
0216  09A5  20 C3 09             JSR RETURN
0217  09A8  D0 C3                BNE FD04
0218  09AA  B1 DF                LDA (NOWLN),Y
0219  09AC  C9 37                CMP #'7'
0220  09AE  F0 EE                BEQ FD05
0221  09B0  80 BB                BRA FD04
0222  09B2                ;
0223  09B2                ;SUCHE NACH 'BNE  *+11' UND AENDERE ZU '*+10'
0224  09B2                ;
0225  09B2  A2 1D        SR01   LDX #29
0226  09B4  A0 02                LDY #2
0227  09B6  20 C3 09             JSR RETURN
0228  09B9  D0 B2                BNE FD04
0229  09BB  B1 DF                LDA (NOWLN),Y
0230  09BD  C9 31                CMP #'1'
0231  09BF  F0 DD                BEQ FD05
```

```
0232  09C1  80 AA        BRA FD04
0233  09C3              ;
0234  09C3              ;SR: AUFWAERTS IM TEXT
0235  09C3              ;
0236  09C3  20 28 F9    RETURN JSR SUB
0237  09C6  CA                 DEX
0238  09C7  D0 FA              BNE RETURN
0239  09C9  B2 DF              LDA (NOWLN)
0240  09CB  C9 2B              CMP #'+'
0241  09CD  60                 RTS
0242  09CE              ;
0243  09CE              ;SR: SUCHWORT INITIALISIEREN
0244  09CE              ;
0245  09CE  A0 00       INIWRD LDY #0
0246  09D0  BD 43 09    IW00   LDA STR1,X
0247  09D3  F0 07              BEQ IW01
0248  09D5  99 EB 00           STA STRARG,Y
0249  09D8  E8                 INX
0250  09D9  C8                 INY
0251  09DA  80 F4              BRA IW00
0252  09DC  8C 29 A4    IW01   STY PTR
0253  09DF  60                 RTS

0255  09E0              ;SR: PROMT SHURE AND RETURN TO PLUTIL, IF NO
0256  09E0              ;
0257  09E0  A9 ED       START  LDA #<SHURE
0258  09E2  A2 08              LDX #>SHURE
0259  09E4  20 3E 08           JSR PRINT
0260  09E7  20 73 E9           JSR REDOUT
0261  09EA  C9 59              CMP #'Y'
0262  09EC  F0 05              BEQ YES
0263  09EE  68                 PLA
0264  09EF  68                 PLA
0265  09F0  4C 00 08           JMP PLUTIL
0266  09F3              ;
0267  09F3  20 13 EA    YES    JSR CRLOW
0268  09F6  4C CE F8           JMP TOPNO

0271  09F9              ;##############################################
0272  09F9              ;## DELETE LINES WITH 'STA R4 '                ##
0273  09F9              ;##############################################
0274  09F9              ;
0275  09F9              ;SUCHE VORBEREITEN
0276  09F9              ;
0277  09F9  20 E0 09    DELR4  JSR START
0278  09FC  A2 07              LDX #STR2-STR1 ;'STA R4 '
0279  09FE  20 CE 09           JSR INIWRD
0280  0A01              ;
0281  0A01              ;ZEILE NACH SUCHWORT PRUEFEN
0282  0A01              ;
0283  0A01  20 5B F8    DE00   JSR FC5
0284  0A04              ;
0285  0A04              ;ZEILE LOESCHEN
0286  0A04              ;
0287  0A04  64 EA              STZ LENGTH
0288  0A06  9C 19 A4           STZ COUNT
0289  0A09  A0 07              LDY #7
0290  0A0B  84 E9              STY OLDLEN
0291  0A0D  20 4A F9           JSR REPLAC
0292  0A10  80 EF              BRA DE00
```

```
0295  0A12                ;***********************************************
0296  0A12                ;** LIST PROCEDUR FROM PL-SOURCE TEXT           **
0297  0A12                ;***********************************************
0298  0A12                ;
0299  0A12                ;VORBEREITEN, 1.ZEILE MIT SUCHWORT
0300  0A12                ;
0301  0A12  20 CE F8     FIND    JSR TOPNO
0302  0A15  20 3B E8             JSR BLANK2
0303  0A18  20 71 E8             JSR WHEREO
0304  0A1B  20 13 EA             JSR CRLOW
0305  0A1E  A9 62                LDA #<LIST
0306  0A20  A2 0A                LDX #>LIST
0307  0A22  20 3E 00             JSR PRINT
0308  0A25  20 1E F8             JSR FCHAR
0309  0A28  20 3F F9             JSR SAVNOW
0310  0A2B  64 08                STZ PARITY
0311  0A2D  9C 19 A4             STZ COUNT
0312  0A30                ;
0313  0A30                ;ZEILE AUF BEGIN      ; DO; UND END; PRUEFEN
0314  0A30                ;
0315  0A30  EE 19 A4     SRCH0   INC COUNT
0316  0A33  A2 0F                LDX #STR3-STR1 ;BEGIN;
0317  0A35  20 73 0A             JSR SEARCH
0318  0A38  90 04                BCC SRCH1
0319  0A3A  E6 08                INC PARITY
0320  0A3C  80 16                BRA NEXT
0321  0A3E  A2 16        SRCH1   LDX #STR4-STR1 ;DO;
0322  0A40  20 73 0A             JSR SEARCH
0323  0A43  90 04                BCC SRCH2
0324  0A45  E6 08                INC PARITY
0325  0A47  80 0B                BRA NEXT
0326  0A49  A2 1A        SRCH2   LDX #STR5-STR1 ;END;
0327  0A4B  20 73 0A             JSR SEARCH
0328  0A4E  90 04                BCC NEXT
0329  0A50  C6 08                DEC PARITY
0330  0A52  F0 05                BEQ PRPROZ
0331  0A54  20 1B F7     NEXT    JSR UPNO
0332  0A57  80 D7                BRA SRCH0
0333  0A59                ;
0334  0A59                ;PROZEDUR LISTEN
0335  0A59                ;
0336  0A59  20 E2 F8     PRPROZ JSR RESNOW
0337  0A5C  20 1B F7             JSR UPNO
0338  0A5F  4C 02 F8             JMP LST01
0339  0A62                ;
0340  0A62  4C 49 53     LIST    .BYT 'LIST PROCEDURE =',0
0342  0A73                ;SR: SEARCH, SUCHT NACH ZEILE MIT STRARG
0343  0A73                ;C=1 GEFUNDEN, C=0 NICHT GEFUNDEN
0344  0A73                ;
0345  0A73  20 CE 09     SEARCH JSR INIWRD
0346  0A76  A0 00                LDY #0
0347  0A78  8C 15 A4             STY CURPO2
0348  0A7B  AC 15 A4     FC6     LDY CURPO2
0349  0A7E  A2 00                LDX #0
0350  0A80  B1 DF        FC7     LDA (NOWLN),Y
0351  0A82  C9 0D                CMP #CR
0352  0A84  F0 12                BEQ NFOUND
0353  0A86  D5 EB                CMP STRARG,X
0354  0A88  F0 05                BEQ FC9
```

```
0355  0A8A  EE 15 A4          INC CURPO2
0356  0A8D  80 EC             BRA FC6
0357  0A8F  C8          FC9   INY
0358  0A90  E8                INX
0359  0A91  EC 29 A4          CPX PTR
0360  0A94  D0 EA             BNE FC7
0361  0A96  38                SEC
0362  0A97  60                RTS
0363  0A98              ;
0364  0A98  18          NFOUND CLC
0365  0A99  60                RTS
0366  0A9A              ;
0367  0A9A                    .END
```

Roland Löhr

Eproms ab Typ 2764

Die Entwicklung am Markt und in den Geräten ist in den letzten Jahren schnell über die Typen 2532 und 2732 hinausgegangen. Man kann wohl sagen, daß derzeit bereits der Typ 27128 aktuell ist, nicht zuletzt wegen des geringen Preises, der auch bei kleinen Mengen deutlich unter DM 10 liegt.

Beim Programmieren der 'höheren' EPROMs ab 2764 erlebt man nun sehr schnell Überraschungen: Die älteren Eprommer-Geräte sind oft mit der Zusicherung verkauft worden, sie könnten, ggfs. durch Umschaltungen auch die höheren Typen programmieren. In der Praxis zeigte sich in zwei Fällen, daß sie nicht einmal programmierte EPROMs dieser Art in das RAM einlesen können oder daß es nur bei bestimmten Fabrikaten geht. In anderen Fällen wurden alle programmierten Bytes als 'nicht übereinstimmend' moniert, weil man offensichtlich nicht korrekt lesen und vergleichen kann. In einigen Fällen konnten solche EPROMs gleichwohl auf dem Zielrechner in Betrieb genommen werden. Oder es gab Fälle, in denen das EPROM als 'gleich' aus der Programmierung kam und dann trotz gleicher Arbeitsgeschwindigkeit vom Zielrechner nicht akzeptiert wurde.

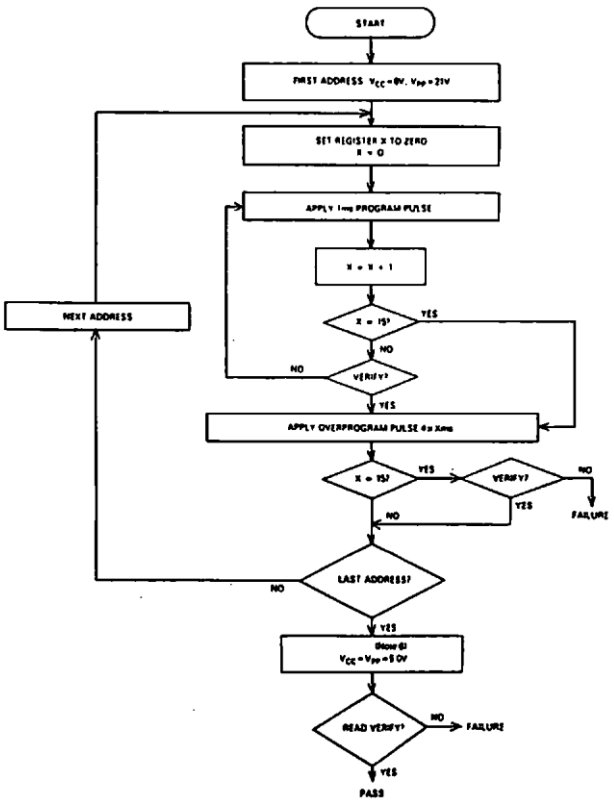
Ich erwähne das und auch das Folgende, damit der Leser nicht verzweifelt, wenn sich einmal der Erfolg nicht einstellen will, denn normalerweise stehen ihm nicht die professionellen Hilfsmittel zur Verfügung, die man in den Betrieben antrifft. Dort wird zusätzlich ja auch eine Wareneingangsprüfung vorgenommen, um die Einhaltung der Spezifikationen zu bestimmen.

Die zweite Überraschung betrifft die EPROMs selbst. Mit der nachstehenden Schaltung und mit dem nachstehenden Programm konnten die Typen Intel D27128-4 und Toshiba TMM27128D-250 einwandfrei programmiert werden, jedoch nicht der entsprechende Typ eines anderen japanischen Herstellers. Bei diesem waren die Bytes schon nach 1 ms Brennimpuls zu vorwiegend 00 'verbrannt. - Insofern bitte ich die nachfolgenden Vorschläge als solche zu betrachten, und nicht als Erfolgsgarantie.

In Heft 36 wurde bereits ein Eprommer für 2764 und 27128 aus der Feder von W. Krebs vorgestellt. Diese Schaltung war in mancher Weise Vorbild für die folgende. Wegen drängender Arbeiten wurde jedoch nach einer noch einfacheren Lösung in der Hardware gesucht. Auch sollte der sog. 'intelligente Programmieralgorithmus' im Programm verwirklicht werden. Er kürzt das Programmieren erheblich ab. Braucht man für ein 2532 oder 2732 etwa 5 Minuten zur vollständigen Programmierung, so würde man für ein 27128 etwa 20 Minuten benötigen, wenn man die frühere Brennmethode anwendet. Diese bestand darin, daß man für jedes Byte einen einzigen Brennimpuls von 50 ms sendet, um dem EPROM danach 50 ms Pause zu gewähren, damit es thermisch nicht beschädigt wird.

Für den 'intelligenten Algorithmus' das nebenstehende Schema. Er hat folgende Grundidee: Ein Byte wird mit lauter kurzen Impulsen von je 1 ms Dauer programmiert und sofort wieder gelesen. Und das höchstens 15mal. Parallel zählt man die ausgesandten Brennimpulse in einem Register,

z.B. X. Sobald nun ein Byte erstmals 'festgeklopft' ist, daß es sich so ausliest, wie programmiert, dann gibt man einen Rest-Brennimpuls (Overburn) auf das Byte, dessen Zeit 4xX ms ist, also maximal 60 ms, in den meisten Fällen erheblich weniger. War das Byte nach den ersten 15 Brennimpulsen noch nicht 'fest', so liegt es wohl außerhalb der üblichen Toleranz, was noch nicht fatal ist. Spätestens aber muß es nach dem Overburn mit der Programmierung übereinstimmen, andernfalls sollte man das EPROM ausscheiden.



Das hier beschriebene 'intelligente' Brennverfahren ist daran gekoppelt, daß man während der Programmierung eines EPROMs eine auf 6 Volt erhöhte Betriebsspannung an den Pin 28 des EPROMs als Vcc anlegt. Die Spannungen sind in recht engen Grenzen einzuhalten. Für diese +6V gilt plus/minus 0,25 V. Auch bei der normalen Betriebsspannung von 5V gilt plus/minus 5%. Man achte nun auf das Datenblatt: Etliche Typen verlangen eine Programmierspannung von 21V +-0,5V, andere, wie zum Beispiel 2764A (man achte auf das 'A') 12,5V +-0,3 V.

Im Datenblatt zum 27256 von Intel steht dazu noch ein abweichender Algorithmus: Bis zu 25mal einen Brennimpuls von 1 ms ausgeben, wenn das Byte schon vorher sich wie vorgegeben liest, dann noch 3xX ms Overburn nachbrennen. Wenn sich das Byte bei X=25 noch nicht wie die Vorgabe auslesen läßt, dann ist das EPROM schadhaft. Wenn doch übereinstimmend, so wird 3xX ms nachgebrannt. Für den Typ 27256 wird die Programmierspannung Vpp mit 12,5V angegeben (Intel).

Man beachte, daß das EPROM in spannungslosem Zustand in seinen Sockel eingelegt wird. Die Programmierspannung Vpp darf erst mit oder nach der Betriebsspannung Vcc angelegt werden. Das Signal CE (Chip Enable) wird man wohl permanent auf Low legen. Pin 22, OE ist wäh-

rend des Brennimpulses auf High zu halten. Er wird zum Zwecke des Kontroll-Lesens auf Low gezogen. Im vorliegenden Kontext ist das Signal OE mit dem Pin CB2 der VIA verbunden. Die Konstante READ–EIN wird beim Lesen in das PCR geschrieben (Peripheral Control Register). Dadurch wird CA2 im Handbetrieb (Manual Output Mode) auf Low gezogen. Umgekehrt mit READ–AUS.

Man muß noch eine Besonderheit erwähnen, die sicher auf professionellen Programmiergeräten verwirklicht sind, wenn sie der Technik der Zeit entsprechen: Ein EPROM kann sich selbst nach Hersteller und Typ ausweisen, wenn man an den Adreßpin A9 eine Spannung von +12V anlegt und alle Pins Ax, außer A0 auf Low legt, dann kann man die Signatur des Herstellers als 1 Byte auf dem Datenbus auslesen, wenn A0=Low. Wenn dann A0=High gesetzt wird, dann verrät das EPROM in einem Byte auch seinen Typ, so daß sich ein prozessorgesteuerter Eprommer von selbst auf die Parameter einstellen kann. Leider wird diese Identifizierung noch nicht bei allen Herstellern und Typen benutzt. Intel nennt sie 'Intelligent Identifier Mode, Es gibt aber auch die Bezeichnungen Silicon Signature, Electric Signatur Mode oder Auto Select Mode.

Fachleute weisen jedoch darauf hin, daß das EPROM durch die Spannung von 12V an Pin A9 zerstört werden kann, wenn es nicht für die Selbst-Identifizierung eingerichtet ist. Es sind also einige Spielarten anzutreffen. Es ist daher begrüßenswert, wenn sich die Leser mit zunehmenden Erfahrungen zu diesem Thema einmal rückäußern würden.

Der Eprommer und sein Betriebsprogramm fußen auf folgenden Grundgedanken: Man hat an einer VIA 6522 als Interface nicht genug Pins, um alle notwendigen Signale an das EPROM anlegen zu können. Viele veröffentlichte Schaltungen arbeiten daher mit gemultiplexten Signalen. Das wurde hier vermieden. Stattdessen wird ein Binärzähler 4040 benutzt, um die Adressen A0 bis A11 an das EPROM anzulegen. Das spart immerhin 11 Interface-Pins. Im praktischen Betrieb heißt es, daß der Binärzähler per Reset zunächst auf 00 zu setzen ist. Mit dem Unterprogramm PULS–UP wird er dann zunächst auf die erste benötigte Adresse des EPROMs hochgezählt. Danach kann das Auslesen oder das Brennen von Bytes beginnen, wenn der Computer auch die Adressen A12 und A13 angelegt hat (Routine GEN–A12). Die Erfahrung zeigt, daß ab einer Startadresse immer viele Bytes gelesen oder gebrannt werden, so daß die Zeit für das Hochpulsen des Binärzählers nicht ins Gewicht fällt. Später wird er dann immer mit PULS–PLUS um +1 inkrementiert.

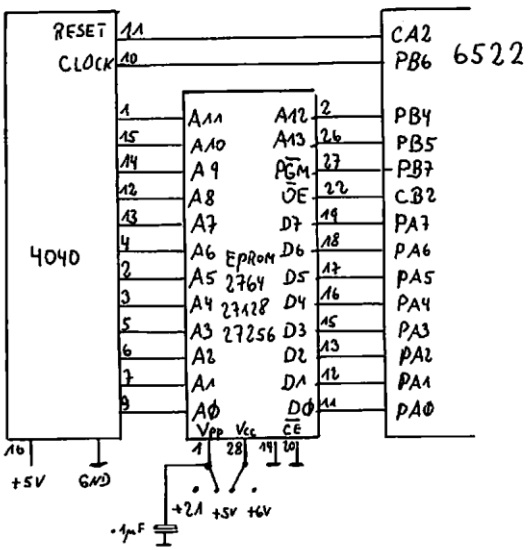
Bei der Erzeugung des Brennimpulses am Pin 27 mit dem Signal PGM (aktiv Low) wird von einer besonderen Betriebsart der VIA Gebrauch gemacht: Im Zusammenhang mit der Steuerung des ACR (Auxiliary Control Register) kann an Pin PB7 des Portes B ein nach Low gehender Puls ausgegeben werden, dessen Dauer sich nach der Vorladung des Timer 1 richtet; Betrieb als Monoflop also. Dieser Impuls wird je nach Algorithmus und notwendiger Rest-Brenndauer bemessen (OVER–BURN).

Nach dem Start nimmt das Programm eine Sicherheits-Abfrage vor (ob die 21V ausgeschaltet sind) und zeigt dann das Menue. Wir haben folgende Befehle:

- E Einlesen des EPROM-Inhaltes in das RAM, z.B. zum Kopieren
- L Listen eines EPROM-Bereiches
- R Listen eines RAM Bereiches
- V Verifizieren EPROM gegen RAM
- C Checken, prüfen, ob ein EPROM-Bereich leer ist
- P Programmieren des EPROMs. Mit der Taste 'I' kann der intelligente Algorithmus gewählt werden, sonst gilt die Methode mit 50 ms Brenndauer je Byte.
  Das Programm prüft genau, ob die Mengen nach den Bereichsvorgaben für RAM und für EPROM gleich sind. Wenn nicht, dann gelangt man wieder in das Menue. Wenn alles klar ist, wird man zum Umschalten der Spannungen aufgefordert, am Schluß ebenso zum Zurückschalten.
  Während des Brennvorganges zeigt das Programm bei jedem 256. Byte den in das EPROM zeigenden Zählerstand an, z.B. 0800 .. 0900 usw.
  Wenn ein Byte 'lahm' ist, also außerhalb der normalen Spezifikation, dann wird das als Warnung ausgegeben. Hauptsache ist, daß man am Schluß mit . . . . . 0000 Fatale Fehler' herauskommt.

A Abweichungen EPROM/RAM listen
Q Quit, Sprung zum Betriebsprogramm des Computers

Nach dem Aufbau der Platine prüfe man sorgfältig, daß die Signale dort anliegen, wo sie erwartet werden. Die Verdrehung von Adreßbus- und Datenbussignalen mag dem Eprommer und seinem Programm nicht auffallen, gleichwohl mag der Computer einen recht chaotischen Festwertspeicher erhalten. - Es ist auch sehr zu empfehlen, nach dem Eintippen des Programmes Probeläufe zu machen, um einerseits den Ablauf der Bedienerführung zu üben und um andererseits mit dem Oszilloskop festzustellen, daß die Zählimpulse am Binärzähler ankommen und daß der Puls an PGM die richtige Weite hat.



Beschaltung des Eprommers

```
0001                            ;$KD"PROMMER 64/128"
0002                            ;PROGRAMM F. DEN EPROMMER F. AIM 65
0003                            ;BRENNEN DER TYPEN 2764/27128
0004                            ;DATUM 8.10.85
0005
0006                            ;HARDWARE-ADRESSEN:
0007 000000     VIA             =$A000          ;USER-VIA
0008 000000     PORTA           =VIA+1
0009 000000     PORTB           =VIA
0010 000000     DDRA            =VIA+3          ;DATENRICHTUNG
0011 000000     DDRB            =VIA+2
0012 000000     ACR             =VIA+$B         ;STEUERUNG TIMER 1
0013 000000     PCR             =VIA+$C
0014 000000     IFR             =VIA+$D         ;INTERRUPT FLAG REGISTER
0015 000000     T1_LLOW         =VIA+4          ;LATCH LOW
0016 000000     T1_CHIGH        =VIA+5          ;COUNTER HIGH
0017
0018                            ;KONSTANTEN:
0019 000000     PULS_1MS        =1000           ;1 MS-PULS BEI 1 MHZ
0020 000000     PULS_50MS       =50000          ;50000 TAKTZYKLEN 1 MHZ
0021 000000     CR              =$0D
0022 000000     SPACE           =$20
0023 000000     BIT6            =%01000000      ;BIT FUER CLOCK
0024 000000     CNT_EIN         =%00001100      ;RESET AM 4040 AUF LOW ZIEHEN
0025 000000     CNT_AUS         =%00001110      ;DITO AUF HIGH= RESET TO 00
0026 000000     READ_EIN        =%11000000      ;EPROM PER CA2 AUF AUSGABE BT
                                  ELLEN
```

```
0027 000000             READ_AUS  =%11100000          ;DITO, AUSGABE BEIM PROGRAMMI
                                      EREN SPERREN
0028
0029                              ;SYSTEMROUTINEN
0030 000000             CRLF      =$E9F0               ;AUSGABE CR/LF
0031 000000             OUTPUT    =$E97A               ;ZEICHENAUSGABE AUF AKTIVE EI
                                      NHEIT
0032 000000             NUMA      =$EA46               ;1 HEXZEICHEN ALS 2 ASCII AUS
                                      GEBEN
0033 000000             FROM      =$E7A3               ;ABFRAGE ANFANGSADRESSE
0034 000000             TO        =$E7A7               ;ABFRAGE ENDADRESSE
0035 000000             GETKEY    =$EC40               ;ZEICHEN VON TASTATUR HOLEN
0036                              ;SYSTEM-RAM
0037 000000             ADDR      =$A41C               ;HIER LANDEN EINGABE-ADRESSEN
0038 000000             JUMP      =$A47D               ;ABLAGE INDIREKTER SPRUNGADRE
                                      SSEN
0039
0040                              ;VARIABLEN IM RAM
0041 000000                       *=$EA                ;ZERO PAGE-BEREICH
0042 0000EA             COUNTER   *=*+2                ;ZUORDNUNGSZAEHLER F. ZAEHLER
                                      PULSUNG
0043 0000EC             PROMLOW   *=*+2                ;PROM-ADRESSE AB DER GEBRANNT
                                      WIRD
0044 0000EE             PROMHIGH  *=*+2                ;LETZTE PROM-ADRESSE, DIE GEB
                                      RANNT WIRD
0045 0000F0             IMPULSE   *=*+1                ;ZAEHLER FUER BRENNPULSE
0046 0000F1             RAMLOW    *=*+2                ;RAM-ADRESSE LOW, BRENNEN, LE
                                      SEN USW.
0047 0000F3             RAMHIGH   *=*+2                ;RAM-ENDADRESSE,BRENNEN, VERG
                                      LEICHEN
0048 0000F5             RAMLFP    *=*+2                ;LAUFENDE RAM-ADESSE
0049 0000F7             TEMP      *=*+1                ;HILFSREGISTER
0050 0000F8             MESPTR    *=*+2                ;POINTER F. MESSAGES
0051 0000FA             MODE      *=*+1                ;ZEIGT BEFEHLSART AN
0052 0000FB             ERRORS    *=*+2                ;FEHLERZAHL
0053
0054                              ;***** HAUPTPROGRAMM
0055 0000FD                       *=$7000
0056 007000 201074      MAIN      JSR INITALL          ;INTERFACES ZURUECKSETZEN
0057 007003 A900                  LDA #0               ;SYSTEM-MELDUNG AUSGEBEN
0058 007005 202D74                JSR TEXT_OUT
0059 007008 2040EC      MAIN1     JSR GETKEY
0060 00700B C90D                  CMP #CR              ;BESTAETIGT?
0061 00700D D0F9                  BNE MAIN1
0062
0063 00700F 201074      RE_ENTER  JSR INITALL          ;ZUR SICHERHEIT RE-INIT.
0064 007012 A901                  LDA #1               ;REENTRY NACH AUSFUEHRENDEN R
                                      OUTINEN
0065 007014 202D74                JSR TEXT_OUT         ;HAUPTMENUE
0066 007017 2040EC      RE_ENTER1 JSR GETKEY           ;BEFEHLSTASTE HOLEN
0067 00701A 207AE9                JSR OUTPUT
0068 00701D 48                    PHA                  ;ZEICHEN RETTEN
0069 00701E 20F0E9                JSR CRLF
0070 007021 68                    PLA                  ;BEFEHL ZURUECK
0071 007022 A207                  LDX #7               ;8 BEFEHLE KOMMEN IN BETRACHT
0072 007024 DD7C77      BEFLOOP   CMP BEFEHLE,X
0073 007027 F006                  BEQ EXECUTE
0074 007029 CA                    DEX
0075 00702A 10F8                  BPL BEFLOOP
0076 00702C 4C1770                JMP RE_ENTER1
0077
0078 00702F 8A          EXECUTE   TXA                  ;BEFEHLSENTSCHLUESSELUNG
0079 007030 0A                    ASL A                A ;MAL 2
0080 007031 AA                    TAX
0081 007032 BD8477                LDA BEFTAB,X
0082 007035 8D7DA4                STA JUMP
0083 007038 BD8577                LDA BEFTAB+1,X
0084 00703B 8D7EA4                STA JUMP+1
0085 00703E A900                  LDA #0
0086 007040 85FA                  STA MODE             ;NEUTRALISIERE BEFEHLSART
0087 007042 85FB                  STA ERRORS           ;FEHLERZAHL
0088 007044 85FC                  STA ERRORS+1
0089 007046 6C7DA4                JMP (JUMP)           ;ZUR AUSFUEHRUNG
0090
```

```
0091 007049 4CB2E1   QUIT      JMP $E1B2          ;START DES AIM 65
0092 00704C A9C0     ABWEICH   LDA #$C0           ;ABWEICHUNGEN AUSLISTEN
0093 00704E 85FA               STA MODE
0094 007050 200373             JSR OPEN_RD2
0095 007053 4C5D70             JMP LESEN2
0096 007056 A980     VERIFY    LDA #$80           ;BEFEHLSART
0097 007058 85FA               STA MODE
0098
0099 00705A 20F572   EINLESEN  JSR OPEN_READ      ;INTERFACE ETC. OEFFNEN
0100 00705D 20D873   LESEN2    JSR WAIT_ABIT      ;KONSOLIDIEREN
0101 007060 AD01A0             LDA PORTA          ;DATEN HOLEN
0102 007063 A000               LDY #0
0103 007065 24FA               BIT MODE
0104 007067 3005               BMI LESEN2A
0105 007069 91F5               STA (RAMLFP),Y     ;BYTE ABLEGEN
0106 00706B 4C8970             JMP LESEN2B
0107 00706E D1F5     LESEN2A   CMP (RAMLFP),Y
0108 007070 F017               BEQ LESEN2B
0109 007072 E6FB               INC ERRORS
0110 007074 D013               BNE LESEN2B
0111 007076 E6FC               INC ERRORS+1
0112 007078 24FA               BIT MODE           ;FEHLER ANZEIGEN?
0113 00707A 500D               BVC LESEN2B
0114 00707C 20F0E9             JSR CRLF
0115 00707F A5EB               LDA COUNTER+1
0116 007081 2046EA             JSR NUMA           ;ADRESSE AUSGEBEN
0117 007084 A5EA               LDA COUNTER
0118 007086 2046EA             JSR NUMA
0119 007089 A5EB     LESEN2B   LDA COUNTER+1      ;ENDBEDINGUNG ERREICHT?
0120 00708B C5EF               CMP PROMHIGH+1
0121 00708D D018               BNE LESEN3
0122 00708F A5EA               LDA COUNTER
0123 007091 C5EE               CMP PROMHIGH
0124 007093 D012               BNE LESEN3
0125 007095 A5FC               LDA ERRORS+1
0126 007097 05FB               ORA ERRORS
0127 007099 D008               BNE LESEN2D
0128 00709B A904               LDA #4             ;MELDUNG:KEINE FEHLER
0129 00709D 202D74   LESEN2E   JSR TEXT_OUT
0130 0070A0 4C0F70   LESEN2F   JMP RE_ENTER       ;FERTIG, ZUM MENUE
0131 0070A3 A905     LESEN2D   LDA #5
0132 0070A5 D0F6               BNE LESEN2E
0133
0134 0070A7 205773   LESEN3    JSR RAM_PLUS       ;ZEIGER ERHOEHEN
0135 0070AA 20C473             JSR PULS_PLUS      ;NEUE ADRESSE ANLEGEN
0136 0070AD 201473             JSR GEN_A12
0137 0070B0 4C5D70             JMP LESEN2
0138
0139 0070B3 203973   RAMLIST   JSR RAM_VON
0140 0070B6 205E73             JSR RAM_BIS        ;ABFRAGEN
0141 0070B9 204E73             JSR RAMLFPINT
0142 0070BC A000               LDY #0
0143 0070BE A20F     RAMLIST1  LDX #15            ;16 ITEMS
0144 0070C0 A5F6               LDA RAMLFP+1
0145 0070C2 2046EA             JSR NUMA
0146 0070C5 A5F5               LDA RAMLFP
0147 0070C7 2046EA             JSR NUMA
0148 0070CA A920               LDA #SPACE
0149 0070CC 207AE9             JSR OUTPUT
0150 0070CF B1F5     RAMLIST2  LDA (RAMLFP),Y
0151 0070D1 2046EA             JSR NUMA           HEXBYTE AUSGEBEN
0152 0070D4 A920               LDA #SPACE
0153 0070D6 207AE9             JSR OUTPUT
0154 0070D9 205773             JSR RAM_PLUS       ;POINTER +1
0155 0070DC CA                 DEX
0156 0070DD 10F0               BPL RAMLIST2       ;IMMER 16 ITEMS AUSGEBEN
0157 0070DF 20F0E9             JSR CRLF
0158 0070E2 38                 SEC
0159 0070E3 A5F5               LDA RAMLFP         ;ENDE?
0160 0070E5 E5F3               SBC RAMHIGH
0161 0070E7 A5F6               LDA RAMLFP+1
0162 0070E9 E5F4               SBC RAMHIGH+1
0163 0070EB 90D1               BCC RAMLIST1
0164 0070ED 4C0F70             JMP RE_ENTER       ;FERTIG
```

```
0166 0070F0 A980      LISTEN    LDA #$80          ;MERKE MODUS
0167 0070F2 85FA                STA MODE
0168 0070F4           CHECKEN                     ;SEHEN, OB PROM LEER IST
0169 0070F4 207373              JSR PROM_VON
0170 0070F7 208873              JSR PROM_BIS      ;ANFANG UND ENDE?
0171 0070FA 209D73              JSR PULS_UP
0172 0070FD 200F73              JSR READ_INT
0173 007100 A000                LDY #0            ;RELATIVER ZEIGER
0174 007102 A200      CHECKE    LDX #0            ;SPALTENZAEHLER
0175 007104 E000      CHECKENX  CPX #0            ;VORSPALTE AUSGEBEN?
0176 007106 D013                BNE CHECKEN0
0177 007108 24FA                BIT MODE
0178 00710A 100F                BPL CHECKEN0      ;NEIN, WENN AUF $FF GEPRUEFT
                                    WIRD
0179 00710C A5EB                LDA COUNTER+1     ;ADRESSE AUSGEBEN
0180 00710E 2046EA              JSR NUMA
0181 007111 A5EA                LDA COUNTER
0182 007113 2046EA              JSR NUMA
0183 007116 A920                LDA #SPACE
0184 007118 207AE9              JSR OUTPUT
0185 00711B 20D873    CHECKEN0  JSR WAIT_ABIT     ;KONSOLIDIEREN
0186 00711E AD01A0              LDA PORTA         ;EINGABEWERT
0187 007121 24FA                BIT MODE
0188 007123 3007                BMI CHECKENL      ;LISTEN, NICHT VERGLEICHEN
0189 007125 C9FF                CMP #$FF          ;LEER?
0190 007127 F015                BEQ CHECKEN1
0191 007129 4CA370              JMP LESEN2D       ;FEHLER
0192 00712C 2046EA    CHECKENL  JSR NUMA          ;AUSGABE BYTE
0193 00712F A920                LDA #SPACE
0194 007131 207AE9              JSR OUTPUT
0195 007134 E8                  INX
0196 007135 E010                CPX #16           ;16 ZEICHEN DARGESTELLT?
0197 007137 D005                BNE CHECKEN1
0198 007139 A200                LDX #0            ;NEU INITIALISIEREN
0199 00713B 20F0E9              JSR CRLF
0200 00713E A5EB      CHECKEN1  LDA COUNTER+1     ;ENDBEDINGUNG ERREICHT?
0201 007140 C5EF                CMP PROMHIGH+1
0202 007142 D00E                BNE CHECKEN2
0203 007144 A5EA                LDA COUNTER
0204 007146 C5EE                CMP PROMHIGH
0205 007148 D008                BNE CHECKEN2
0206 00714A A90B                LDA #11
0207 00714C 202D74              JSR TEXT_OUT      ;PROM IST LEER
0208 00714F 4C0F70              JMP RE_ENTER
0209 007152 20C473    CHECKEN2  JSR PULS_PLUS
0210 007155 4C0471              JMP CHECKENX
0211
0212 007158 A90C      FEHLER    LDA #12           ;ADRESSBEREICH UNKLAR
0213 00715A 202D74              JSR TEXT_OUT
0214 00715D 4C0F70    FEHLER1   JMP RE_ENTER      ;EINGABEN WIEDERHOLEN
0215
0216 007160 203973    PROGRAM   JSR RAM_VON       ;EPROM PROGRAMMIEREN
0217 007163 205E73              JSR RAM_BIS
0218 007166 202F73              JSR RAM_ERR?
0219 007169 90ED                BCC FEHLER
0220 00716B 207373              JSR PROM_VON
0221 00716E 208873              JSR PROM_BIS      ;BEREICHSABFRAGEN
0222 007171 202573              JSR PROM_ERR?
0223 007174 90E2                BCC FEHLER
0224 007176 38                  SEC               ;PRUEFUNG GLEICHER MENGEN
0225 007177 A5F3                LDA RAMHIGH
0226 007179 E5F1                SBC RAMLOW
0227 00717B 85F8                STA MESPTR
0228 00717D A5F4                LDA RAMHIGH+1
0229 00717F E5F2                SBC RAMLOW+1
0230 007181 85F9                STA MESPTR+1      ;ABLAGE ZUM VERGLEICH
0231 007183 90D3                BCC FEHLER        ;FALSCH ADRESSIERT
0232 007185 38                  SEC               ;NUN MENGE PROM-BYTES
0233 007186 A5EE                LDA PROMHIGH
0234 007188 E5EC                SBC PROMLOW
0235 00718A 85F5                STA RAMLFP        ;ABLAGE ZUM VERGLEICH
0236 00718C A5EF                LDA PROMHIGH+1
0237 00718E E5ED                SBC PROMLOW+1
0238 007190 85F6                STA RAMLFP+1      ;Z. VERGLEICH
0239 007192 90C4                BCC FEHLER
```

```
0240 007194 C5F9              CMP MESPTR+1
0241 007196 D0C0              BNE FEHLER
0242 007198 A5F5              LDA RAMLFP
0243 00719A C5F8              CMP MESPTR          ;DIE ANDERE MENGE
0244 00719C D0BA              BNE FEHLER
0245 00719E A90A              LDA #10             ;ABFRAGE, OB SICHER
0246 0071A0 202D74            JSR TEXT_OUT
0247 0071A3 2040EC            JSR GETKEY
0248 0071A6 C94A              CMP #'J'            ;JA?
0249 0071A8 D0B3              BNE FEHLER1         ;NEIN
0250
0251 0071AA 209D73            JSR PULS_UP         ;ZAEHLERVORLAUF
0252 0071AD 204E73            JSR RAMLFPINT       ;LAUFPOINTER RAM EINSTELLEN
0253 0071B0 A90F              LDA #15             ;ABFRAGE OB NORMAL ODER INTEL
                                LIGENT BRENNEN
0254 0071B2 202D74            JSR TEXT_OUT
0255 0071B5 2040EC            JSR GETKEY
0256 0071B8 C949              CMP #'I'
0257 0071BA F002              BEQ PROGR1          ;INTELLIEGENT BRENNEN
0258 0071BC C6FA              DEC MODE            ;NEGATIV MACHEN ALS FLAG
0259
0260 0071BE A902      PROGR1  LDA #2              ;21 VOLT EINSCHALTEN
0261 0071C0 202D74            JSR TEXT_OUT
0262 0071C3 2040EC            JSR GETKEY          ;BESTAETIGUNG?
0263 0071C6 C90D              CMP #CR
0264 0071C8 D0F4              BNE PROGR1
0265 0071CA A90D              LDA #13             ;ANZEIGE, DASS GEBRANNT WIRD
0266 0071CC 202D74            JSR TEXT_OUT
0267 0071CF A000              LDY #0
0268 0071D1 AD04A0            LDA TI_LLOW         ;INTERRUPT-FLAG VORSORGLICH L
                                OESCHEN
0269 0071D4 84F0              STY IMPULSE         ;=00
0270 0071D6 84FB              STY ERRORS
0271 0071D8 84FC              STY ERRORS+1        ;FEHLERZAHL=0
0272
0273 0071DA 24FA              BIT MODE            ;WELCHER ALGORITHMUS?
0274 0071DC 1003              BPL PROGR2          ;INTELLIGENT
0275 0071DE 4CA372            JMP NPROGR          ;NORMAL
0276 0071E1 201473    PROGR2  JSR GEN_A12         ;HOHE ADRESSE ANLEGEN
0277 0071E4 A900              LDA #0
0278 0071E6 8D03A0            STA DDRA            ;ABKUERZEN, WENN BYTE SCHON G
                                LEICH
0279 0071E9 AD01A0            LDA PORTA           ;BZW. WENN $FF GEBRANNT WERDE
                                N SOLL
0280 0071EC D1F5              CMP (RAMLFP),Y
0281 0071EE F060              BEQ PROG04          ;BRENNEN UEBERGEHEN
0282 0071F0 20D973            JSR BURN_1MS        ;BRENNIMPULS ERZEUGEN
0283 0071F3 F00B              BEQ OVER_BURN       ;DATA SITZT SCHON FEST
0284 0071F5 A5F0              LDA IMPULSE         ;PRUEFEN, OB EPROM GUT IST
0285 0071F7 C90F              CMP #15             ;LIMIT NOCH NICHT ERREICHT
0286 0071F9 90E6              BCC PROGR2          ;+ 1 SCHUSS
0287 0071FB A90E              LDA #14             ;HINWEIS: EPROM SCHADHAFT
0288 0071FD 202D74            JSR TEXT_OUT
0289 007200           OVER_BURN                   ;NUN 4X SUMME BISHERIGER ZEIT
                                BRENNEN
0290 007200 A9FF              LDA #$FF            ;PORTA ZUM AUSGANG MACHEN
0291 007202 8D03A0            STA DDRA
0292 007205 B1F5              LDA (RAMLFP),Y      ;BYTE HOLEN ZUM BRENNEN
0293 007207 8D01A0            STA PORTA           ;DATENPORT
0294 00720A 20D873            JSR WAIT_ABIT       ;SIGNALE KONSOLIDIEREN
0295 00720D A980              LDA #$80            ;ENABLE PB7  ONE-SHOT PULSAUS
                                GANG
0296 00720F 8D0BA0            STA ACR
0297 007212 A5F0              LDA IMPULSE         ;NUN ERFORDERL. PULSWEITE
0298 007214 0A                ASL A               A ;X2
0299 007215 0A                ASL A               A ;NUN X4 ;FUER TYP 27128
0300 007216 AA                TAX                 ;NUN PRIMITIVE MULTIPLIKATION
0301 007217 A9E8              LDA #$E8            ;DEZIMAL 1000
0302 007219 85F8              STA MESPTR          ;HILFSZAEHLER
0303 00721B A903              LDA #03
0304 00721D 85F9              STA MESPTR+1
0305 00721F CA        MUL     DEX                 ;WEIL WIR VORGELADEN HABEN
0306 007220 F010              BEQ MUL_END
```

```
0307 007222 18              CLC
0308 007223 A5F8            LDA MESPTR          ;ZWISCHENERGEBNIS
0309 007225 69E8            ADC #$E8
0310 007227 85F8            STA MESPTR
0311 007229 A5F9            LDA MESPTR+1
0312 00722B 6903            ADC #$03            ;ZUSAMMMEN DEZ. 1000
0313 00722D 85F9            STA MESPTR+1
0314 00722F 4C1F72          JMP MUL
0315 007232 A5F8  MUL_END   LDA MESPTR          ;NUN ALS LADUNG IN DEN TIMER
0316 007234 8D04A0          STA T1_LLOW
0317 007237 A5F9            LDA MESPTR+1
0318 007239 20F273          JSR BURNO           ;RESTZEIT EINBRENNEN
0319
0320 00723C F012            BEQ PROG04          ;SCHLUSSPRUEFUNG
0321 00723E A905  PROGF     LDA #5              ;PROM IST UNGLEICH
0322 007240 202D74          JSR TEXT_OUT
0323 007243 18              CLC
0324 007244 A5FB            LDA ERRORS          ;FEHLER +1
0325 007246 6901            ADC #1
0326 007248 85FB            STA ERRORS
0327 00724A A5FC            LDA ERRORS+1
0328 00724C 6900            ADC #0
0329 00724E 85FC            STA ERRORS+1
0330 007250 38    PROG04    SEC                 ;NUN ENDBEDINGUNG PRUEFEN
0331 007251 A5F5            LDA RAMLFP
0332 007253 E5F3            SBC RAMHIGH
0333 007255 A5F6            LDA RAMLFP+1
0334 007257 E5F4            SBC RAMHIGH+1
0335 007259 902A            BCC PROG_FORT       ;WEITER BRENNEN
0336 00725B D028            BNE PROG_FORT       ;DITO
0337 00725D A900            LDA #0
0338 00725F 201074          JSR INITALL         ;FERTIG, IMPULSAUSGABE UNTERB
                            INDEN
0339 007262 A903  PROG4     LDA #3              ;MELDUNG AN BEDIENER
0340 007264 202D74          JSR TEXT_OUT        ;21 VOLT ABSCHALTEN
0341 007267 2040EC          JSR GETKEY
0342 00726A C90D            CMP #CR
0343 00726C D0F4            BNE PROG4
0344 00726E A5FC            LDA ERRORS+1        ;FEHLERZAHL AUSGEBEN
0345 007270 2046EA          JSR NUMA
0346 007273 A5FB            LDA ERRORS
0347 007275 2046EA          JSR NUMA
0348 007278 A920            LDA #SPACE
0349 00727A 207AE9          JSR OUTPUT
0350 00727D A910            LDA #16             ;FEHLERTEXT
0351 00727F 202D74          JSR TEXT_OUT
0352 007282 4C0F70          JMP RE_ENTER        ;FERTIG, NUN ZUM MENUE
0353 007285 20C473  PROG_FORT JSR PULS_PLUS     ;PROM-ADRESSE +1
0354 007288 205773          JSR RAM_PLUS
0355 00728B A5EA            LDA COUNTER         ;LEBENSZEICHEN ALLE 256 BYTES
0356 00728D D00D            BNE PROG_FA         ;NEIN
0357 00728F A5EB            LDA COUNTER+1       ;ADRESSE AUSGEBEN
0358 007291 2046EA          JSR NUMA
0359 007294 A5EA            LDA COUNTER
0360 007296 2046EA          JSR NUMA
0361 007299 20F0E9          JSR CRLF            ;NEUE ZEILE
0362
0363 00729C 24FA  PROG_FA   BIT MODE            ;NORMAL/INTELLIGENT?
0364 00729E 3003            BMI NPROGR          ;NORMAL
0365 0072A0 4CE171  PROG_FI JMP PROGR2          ;WEITERES BYTE BRENNEN
0366
0367 0072A3 20DB73  NPROGR  JSR WAIT_ABIT       ;NORMAL 50 MS PROGRAMMIEREN
0368 0072A6 201473          JSR GEN_A12         ;HOHE ADRESSE ANLEGEN
0369 0072A9 A900            LDA #0
0370 0072AB 8D03A0          STA DDRA            ;ABKUERZEN, WENN BYTE SCHON G
                            LEICH
0371 0072AE AD01A0          LDA PORTA           ;BZW. WENN $FF GEBRANNT WERDE
                            N SOLL
0372 0072B1 D1F5            CMP (RAMLFP),Y
0373 0072B3 F09B            BEQ PROG04          ;BYTE IST SCHON GLEICH, SKIP
                            OVER
0374 0072B5 A980            LDA #$80            ;ONE SHOT MODE AN PB7
0375 0072B7 8D0BA0          STA ACR
0376 0072BA A9FF            LDA #$FF            ;PORTA ALS AUSGANG
```

```
0377 0072BC 8D03A0              STA DDRA
0378 0072BF 91F5                LDA (RAMLFP),Y      ;BYTE AUF DEN PORT
0379 0072C1 8D01A0              STA PORTA
0380 0072C4 A950                LDA #<PULS_50MS
0381 0072C6 8D04A0              STA T1_LLOW         ;LATCH VORLADEN
0382 0072C9 A9C3                LDA #>PULS_50MS
0383 0072CB 20F273              JSR BURN0           ;BRENNIMPULS
0384 0072CE A940                LDA #$40            ;ONE SHOT MODE
0385 0072D0 8D0BA0              STA ACR             ;PB7 DISABLED
0386 0072D3 A9C3                LDA #>PULS_50MS     ;NUN 50 MS NUR WARTEN
0387 0072D5 8D05A0              STA T1_CHIGH        ;TIMER STARTEN
0388 0072D8 2C0DA0  NPROGR1     BIT IFR             ;ABWARTEN
0389 0072DB 50FB                BVC NPROGR1
0390 0072DD AD04A0              LDA T1_LLOW         ;IFR-FLAG LOESCHEN
0391 0072E0 200A74              JSR BURN2           ;BYTE-VERGLEICH
0392 0072E3 D003                BNE NPROGR2         ;UNGLEICH
0393 0072E5 4C5072              JMP PROG04          ;GLEICH, ENDBEDINGUNG PRUEFEN
0394 0072E8 A5EB    NPROGR2     LDA COUNTER+1       ;ADRESSE ANZEIGEN
0395 0072EA 2046EA              JSR NUMA
0396 0072ED A5EA                LDA COUNTER         ;LOW
0397 0072EF 2046EA              JSR NUMA
0398 0072F2 4C3E72              JMP PROGF           ;FEHLERHINWEIS
0399
0400 0072F5 203973  OPEN_READ   JSR RAM_VON         ;EPROM INS RAM LESEN/VERGLEIC
                                    HEN
0401 0072F8 207373  OPEN_RD1    JSR PROM_VON
0402 0072FB 208873              JSR PROM_BIS
0403 0072FE 202573              JSR PROM_ERR?       ;BEREICH PRUEFEN
0404 007301 90F5                BCC OPEN_RD1        ;FEHLER
0405 007303 209D73  OPEN_RD2    JSR PULS_UP         ADRESSEN EINSTELLEN
0406 007306 204E73              JSR RAMLFPINT       ;RAM-ZEIGER INITIALISIEREN
0407 007309 200F73              JSR READ_INT        ;DATEN-LESEN EINRICHTEN
0408 00730C A000                LDY #0
0409 00730E 60                  RTS
0410 00730F A900    READ_INT    LDA #0              ;DATENPORT WIRD EINGANG
0411 007311 8D03A0              STA DDRA            ;DATENRICHTUNG
0412 007314 A5EB    GEN_A12     LDA COUNTER+1       LEGE HIGH ADDRESS AUF PORTB
0413 007316 2930                AND #%00110000      ;A13, A12 ISOLIEREN
0414 007318 85F7                STA TEMP            ;ZWISCHENSPEICHERN
0415 00731A AD00A0              LDA PORTB
0416 00731D 29CF                AND #%11001111      ;ALTE A13, A12 LOESCHEN
0417 00731F 05F7                ORA TEMP            ,NEUE UEBERNEHMEN
0418 007321 8D00A0              STA PORTB
0419 007324 60                  RTS
0420
0421 007325 38      PROM_ERR?   SEC
0422 007326 A5EE                LDA PROMHIGH        ;ADRESSEINGABEN PRUEFEN
0423 007328 E5EC                SBC PROMLOW         ;PROMHIGH MUSS >= PROMLOW SEI
                                    N
0424 00732A A5EF                LDA PROMHIGH+1
0425 00732C E5ED                SBC PROMLOW+1
0426 00732E 60                  RTS
0427 00732F 38      RAM_ERR?    SEC                 ;ADRESSEN PRUEFEN
0428 007330 A5F3                LDA RAMHIGH
0429 007332 E5F1                SBC RAMLOW
0430 007334 A5F4                LDA RAMHIGH+1
0431 007336 E5F2                SBC RAMLOW+1
0432 007338 60                  RTS
0433 007339 A908    RAM_VON     LDA #8              ;PROMPT AUSGEBEN
0434 00733B 202D74              JSR TEXT_OUT
0435 00733E 20A3E7              JSR FROM            ;HOLE RAM-STARTADRESSE
0436 007341 AD1CA4              LDA ADDR
0437 007344 AE1DA4              LDX ADDR+1
0438 007347 85F1                STA RAMLOW
0439 007349 86F2                STX RAMLOW+1
0440 00734B 4CF0E9              JMP CRLF            RTS DORT
0441 00734E A5F1    RAMLFPINT   LDA RAMLOW
0442 007350 85F5                STA RAMLFP          ;RAM-LAUFPOINTER AUF ANFANG S
                                    ETZEN
0443 007352 A5F2                LDA RAMLOW+1
0444 007354 85F6                STA RAMLFP+1
0445 007356 60                  RTS
0446 007357 E6F5    RAM_PLUS    INC RAMLFP
0447 007359 D002                BNE RAMLFPAD1
```

```
0448 00735B E6F6            INC RAMLFP+1
0449 00735D 60       RAMLFPAD1 RTS
0450
0451 00735E A909     RAM_BIS   LDA #9
0452 007360 202D74             JSR TEXT_OUT
0453 007363 20A7E7             JSR TO              ;BEGRENZER-ADRESSE F. RAM HOL
                                  EN
0454 007366 AD1CA4             LDA ADDR
0455 007369 AE1DA4             LDX ADDR+1
0456 00736C 85F3               STA RAMHIGH
0457 00736E 86F4               STX RAMHIGH+1
0458 007370 4CF0E9             JMP CRLF
0459 007373 A906     PROM_VON  LDA #6
0460 007375 202D74             JSR TEXT_OUT
0461 007378 20A3E7             JSR FROM            ;STARTADRESSE PROM HOLEN
0462 00737B AD1CA4             LDA ADDR
0463 00737E AE1DA4             LDX ADDR+1
0464 007381 85EC               STA PROMLOW
0465 007383 86ED               STX PROMLOW+1
0466 007385 4CF0E9             JMP CRLF
0467 007388 A907     PROM_BIS  LDA #7              ;PROMPT
0468 00738A 202D74             JSR TEXT_OUT
0469 00738D 20A7E7             JSR TO              ;ENDADRESSE PROM HOLEN
0470 007390 AD1CA4             LDA ADDR
0471 007393 AE1DA4             LDX ADDR+1
0472 007396 85EE               STA PROMHIGH
0473 007398 86EF               STX PROMHIGH+1
0474 00739A 4CF0E9             JMP CRLF
0475
0476 00739D          PULS_UP                        ;IMPULSE AUSGEBEN, BIS PROM-S
                                  TARTADRESSE ERREICHT
0477 00739D A9CC               LDA #CNT_EIN+READ_EIN;LESEN EPROM ENABLE
0478 00739F 8D0CA0             STA PCR
0479 0073A2 38       PULS_UP1  SEC
0480 0073A3 A5EA               LDA COUNTER
0481 0073A5 E5EC               SBC PROMLOW
0482 0073A7 A5EB               LDA COUNTER+1
0483 0073A9 E5ED               SBC PROMLOW+1
0484 0073AB F016               BEQ PULS_END
0485 0073AD AD00A0             LDA PORTB
0486 0073B0 4940               EOR #BIT6           ;ABFALLENDE FLANKE ERZEUGEN
0487 0073B2 8D00A0             STA PORTB
0488 0073B5 E6EA               INC COUNTER         ,NUTZE ZWISCHENZEIT
0489 0073B7 D002               BNE PULS_UP2
0490 0073B9 E6EB               INC COUNTER+1
0491 0073BB 4940     PULS_UP2  EOR #BIT6           ;AUFSTEIGENDE FLANKE
0492 0073BD 8D00A0             STA PORTB
0493 0073C0 4CA273             JMP PULS_UP1
0494 0073C3 60       PULS_END  RTS
0495
0496 0073C4 AD00A0   PULS_PLUS LDA PORTB
0497 0073C7 4940               EOR #BIT6           ;ABFALLENDE FLANKE ERZEUGEN
0498 0073C9 8D00A0             STA PORTB
0499 0073CC E6EA               INC COUNTER         ,NUTZE ZWISCHENZEIT
0500 0073CE D002               BNE PULS_PL2
0501 0073D0 E6EB               INC COUNTER+1
0502 0073D2 4940     PULS_PL2  EOR #BIT6           ;AUFSTEIGENDE FLANKE
0503 0073D4 8D00A0             STA PORTB
0504 0073D7 60                 RTS
0505
0506 0073D8 60       WAIT_ABIT RTS                 ;KONSOLIDIERUNG
0507
0508 0073D9 A9FF     BURN_1MS  LDA #$FF            ;PORTA ZUM AUSGANG MACHEN
0509 0073DB 8D03A0             STA DDRA
0510 0073DE B1F5               LDA (RAMLFP),Y      ;BYTE HOLEN ZUM BRENNEN
0511 0073E0 8D01A0             STA PORTA           ;DATENPORT
0512 0073E3 20D873             JSR WAIT_ABIT       ;SIGNALE KONSOLIDIEREN
0513 0073E6 A980               LDA #$80            ;ENABLE PB7  ONE-SHOT PULSAUS
                                  GANG
0514 0073E8 8D0BA0             STA ACR
0515 0073EB A9E8               LDA #<PULS_1MS      ;TIMER 1 VORLADEN, LOW
0516 0073ED 8D04A0             STA T1_LLOW         ;INS LATCH
0517 0073F0 A903               LDA #>PULS_1MS      ;NUN PULSAUSGABE STARTEN
0518 0073F2 8D05A0   BURNO     STA T1_CHIGH        ;TIMER STARTEN
```

```
0519 0073F5 2C0DA0    BURN1     BIT IFR             ;WANN KOMMT DAS INTERRUPT-FLA
                                  G?
0520 0073F8 50FB                BVC BURN1           ;NOCH NICHT
0521 0073FA AD04A0              LDA T1_LLOW         ;INTERRUPT-FLAG BESEITIGEN
0522 0073FD E6F0                INC IMPULSE
0523 0073FF 20DB73              JSR WAIT_ABIT       ;KONSOLIDIEREN
0524 007402 A900                LDA #0              ;PORTA LESBAR MACHEN
0525 007404 8D03A0              STA DDRA
0526 007407 8D0BA0              STA ACR             ;PB7-ENABLE AUFHEBEN
0527 00740A AD01A0    BURN2     LDA PORTA           ;HOLEN, WAS SCHON ERZEUGT WUR
                                  DE
0528 00740D D1F5                CMP (RAMLFP),Y      ;VERGLEICH MIT DER VORLAGE
0529 00740F 60        BURNEND   RTS
0530
0531                            ;INITIALISIERUNG DER INTERFACES
0532 007410 A9FF      INITALL   LDA ##$FF
0533 007412 8D00A0              STA PORTB
0534 007415 8D01A0              STA PORTA
0535 007418 8D02A0              STA DDRB            ;DATENRICHTUNG=AUSGANG
0536 00741B 8D03A0              STA DDRA
0537 00741E A9EE                LDA ##$EE           ;CA2 UND CB2 HIGH SETZEN, SIG
                                  NALE RESET, OUTPUT EN
0538 007420 A900                LDA #00             ;PULSMODE AN PB7 AUSSCHALTEN
0539 007422 8D0BA0              STA ACR
0540 007425 8D0CA0              STA PCR             ;RESET COUNTER, LESEN EPROM S
                                  PERREN
0541 007428 85EA                STA COUNTER         ;ZAEHLER ZURUECKSETZEN
0542 00742A 85EB                STA COUNTER+1
0543 00742C 60                  RTS
0544
0545                            ;AUSGABE EINES TEXTES, DIE TEXT-NUMMER MUSS IM AK
KU SEIN
0546                            ;TEXTBEGRENZUNG DURCH 00
0547
0548 00742D 0A        TEXT_OUT  ASL A               A ;MULTIPLIKATION NUMMER IM A
                                  KKU MIT 2
0549 00742E AA                  TAX
0550 00742F BD5A77              LDA MESTAF,X        ;POINTER ZUR MESSAGE ZURECHTL
                                  EGEN
0551 007432 85F8                STA MESPTR
0552 007434 BD5B77              LDA MESTAF+1,X
0553 007437 85F9                STA MESPTR+1        ;MESSAGE-POINTER
0554 007439 A000                LDY #0
0555 00743B B1F8      TXT1      LDA (MESPTR),Y      ;ZEICHEN HOLEN
0556 00743D F009                BEQ TXTEND          ;NULL ALS BLOCKBEGRENZER
0557 00743F 297F                AND ##$7F           ;REINES ASCII HERSTELLEN
0558 007441 207AE9              JSR OUTPUT          ;ZEICHENAUSGABE
0559 007444 C8                  INY                 ;POSITIONIERUNG AUF FOLGENDES
                                   BYTE
0560 007445 4C3B74              JMP TXT1
0561 007448 A000      TXTEND    LDY #0              VORSORGLICH
0562 00744A 60                  RTS
0563
0564                            ;MESSAGES
0565 00744B 50524F4D  MES1      .BYT 'PROMMER FUER 2764 UND 27128, R. LOEHR',CR
0565 007470 0D
0566 007471 5350414E            .BYT 'SPANNUNG +5 VOLT EINGESCHALTET?',CR
0566 007490 0D
0567 007491 32312056            .BYT '21 VOLT AUSGESCHALTET? CR=JA',CR,00
0567 0074AD 0D00
0568 0074AF 0D        MES2      .BYT CR,'HAUPTMENUE, WAHL MIT TASTE',CR
0568 0074B0 48415550
0568 0074CA 0D
0569 0074CB 45202045            .BYT 'E  EINLESEN PROM-INHALT INS RAM',CR
0569 0074EA 0D
0570 0074EB 4C20204C            .BYT 'L  LISTEN EINES PROM-BEREICHES',CR
0570 007509 0D
0571 00750A 52202052            .BYT 'R  RAM-BEREICH LISTEN',CR
0571 00751F 0D
0572 007520 56202056            .BYT 'V  VERIFIZIEREN PROM/RAM',CR
0572 007538 0D
0573 007539 43202043            .BYT 'C  CHECKEN OB LEER',CR
```

```
0573 00754B 0D
0574 00754C 50202050          .BYT 'P  PROGRAMMIEREN',CR
0574 00755C 0D
0575 00755D 41202041          .BYT 'A  ABWEICHUNGEN LISTEN?',CR
0575 007574 0D
0576 007575 51202051          .BYT 'Q  QUIT ZUM MONITORPROGRAMM',CR,00
0576 007590 0D00
0577 007592 32312056 MES3     .BYT '21 VOLT EINSCHALTEN     CR=EINSCHALTEN',CR
0577 0075B7 0D
0578 0075B8 4552484F          .BYT 'ERHOEHTE SPANNUNG 6 VOLT EINSCHALTEN',CR,0
                                 0
0578 0075DC 0D00
0579 0075DE 46455254 MES4     .BYT 'FERTIG, 21 VOLT AUSSCHALTEN CR=AUS',CR
0579 007600 0D
0580 007601 41554620          .BYT 'AUF 5 VOLT ZURUECKSCHALTEN !',CR,00
0580 00761D 0D00
0581 00761F 50524F4D MES5     .BYT 'PROM IST GLEICH',CR,CR,00
0581 00762E 0D0D00
0582 007631 0D        MES6    .BYT CR,'***** ACHTUNG: PROM IST UNGLEICH ****',
                                 CR,00
0582 007632 2A2A2A2A
0582 007657 0D00
0583 007659 41422050 MES7     .BYT 'AB PROM-ADRESSE?',CR,00,00
0583 007669 0D0000
0584 00766C 42495320 MES8     .BYT 'BIS PROM-ADRESSE?',CR,00
0584 00767D 0D00
0585 00767F 41422052 MES9     .BYT 'AB RAM-ADRESSE?',CR,00
0585 00768E 0D00
0586 007690 42495320 MES10    .BYT 'BIS RAM-ADRESSE?',CR,00
0586 0076A0 0D00
0587 0076A2 49535420 MES11    .BYT 'IST DAS SICHER? J=JA',CR,00
0587 0076B6 0D00
0588 0076B8 0D        MES12   .BYT CR,'PROM IST LEER',CR,00
0588 0076B9 50524F4D
0588 0076C6 0D00
0589 0076C8 0D        MES13   .BYT CR,'ADRESSBEREICHE UNKLAR !',CR,00
0589 0076C9 41445245
0589 0076E0 0D00
0590 0076E2 0D        MES14   .BYT CR,' *** ICH BRENNE, BITTE WARTEN ***',CR,0
                                 0
0590 0076E3 202A2A2A
0590 007704 0D00
0591 007706 2A2A2A20 MES15    .BYT '*** EPROM IST SCHADHAFT ***',CR,00
0591 007721 0D00
0592 007723 0D        MES16   .BYT CR,'50 MS BRENNEN = CR, INTELLIGENT = I',CR
                                 ,0
0592 007724 3530204D
0592 007747 0D00
0593 007749 20204641 MES17    .BYT '  FATALE FEHLER',CR,00
0593 007758 0D00
0594
0595                          ;WORTADRESSEN DER TEXTBEGINNE
0596 00775A           MESTAF                    ;MESSAGE-TAFEL
0597 00775A 4B74AF74          .WOR MES1,MES2,MES3,MES4,MES5,MES6
0597 00775E 9275DE75
0597 007762 1F763176
0598 007766 59766C76          .WOR MES7,MES8,MES9,MES10,MES11,MES12,MES13,MES1
                                 4,MES15
0598 00776A 7F769076
0598 00776E A276BB76
0598 007772 C876E276
0598 007776 0677
0599 007778 23774977          .WOR MES16,MES17
0600 00777C 454C5256 BEFEHLE  .BYT 'ELRVCPAQ'     ;ERLAUBTE BEFEHLSTASTEN
0601
0602 007784 5A70F070 BEFTAB   .WOR EINLESEN,LISTEN,RAMLIST,VERIFY,CHECKEN,PROG
                                 RAM
0602 007788 B3705670
0602 00778C F4706071
0603 007790 4C704970          .WOR ABWEICH,QUIT  ;ADRESSEN DER TEILPROGRAMME
0604 007794                   .END
FEHLERZAHL INSGESAMT: 0000
```

Roland Löhr

PC-128

Der neue Rechner wurde dem Fachhandel bereits Ende Januar 1985 vorgestellt, an diesen jedoch erst ab etwa Ende Oktober ausgeliefert, nachdem es bereits eine heftige Verstimmung gegeben hatte, weil vor allem der Kaufhof bereits ab Ende Juli beliefert wurde. Man kann auch nicht ausschließen, daß die ersten Geräte noch Eigenheiten hatten, die zu Rücksendungen führten. Berichte über den PC-128 füllen seit einiger Zeit die Zeitschriften, so richtig hat man seine besonderen Fähigkeiten aber wohl noch nicht ganz erkannt und auch noch nicht ausgenutzt. Er ist noch zu neu, und dazu bedarf es der Zeit.

Ein Blick zurück auf Commodore: In wenigen Jahren entwickelte man aus dem PET 2001 schließlich den CBM 720. Dieser, wie auch andere Eigenentwicklungen, z.B. der 8296, sind in den Augen des Herstellers nicht mehr aktuell. Der C-64 wurde ein millionenfacher Erfolg vor allem bei jungen Menschen, seine für diverse andere Zielgruppen entwickelten Abkömmlinge setzen sich nur wenig durch. Der PC-128 soll nun ein Modell für 'Aufsteiger' sein und den inzwischen gehobenen Ansprüchen des Marktes im Übergangsbereich zwischen Heim- und Personal Computern entsprechen. - Bei CBM finden wir andererseits die Erwartung, daß man mit den kompatiblen Geräten PC 10 und PC 20, die sich gut verkaufen, weiterhin Geld verdient und daß man schließlich mit dem kommenden AMIGA, auf den wir in diesem Heft auch eingehen, einen neuen Standard in Leistung und Preis setzt. - Bei den typischen Eigenentwicklungen von Commodore haben wir in der nächsten Zeit neben den 'kompatiblen' folgende Haupt-Geräte: C-64 (unter DM 500), PC-128, etwa DM 1000, und AMIGA, den man wohl um die DM 5000 preislich erwarten sollte. Dieweil er auch den PC 10 (IBM-PC) per Software nachfahren können soll, muß man dann zugleich einen Preisrutsch bei solchen kompatiblen Computern erwarten. - Dieses im Umfeld eines umkämpften Marktes, in dem ja auch noch IBM, Apple und Atari tätig sind, bei den Heimcomputern auch noch Schneider.

Die reine Rechenleistung eines Computers ist heute nicht mehr sein wichtigstes Merkmal. Inzwischen geht es mehr um Graphik und Sound, und daß man diese Dinge mit wenig Programmieraufwand erzeugen kann. Wenn man auf die früheren Messen in Hannover und München zurückblickt, bei denen der monochrome Bildschirm für Textverarbeitung erst im Laufe der Jahre eine Selbstverständlichkeit wurde, so kann man jetzt nach der SYSTEMS 85 in München sagen, daß der farbige hochauflösende Bildschirm sich nunmehr durchsetzt. Es geht um computer-gestützte Entwurfs- und Darstellungstechniken, für die z.T. sehr hohe Auflösungen notwendig sind. Die Erzeugung von Geräusch, Musik und Sprache zielt einerseits auf Unterhaltung/Spiele mit dem Computer, andererseits auf die Erwartung, daß man mit dem Computer etwas lehren und erklären kann und daß man andererseits wohl eine bessere Bedienerführung erreicht. Insofern liegt der PC-128 im Trend der Entwicklung, von der man nicht genau weiß, wie sie im einzelnen im Publikum einen breiten Markt findet.

In diesen Jahren, wo die visuellen Medien (TV, Videorekorder, Verkabelung, Satellit, Btx) sich um eine größere Verbreitung und Einflußnahme bemühen, bleibt auch der Computer als 'visuelles' Gerät nicht vom Trend unbeeinflußt. Er kann mit Btx korrespondieren. Allein das Publikum spricht diesem Medium nicht in dem Maße zu, wie es sich der Postminister und Informationsanbieter vorstellen, weswegen letztere z.T. schon zum Rückzug blasen. - Der Computer, auch ein PC-128 oder AMIGA, ist damit in der heutigen Form noch nicht das Gerät, das in wenigen Jahren die ganz große Verbreitung, als 'vernetztes Benutzergerät' haben wird.

In Heft 44 erfolgten bereits erste Hinweise zum PC-128 mit dem ersten Befund

'gut gemacht'. Wir betrachten ihn jetzt weiter. Seine Hauptmerkmale dürften bekannt sein: Formschöne Computer-Konsole mit deutscher Tastaturbelegung (wahlweise ASCII), Zehnerblock, Cursour-Tasten und Funktionstasten. 48 KB Festwertspeicher mit Betriebssystem und dem leistungsfähigen BASIC 7.0, 128 KB RAM als Arbeitsspeicher, Maschinentakt per FAST auf 2 MHz einstellbar, 3 Video-Ausgänge für Fernseher oder Monitor (Composite Video) mit 40 Zeichen/Zeile, ein Ausgang für 80 Zeichen Monitor mit RGBI-Signalen. Schnittstellen RS 232 mit TTL-Pegeln, serieller Gerätebus, User-Port und zwei Spiele-Ports.

Die wesentlichen Funktionsbausteine des Computers sind die CPU 8502 mit dem Befehlssatz des 6502, eine CPU Z80B mit 4 MHz Takt (effektiv auf ca. 2,5 MHz gebremst), eine Memory Management Unit 8722 mit interessanten Merkmalen und mit angeschlossenem Adreß-Modifier-Chip 8721, die bekannte Sound Interface Device (SID 6581) und gleich zwei Video-Chips. Der VIC 8564 entspricht dem VIC des C-64, ergänzt um zwei Register. Er kann nur im Takt 1 MHz gefahren werden und nur mit 40 Zeichen je Zeile. Er hat ein Video-RAM im allgemeinen Speicher, das vom RAM des zweiten Video-Chips unabhängig ist. Aus diesem Grunde kann der PC-128 zwei Bildschirminhalte zugleich ausgeben, wenn er im Betrieb mit 1 MHz ist. Die neuen graphischen Befehle des BASIC 7 unterstützen diesen CHIP, der auch einen geteilten Bildschirm Graphik/Text ausgeben kann. Der zweite Video-Chip wird 8563 VDC genannt. Er kann mit 2 MHz arbeiten, erzeugt 80 Zeichen je Zeile, ist farbtüchtig und kann sogar auch Punktgrafik mit 640x200 Pixels erzeugen, was jedoch von der Firmware nicht unterstützt wird. Dieser Chip verwaltet ein eigenes Video-RAM, das dem direkten Zugriff des Prozessors entzogen ist. Stattdessen muß man wie beim CRT-Controller 6845 einen 'Trichter' aufsetzen, wenn man die verschiedenen Register (37) und das RAM erreichen will. Es gibt ein Vorauswahl- und ein Datenregister.

Die Neuerungen in der Hardware sind also im wesentlichen: Betrieb 1 oder 2 MHz, 2 unabhängige Bildschirme mit Farbausgabe und eine leichte Konfigurationsmöglichkeit für RAM/ROM und E/A durch die Memory Management Unit und ihre voreinstellbaren Register. Diese erlaubt es auch, für Speicherbänke gemeinsame Bereiche von 1 bis 16 KB zu definieren, die an den unteren oder oberen oder an beiden Adreßbereichen liegen. Insgesamt werden damit bis 32 KB gemeinsame Parameter möglich. Und: Man kann der MMU vorschreiben, ab welcher physikalischen Adresse die logische Zero Page und unabhängig davon, wo der logische Stackbereich liegen soll. Der Adreß-Modifier setzt das um. Der Assembler-Programmierer erkennt, daß damit elegante Programmiertechniken möglich werden, deren Stil sich hinsichtlich Sicherheit und Effizienz sicher erst noch herausbilden muß. Gleichwohl sollte man sich dazu schon einmal Gedanken machen.

Die Neuerungen in der Software betreffen vor allem das BASIC 7 und den CP/M-Betrieb mit der CPU Z80B, auf die wir weiter eingehen. Der auch mögliche C-64 Betrieb mit einem großen vorhandenen Programmangebot soll nur erwähnt werden. Dazu ist nur zu bemerken, daß in ihm die Sondertastenfelder totgelegt sind. Es verbleiben die belegbaren Funktionstasten.

BASIC 7 erlaubt zunächst einmal die strukturierte Programmierung, die noch über den Konfort im BASIC des 720 hinausgeht. Folgende Konstrukte sind möglich: IF .. BEGIN .. BEND, IF .. THEN .. ELSE, DO .. UNTIL, DO .. WHILE, DO .. LOOP.

Gleichlautend mit BASIC 4 sind die Diskettenbefehle Append, Backup, Bload, Bsave (binär, auch verschieblich laden, abspeichern), Catalog bzw. Directory (zerstört nicht das Programm), Collect, Concat, Copy, Dclear, Dclose, Dopen, Dload, Dsave, Dverify (neu), Header, Rename, Scratch und Record für relative Dateien. Neu ist BOOT.

Der Editierung dienen Auto, Delete und Renumber. Hinzu tritt ein recht komfortabler Bildschirm-Editor über Escape-Sequenzen. In Anwendungsprogrammen kann mit Windows gearbeitet werden. FAST und SLOW beeinflussen den Maschinentakt. Der Fehlerbehandlung und -erkennung dienen ERR$, TRAP und HELP. Hinzu kommen zum Debuggen TRON und TROFF (Trace). FETCH, STASH und SWAP dienen dem Block-Move/Austausch zwischen Speicherbereichen. POINTER ergibt die Adresse einer Variablen. RREG weist nach einem mit SYS aufgerufenen Maschinenprogramm die Registerinhalte der CPU zu benennenden BASIC-Variablen zu. HEX$ und DEC erlauben den Umgang mit hexadezimalen Argumenten. - Mit diesen Möglichkeiten und Befehlen ist BASIC in sich transparenter geworden, es bietet mehr Leistung, auch bei der Niederschrift und Erprobung von Programmen. Erwähnenswert sind ferner für das Suchen die Instring-Funktion und für die Ausgabe das Print Using.

Die graphischen Möglichkeiten werden sehr durch Befehle unterstützt, es mögen etwa 28 sein. Die Befehle betreffen geometrische Figuren, Sprites und Farben sowie der Lokalisation von Gebilden. Diesen Befehlen können z.T. recht umfangreiche Parameterlisten beigefügt werden, die die Ausführungsart beeinflussen. Das Handbuch zum Rechner enthält die notwendige Syntax mit Beispielen. Das gilt auch für Geräusche und Sound mit den Befehlen Envelope, Filter, Play, Sound, Tempo, Vol (Lautstärke). Man kann eine Melodie nach Noten abspielen, die in einer Stringvariablen enthalten sind. - Wenn die hinter den graphischen Befehlen stehende Logik erst einmal von Tüftlern durchschaut ist, dann darf man auch für die hochauflösende Graphik des VDC 8563 beachtliche Programme und ggfs. Befehlserweiterungen erwarten.

Zur Transparenz des Betriebssystems und des BASIC ist zu bemerken, daß die bei BASIC 4 übliche Sprungleiste am Ende des Festwertspeichers auch hier enthalten ist, z.T. an den gleichen Adressen und mit den gleichen Funktionen, zu einem Teil aber auch um 1 Byte versetzt. Dieses Sprungleiste führt mit indirekten Sprüngen über Vektoren, die im RAM nach der Initialisierung vorbesetzt sind. Mit ihrer Abänderung kann man den Computer schnell in eine andere Umgebung führen, z.B. in eine andere Sprache oder in ein Textsystem. Die Beibehaltung dieser Einrichtung wird die für den PC-128 spezifische Programmierung sehr erleichtern. Und man kann sich auch vorstellen, daß dieser Computer weniger reichlich ausgestattete kleine Entwicklungssysteme, wie z.B. den AIM 65 sehr schnell obsolet macht.

Wir kommen damit auf das Monitor-Programm. Es wird mit dem Befehl MONITOR von BASIC her aufgerufen. Er bietet folgende Funktionen, die der Benutzer eines CBM oder des Monitorprogrammes Newtim im wesentlichen schon kennt: Assembler Line by Line, Disassembler, Speicherbereich füllen, Go (Ausführung), Hunt (Bytefolge suchen), Laden und Abspeichern, Memory-Dump und -veränderung, Registeranzeige, Block-Move und Disketten-Kurzbefehle. Damit ist eine schon recht komfortable Grundausrüstung gegeben.

Erste Versuche wurden hier auch schon mit dem Betrieb unter CP/M gefahren. Es wird als Diskette mit dem System zusammen geliefert. Man kann ferner eine Zusatzdiskette bestellen (DM 80), die hier allerdings nach Wochen noch nicht eingetroffen ist. Ein Handbuch von 129 Seiten erklärt die Befehle des CP/M. Die mitgelieferte Diskette ist auf zwei Seiten beschrieben und von einer Floppy 1541 lesbar. Man muß allerdings sagen, daß der Ladevorgang jeweils eine recht langweilige Angelegenheit ist. Zum Kennenlernen aber reicht es. Die Diskette enthält die übliche Grundausstattung: CPM+, Console Command Processor, Help mit vielen Zusatz-Screens (ersetzt praktisch das kleine Handbuch), Keyfig (Tastaturbelegung), Format, Pip, Dir und Copysys zum Anlegen von Disketten, Kopieren. Auf der Rückseite finden folgende transiente Kommandos: Device zur Zuordnung, Dir mit Optionen, Dump, Ed (Editor), Erase, Rename, Submit und Type, wenn man von einigen Dateien zur Umlenkung der Ein-

und Ausgabe einmal absieht. Sobald die schnelleren Laufwerke 1570 und 1571 (kommt erst später) zur Verfügung stehen, wird das CP/M besser hantierbar, es bleibt aber abzuwarten, welchen Zuspruch es auf diesem Rechner noch finden wird.

In der Summe ist zu sagen, daß der PC-128 viele neue Möglichkeiten erschließt, die nicht nur im Bereich des gehobenen Heimcomputers liegen, sondern auch im professionellen Bereich, wenn an ihn ein guter geeigneter Monitor angeschlossen wird. Außer an Textverarbeitung, Rechengerät, Datenbank und kleines Entwicklungssystem ist auch an den Einsatz als Terminal oder im Rahmen lokaler Netzwerke und Benachrichtigungssysteme zu denken. Und er wird sicher auch im ausbildenden und belehrenden Sektor Anwendung finden. Er wird sich aber auch den Wettbewerb von Geräten gefallen lassen müssen, die etwa in der gleichen Preisklasse sind oder von solchen, die noch mehr können, dann allerdings schon in den Bereich der Personal Computer gehören. Wünsche bleiben natürlich selbst auch bei diesen offen. Der Markt entscheidet über die Akzeptanz, von der letztlich auch das Angebot von Zusatzplatinen und guter Software abhängt.

Bücher

Jamsa, Kris: The C Library. Verlag Osborne McGraw-Hill, Berkely, Kalifornien 1985, ISBN 0-07-881110-4, 294 Seiten, Softcover, Preis nicht bekannt. Der Autor ist Software Engineer für die US-Airforce. Es ist sein zweites Buch zu 'C' (nach Programming in C). Das Buch enthält 12 Abschnitte mit etwa 125 sorgfältig strukturierten und kommentieren Programmen in 'C'. Dabei werden folgende Themenbereiche abgedeckt: Übersicht zur Sprache, Konstanten und Makros, Stringverarbeitung/-bearbeitung, Pointer, E/A-Routinen, Arbeiten mit Arrays, Rekursion, Sortieren, Trigonometrische Funktionen und Zeichenumwandlung, Programme zur File-Bearbeitung, Programmierung der UNIX-Pipe, Anhang und Index. Das englische Buch ist sprachlich sehr klar geschrieben und dürfte eine wahre Fundgrube für den Aufbau immer nützlicher Tools für die Sprache 'C' sein. Als Programmierer mit dieser Sprache wird man auf die hier gegebenen Anregungen nicht verzichten dürfen. Der Text will dabei kein Lehrbuch für 'C' sein. Gleichwohl sind die enthaltenen Syntax-Diagramme und die weiteren Abbildungen informativ. Der Autor bittet den Leser, die enthaltenen Programmierbeispiele auszuprobieren. Das ist zwar etwas zeitaufwendig, es dürfte gleichwohl der geeignete Weg zum Verständnis der Sprache und der hier gegebenen Anregungen sein. - Mit dem IBM-PC, dem Atari 520 und mit dem kommenden Amiga von Commodore dürfte die Zahl der C-Interessierten schnell zunehmen. Ihnen darf diese Programmsammlung neben einem guten Lehrbuch sehr empfohlen werden.

Kane, Gerry: 68000 Mikroprozessorhandbuch. Verlag McGraw-Hill, Hamburg 1985, 135 S., Softcover, ISBN 3-89028-030-7, DM 35,--. Das Original dieses Buches erschien 1981. Inzwischen sind in der Familie 68xxx erhebliche Fortschritte gemacht worden. Daher war im Anhang C des Buches eine Aktualisierung auf 68010, 68012 und 68020 notwendig. - Der Rezensent muß sich für 'befangen' erklären, weil er für diesen Verlag die deutsche Übersetzung und auch die Aktualisierung besorgte. Gleichwohl: Das Buch ist gut gelungen. Wir finden die Abschnitte Einleitung, Überblick zur Funktionsweise, Zeitabläufe auf den Bussen, Verarbeitung von Ausnahmebedingungen, Die Adressierungsarten, Den Befehlsssatz, Schnittstellen mit Peripheriebausteinen 68xx, Tafeln zum Befehlssatz, Alphabetische Tabellen zu den Befehlsworten, Die neueren CPU-Typen des 68xxx. - Während der erste Teil des Buches vor allem das Handshake der Signale in einem System mit 68xxx verdeutlicht, und damit mehr hardware- und ablaufbezogen ist, dürfte der zweite Teil mit den Adressierungen und Befehlen vor allem dem Assembler-Programmierer nützlich sein. Er findet in deutscher Sprache deutliche Beschreibungen zur Wirkung der Befehle. Beim Studium dieser Aufstellungen dürfte klar werden, daß es beim 68xxx viele Feinheiten gibt, auf die man achten muß, wenn man nicht 'abstürzen' will. - Die Absicht des Autors Kane, ein übersichtliches kleines Nachschlagewerk für den Arbeitstisch zu schaffen, dürfte in geeigneter Form verwirklicht sein.

Stultz, Russel, A.: dBASE III, Einführung und Referenz. te-wi Verlag, München 1985, 464 Seiten, Sofcover, ISBN 3-921803-64-0, DM 79. Für Computer mit 8 Bit war dBASE II wohl das erfolgreichste Datenbankprogramm. Für solche mit 16 Bit (IBM-PC) wurde dBASE III geschaffen, womit auch hier eine große Flexibilität mit diesem Programm geschaffen worden ist.- Das Buch geht in seiner Stoffdarstellung einen ungewöhnlichen Weg: Wir finden 60 Darstellungs-Module. Man arbeitet sich jedoch nicht in ihrer Reihenfolge ein, sondern findet einen Leitfaden, wie man die Module (sie sind thematisch nach ihrer Funktion geordnet) in einer didaktisch aufbauenden Art mit zunehmender Komplexität lesen und erproben sollte. Der Leitfaden setzt damit in unterschiedliche Stellen des Buches auf. Jedes Modul enthält dabei Hinweise auf Auswirkungen und Bedingungen in anderen Modulen. Jedes Modul enthält unter einer deutlich lesbaren Überschrift mit den besprochenen Befehlen einen 'Beschreibungskasten' (eingerahmt) mit der Funktionsbeschreibung, danach weitere Ausführungen zur Übersicht, die Befehle nebst Anwendungsbeispielen, Übungen und Verständnisfragen. Die Anwendungsbeispiele mögen einige hundert sein, sie sind so dokumentiert, daß man seine Eingaben und die Antworten des Systems in den Listen verfolgen kann. Klar wird auch herausgestellt, wie man die Befehle im Direkt-Modus und per Kommandodatei benutzen kann. - Die ungewöhnliche Darstellung in Modulen hat den Vorteil, daß sie jeweils eine vollständige Dokumentation zu einem Einzelthema sein können. Damit ist die Funktion des Buches als 'Referenz' hergestellt. Der Leitfaden für die Einarbeitung ist davon unabhängig. - Ein Buch kann man am besten beurteilen, wenn man seine Ideen am Computer erprobt. Eine entsprechende Möglichkeit gibt es beim Rezensenten nicht, gleichwohl hat er nach der Beschreibung und Erprobung einer anderen Datenbank den Eindruck, daß dieses klar geschriebene Buch sehr hilftreich ist, um zum Aufbau gegliederter Datenbanken zu gelangen.

OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

Editorial

Nach der SYSTEMS 85 in München ist abzulesen: Dieser Herbst zeigt erstmals eine breite Akzeptanz des Marktes für 16 Bit-Systeme mit CPU 68000. Insbesondere ist hier Atari mit seinen neuen Modellen zu erwähnen. Und sicher wird der neue AMIGA von Commodore ab März 1986 einen guten Start haben, weil er wiederum zusätzliche Leistungen bringt. Indirekt hört man, daß er in den USA nach anfänglich $2000 bereits für $1400 zu haben sei. Mit einem solchen Preis dürfte er recht agressiv und günstig liegen.

Man kann das Vordringen neuer Computer-Ideen nur begrüßen, denn lange sah es fast so aus, als ab der architektonisch veraltete IBM-PC auch mit seinen Nachbauten den Markt uniformieren würde. Man kann heute nicht mehr an der Tatsache vorbeisehen, daß solche Geräte bereits sehr verbreitet sind und daß für sie 'Futter' gesucht wird, und sei es in der Form von Datenaustausch und Cross-Software.

Der Computermarkt ist anders zu sehen, als z.B. derjenige für Benzin an den Tankstellen. Dort erhält man eine ziemlich uniforme Ware, und der Wettbewerb findet mit kleinen Preisunterschieden und sonstigem Beiwerk statt. Eine Parallele ist eher zum Pkw-Markt zu ziehen. Hier gibt es die von allen großen Anbietern produzierte 'Golf-Klasse'. Neue Ideen werden dagegen vor allem in den gehobenen Sonderklassen verwirklicht. - In der Computerbranche, deren Technik nach wie vor schnell innovativ ist und in der die gegebenen Möglichkeiten der tatsächlichen Realisierung in Hard- und Software sehr vorauseilen, wäre es tödlich, wenn wir dort nur ein Einerlei hätten. Die besseren Ideen müssen die Chance haben, ältere Konzepte abzulösen. - Die Hersteller erfüllen allerdings auch mit den neuen Computern nie alle Wünsche des Publikums. Insbesondere ist zu erwähnen, daß die Erweiterung des Speichers oder der Interfaces zumeist erschwert ist. Geräte sollen sicher einen attraktiven Preis haben bei einer Grundausstattung, die breiten Wünschen der Anwender und den Erfordernissen der Betriebssysteme und Sprachen entgegenkommen. Man sollte aber eine Computerplatine nicht am Rand 'mit Brettern zunageln', die eine Erweiterung erschweren. Beim hohen Integrationsgrad der heutigen Bausteine läßt sich auf der Hauptplatine häufig noch Platz freihalten. Hier kann man ohne nennenswerte Mehrausgaben Reserve-Steckplätze vorsehen. Interfaces müssen ja von dort nicht herausgeführt worden sein. Wichtig ist doch wohl nur, daß man von dort mit Pinsteckern und Flachkabel bequem herausgelangen kann. Zehn oder so freie Interface-Pins für den User-Port sind heute einfach zu wenig.

Als lästig wirkt sich auch die Einhaltung der FTZ-Bestimmungen bezüglich der Störstrahlung aus: Die Computerplatinen werden in Blech gekapselt und die beiden Halbschalten werden an den Rändern billig und primitiv zugeknebelt. Nach einigen Verdrehungen dieser Knebel sind sie abgebrochen. Warum nimmt man service-freundlicher keine Schrauben, z.B. auch selbsteindrehende?

Im laufenden Heft finden wir neben vielen aktuellen Besprechungen einige größere Beiträge. Insbesondere sind für das Arbeiten mit der Sprache PL/65 erhebliche Verbesserungen geschaffen worden. Der Kreis ihrer Anwender ist sicher klein geblieben. Vielleicht ändert sich das etwas mit diesem Beitrag. - Auch zur Sprache FORTH ist anzunehmen, daß sie trotz der gegebenen Leistungsfähigkeit wohl nur wenig Eingang in den professionellen Bereich gefunden hat. Man soll den Gebrauch einer Sprache nicht zur Philosophie machen. FORTH hat sicher Schwierigkeiten bei der Dokumentation, denn der Leser eines Programmes kann nicht auf Anhieb erkennen, welchen Operanden der Programmierer im Augenblick in Bearbeitung hat und welche anderen Operanden in welcher Ordnung auch noch auf dem Stack liegen. Dokumentation müßte hier eigentlich heißen, daß man ggfs. durch ein Hilfsprogramm jederzeit ein Bild davon bekommen kann, welche Parameter wo verblieben sind.

Klare sprachliche Ausdrucksmöglichkeiten unterstützen von sich aus die Dokumentation.Die Entwicklung eines eigenen Assemblers von Grund auf, hier in den Sommermonaten vollzogen, seine Auslistungen sind schon an verschiedenen Stellen in diesem Heft enthalten, zeigte, daß man mit Bezeichnern von 9 Zeichen Länge, die auch Unterstreichungs-Striche und Fragezeichen enthalten dürfen, auch in Assembler hervorragende Ausdrucksmöglichkeiten gewinnen kann, die die Transparenz und die Dokumentation sehr unterstützen. - Wichtig für den Umgang mit Sprache ist immer auch ein leistungsfähiges Textsystem, mit dem man möglichst in Bildschirmen (Sreens) arbeiten kann. Das für den CBM 720 in diesem Heft abgedruckte Textprogramm hat nicht so weitgesteckte Ziele. Es soll vor allem immer im EPROM auf der Maschine resident sein und bei den Programmierversuchen ohne große Umladungsmaßnahmen schnell zur Verfügung stehen, ebenso wie ein dazu gehörender Assembler im EPROM. In der Summe wird die Produktivität damit sehr befördert. - Das Textsystem wird man logisch auch auf andere Computer übertragen können. Und da es bei CBM eine sehr einheitliche Sprungleiste zu Leistungen des Betriebssystems gibt, wird man es z.B. auch auf den PC-128 übertragen und zum Ablauf bringen können.

Im Rückblick auf den CBM 720 ist zu bemerken, daß die Programmierer nicht so gerne mit Bankumschaltungen gearbeitet haben. Es ist bei einiger Aufmerksamkeit nicht so gefährlich. Und man sollte eine solche Bankumschaltung und neue Konfigurationen auf dem PC-128 üben. Für ihn muß sich noch ein Programmierstil herausbilden, der einerseits die Ressourcen der Maschine geschickt ausnutzt und andererseits hohen Ansprüchen an Ordnung entspricht.

Bei einer Entwicklung im September/Oktober gab es große Überrraschungen mit Eprommern und Eproms, die schließlich zum Aufbau einer kleinen Platine nebst Betriebsprogramm führten. Was in diesem Heft dazu abgedruckt ist, möge ein Anfang sein. Das Thema der EPROMs ist damit sicher nicht vom Tisch.

So sind die Leser denn auch weiterhin gebeten, über ihre Beobachtungen und Entwicklungen in Form von Artikeln zu berichten. Dabei ist klar, daß zu den älteren Computermodellen im Laufe der Zeit weniger zu berichten ist. Aber mit jedem Übergang zu einem aktuellen Computer treten Beobachtungen und Erfahrungen ein, über die berichtet werden sollte, auch über das, was nicht im Handbuch steht oder was dort unzutreffend berichtet wird.

Das nächste Heft soll möglichst noch im Dezember erscheinen. Wahrscheinlich muß der Leser sich jedoch bis Januar etwas gedulden. Es sind hier erhebliche Zeitverluste eingetreten, die so schnell nicht ausgeglichen werden können, wenn man allein vor vielen Anforderungen steht.

Kleinanzeige O=O=O=

Aus der Branche

GTE CMOS-Mikroprozessoren G65SC802 und GS65SC816 mit 8/16 Bit: Nach einer Mitteilung der deutschen Niederlassung in München werden Muster mit 2 und 4 MHz Takt im Januar 1986 erhältlich sein, Produktionsstückzahlen dann im Februar 1986. Über den Preis kann man z.Zt. noch keine Angaben machen.

Commodore: Die Lieferbarkeit der seriellen Floppy 1571 für den PC-128 wird sich wegen technischer Schwiergkeiten noch verzögern. Es wird empfohlen, zwischenzeitlich auf die einseitige Floppy 1570 zurückzugreifen. - Bei älteren Computermodellen und Peripherieeinheiten, die nicht mehr der neuen Linie C-64, PC-128 sowie PC 10 und PC 20 entsprechen, hat es seit Wochen Ausverkaufsaktionen gegeben. Die Bestände scheinen insbesondere bei den Floppys LP8250 und SFD 1001 nunmehr ziemlich geräumt zu sein, so daß man sich ggfs. noch absichern sollte.

Computerboard mct68020-SBC, "Hermann 20". Diese Neuentwicklung enthält die CPU 68020 mit 16 MHz Takt. Ein Floatingpoint Coprozessor FPCP 68881 ist nachsteckbar, ebenso ab Anfang 1986 die Paged Memory Management Unit PMMU 68851. Auf dem Board haben 128 bzw. 512 KB statisches RAM Platz, für dessen Ansprache keine Wartezyklen entstehen, im Unterschied zu dynamischen RAMS. Zur weiteren Ausrüstung gehören VMEbus-Interface mit 32 Bit für Daten und Adressen (der Computer läuft auch ohne dieses Interface), zwei V24-Schnittstellen, programmierbar bis 125 kBaud, Disk-Controller für 5,25 bzw. 3,5" Laufwerke, Interfacebaustein PI/T 68230 mit ca. 30 I/O-Pins, 3 Timern und eine Echtzeituhr. Es können bis zu 64 KB EPROM gesteckt werden, zum Booten, für Festprogramme oder für das bereits erhältliche Monitorprogramm. Ab etwa Jahresende soll dann auch ein Assembler verfügbar sein, der auch die Coprozessoren unterstützt. Als Entwicklungshilfsmittel kann ein IBM-PC oder ein Computer mit CP/M-68K benutzt werden, auf dem ein C-Compiler zur Verfügung steht. CP/M-68K soll danach auf dem Computerboard ablaufen können. - Die Karte eignet mit ihren Merkmalen für sehr schnelle Robotersteuerungen, CAD/CAM (Bitfeldbefehle), Softwareentwicklung oder als Low Cost Single Board Computer mit in dieser Klasse bisher unbekannter Leistungsfähigkeit. - Ebenso von der Fa. MCT in Berlin wurde **ein Europakartencomputer mit der CMOS-CPU G65SC816** entwickelt, der ab Ende Januar 1986 als Bausatz oder komplett lieferbar sein wird, und zwar wahlweise mit Monitor-Programm/Disassembler/Editor oder auch mit Assembler für 65816. Betrieb ist Stand Alone, oder im Anschluß an an verbreitetes Bussystem möglich. Er kann mit 4 MHz Takt als schnelle Steuerung benutzt werden oder auch als Entwicklungssystem. Info: MCT Microcomputertechnik Paul & Scherer, Rostocker Str. 31, 1000 Berlin 21, T. 030 - 392 30 11.

Rechtsstreit: Die Amstrad Comsumer Electronics in England will dem deutschen Verlag Sigrid Huslik in Augsburg den weiteren Verkauf des Buches CPC464 inside out (für den Schneider-Computer) verbieten lassen. Das Buch enthält ein ROM-Listing. Die klagende Partei sieht in der Veröffentlichung eine Urheberrechtsverletzung. Es wurden bereits Verfügungen erlassen, die der Verlag für die Fachpresse dokumentiert hat. - Nach meiner Meinung verhält sich ein Hersteller ungeschickt, wenn er die Transparenz der von ihm gelieferten Systeme erschwert und wenn nicht die Rede davon ist, daß eine Veröffentlichung irgendwie eine Entstellung bedeuten würde.

Drucker: Das Angebot ist inzwischen schier unübersehbar. Nach dem Messebesuch in München kurz folgende Hinweise: Die Fa. Brother bietet jetzt Drucker an, die sowohl einen Matrixkopf für schnelle Listungen und Grafik haben, als auch ein Typenrad für Ausgabe in Schönschrift. Beide Druckarten können durcheinander gebraucht werden (Twinwriter 5, DM 4.198,- + MWSt). - Bei Siemens konnte der Tintenstrahldrucker PT88S im Einsatz besichtigt werden (DM 2.600), der geringes Geräusch, klare Schrift, ladbare Zeichensätze, Grafikauflösung bis 144 Dots per Inch und eine hohe Druckgeschwindigkeit bis zu 340 Zeichen/Sek. vereinigt.

Assembler

Neu konzipierter Assembler für 6502, R65C02, 65802/65816 lauffähig auf AIM 65 und kompatiblen, ferner auf CBM 720

Deutsche interaktive Bedienerführung, Fehlerhinweise in deutschem Klartext. Einstellbare Befehlssätze für die CPU-Typen mit Fehlerhinweis bei nicht implementierten Befehlen/Adressierungen. Schnelldurchgang mit alten Parametern möglich.. Pass 2 kann mit neuen Ausgabeparametern zeitsparend wiederholt werden. Symboltafel alphabetisch geordnet. Symbole dürfen 9 Zeichen lang sein (Unterstreichungszeichen, Fragezeichen erlaubt), dadurch aussagefähige Label. Listbilder in diesem Heft. Übliche Eingabemedien des AIM mit bes. Floppy-Kanal (CBM 720: Floppy + Editor). E/A läuft über Sprungleiste am Programmbeginn. Im Texteditor können Kommandofiles benutzt werden, die Quelltexte von verschiedenen Eingabemedien verbinden. Codeablage und Offset-Ablage ein- und ausschaltbar. Erweiterte logische Operandenverknüpfung. - Eine vorhandene Symboltafel kann zusätzlichen Quelltexten zur Verfügung gestellt werden (nützlich bei engem Speicher). Speziell auf CBM 720 großzügiges Arbeiten: Quelltext im RAM resident bis 128 KB, 1 Bank nur für die Symboltafel (bis ca. 4600 Symbole), 1 Bank für Code-Ablage. Code kann aber auch an beliebige Stellen/Bänke abgelegt werden. Profi-Werkzeug mit Dokumentation. 8 KB-Programm für AIM in 2 Eproms, für CBM 1 Eprom für Bank 15-Erweiterung oder auf Diskette LP 8250. DM 350,-

In Vorbereitung: Assembler für PC 128 (bitte anfragen).

Cross-Assembler für 6800/6802/6803/6303

für die CPU-Typen von Motorola und Hitachi, für den AIM 65 und kompatible Systeme. Erzeugt aus den Mnemonics von Motorola Maschinencode. Zwei-Pass-Assembler mit gewohnten Komfort, Syntax jedoch nach 6502, 2 EPROMs mit Dokumentation DM 300,-

6805/68705 Cross-Assembler

für AIM 65 und kompatible Systeme. 2-Pass-Assembler mit allem gewohnten Komfort und 6502-Syntax. Erzeugt aus den Mnemonics von Motorola 6805-Code. 2 EPROMs wie vor DM 280,-

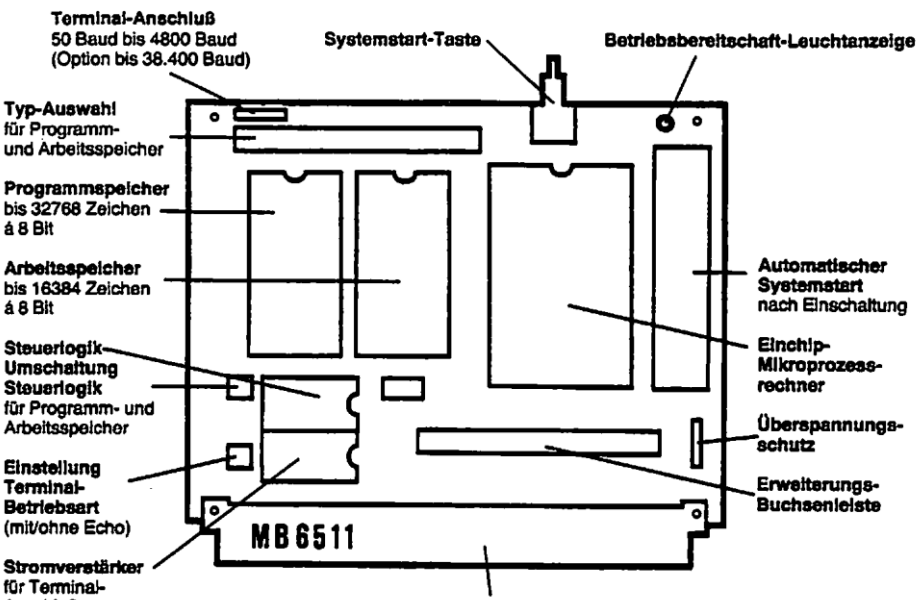
Mathe-ROM für 6502

Implementierung nach Peter Rix (s. Hefte 28/29). Fließkommaarithmetik und höhere mathematische Funktionen wie in Microsoft-BASIC für AIM-65-FORTH und für jedes 6502-Assemblerprogramm (20 S. Dokumentation mit Einsprungspunkten und Argumenten), für Sockel $D000 DM 124,30

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

8408

MINIBEE- MB 6511



TECHNISCHE DATEN

Leiterplattengröße	100mm x 80mm
Gesamtmaße	100mm x 88mm x 14mm
Kartenmaterial	Glasfaser-Epoxy 1,5mm, beidseitig Lötstoplack
Gewicht	80g
Leistungsbedarf	2W
Spannungsversorgung	5V ± 5%
Systemtakt	2MHz (Option 1MHz)
Arbeitstemperaturbereich	0°C bis +70°C
Programmspeicher	2kByte bis 32kByte BYTEWIDE
Arbeitsspeicher	2kByte bis 16kByte BYTEWIDE
extern erweiterbar bis	4MByte (256 x 16kByte)
Terminal-Anschluß	TTL-kompatibel

RECHNER DATEN

Einchip-Mikroprozessor	R6511Q (Rockwell/NCR)
Rechnerstruktur	6500
Programmierung	softwarekompatibel zu 6500 60 Befehle, 14 Adressierungsarten
Ein-/Ausgabe-Anschlüsse	24 O.C. mit Pullup, 8 Tristate, alle TTL kompatibel
Zähler/Zeitmesser/Takt	1 voll programmierbar, retriggerbar 1 für serielle Schnittstelle
Serielle Schnittstelle	voll duplex, asynchron, synchron
Programm-Unterbrechung (IRQ)	10 (je 2 positiv, negativ, Zähler, Serielle Schnittstelle; 1 unbedingte Unterbrechung (NMI)

EINCHIP-MIKROPROZESSOR-STEUERUNG

mit R6511Q

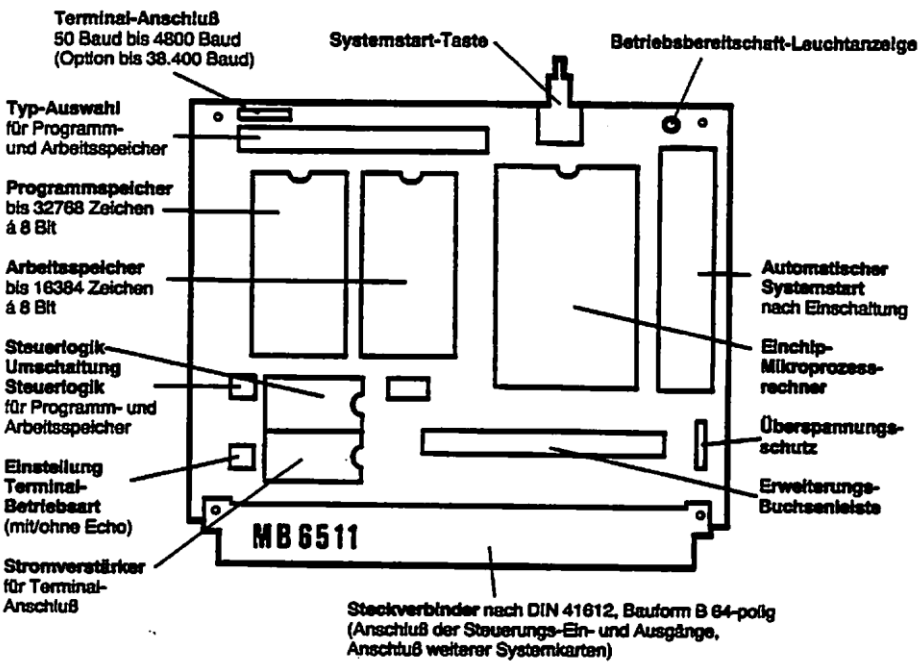
Ab sofort lieferbar:
mit MINIBEE-FORTH ..... DM 395,-
Batterie-Option .....+DM 90,-

BRÜHL ELEKTRONIK ENTWICKLUNGS-GESELLSCHAFT mbH

8500 Nürnberg 10, Hegelstr.10 TEL. 0911-35 90 88

8408

MINIBEE- MB6511



TECHNISCHE DATEN

Leiterplattengröße	100mm x 80mm
Gesamtmaße	100mm x 88mm x 14mm
Kartenmaterial	Glasfaser-Epoxy 1,5mm, beidseitig Lötstoptack
Gewicht	80g
Leistungsbedarf	2W
Spannungsversorgung	5V ± 5%
Systemtakt	2MHz (Option 1MHz)
Arbeitstemperaturbereich	0°C bis +70°C
Programmspeicher	2kByte bis 32kByte BYTEWIDE
Arbeitsspeicher	2kByte bis 16kByte BYTEWIDE
extern erweiterbar bis	4MByte (256 x 16kByte)
Terminal-Anschluß	TTL-kompatibel

RECHNER DATEN

Einchip-Mikroprozessor	R6511Q (Rockwell/NCR)
Rechnerstruktur	6500
Programmierung	softwarekompatibel zu 6500 60 Befehle, 14 Adressierungsarten
Ein-/Ausgabe-Anschlüsse	24 O.C. mit Pullup, 8 Tristate, alle TTL kompatibel
Zähler/Zeitmesser/Takt	1 voll programmierbar, retriggerbar 1 für serielle Schnittstelle
Serielle Schnittstelle	voll duplex, asynchron, synchron
Programm-Unterbrechung (IRQ)	10 (je 2 positiv, negativ, Zähler, Serielle Schnittstelle; 1 unbedingte Unterbrechung (NMI)

EINCHIP-MIKROPROZESSOR-STEUE

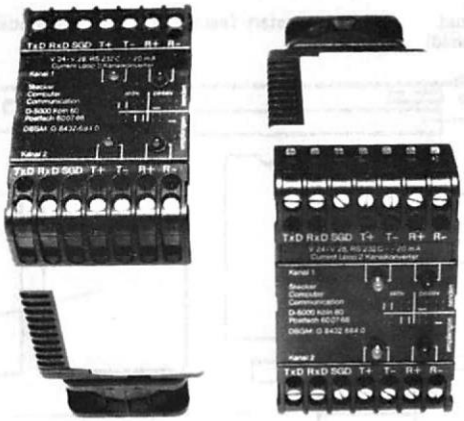
mit R6511Q

DM 350,– + Mwst

SPEZIAL-ANGEBOT für MICRO MAG-Leser:
DM 380,– incl. 8 kByte RAM
und AIM65-Terminal-Monitor
(Mit Assembler + Disassembler)

BRÜHL ELEKTRONIK ENTWICKLUNGS-GESELLSCHAFT mbH

8500 Nürnberg 10, Hegelstr. 10 TEL. 0911-35 90 88



Neu: V.24/V.28 & RS 232 C <——> 20 mA Current Loop 2 Kanalkonverter

in einem Gehäuse mit 11-poligem Stecksockel für Schaltschrankmontage

Der 2 Kanalkonverter, DBGM 8432 684.0, ist eine Weiterentwicklung des bewährten Steckerkonverters, der im Gehäuse des Subminiatur "D" Steckers der V.24-schnittstelle eingebaut wird und dort als 20mA Current Loop Interface fungiert.

Technische Daten:

- 19200 bit/sec bis 2000 Meter
- 2400 bit/sec bei 5700 Meter
- 100% galvanische Trennung
- ein Netzteil für jeden Kanal
- einfache Montage auf Hutschiene
- LED-Kontrolle der Übertragungsstrecke
- 100% kompatibel mit anderen 20mA Systemen
- Betriebsart aktiv oder passiv für jede Stromschleife

Die Datenübertragung mit 20mA Stromschleifen gehört zur digitalen Basisübertragung. Das Stromschleifenprinzip, ein dem Wellenwiderstand angepaßter Leitungsabschluß und ein überduchschnittlich hoher Triggerpegel, logisch 1 => 13 mA und logisch 0 <= 11 mA, garantieren große Reichweiten mit hoher Übertragungsrate und größter Störsicherheit.

Der Einsatz von einem Netzteil je Konverter ermöglicht sowohl die Potentialtrennung zwischen der Übertragungsstrecke und den angeschlossenen Datengeräten, als auch zwischen den Stromschleifen im selben Kabel. Letztere vermindert die Betriebskapazität der Datenleitung und sorgt somit für eine höhere Datenübertragungsrate, die sich mit 19200 bit/sec bei einer Leitungslänge von 2000 Metern dokumentiert.

Getestet werden kann die Datenübertragungsstrecke dank der in den Stromschleifen liegenden Leuchtdioden auch ohne die angeschlossenen Datengeräte, da es sich um ein Ruhestromsystem handelt, d.h. im Ruhezustand wird die logische 1 übertragen. - Bei der Errichtung von Punkt zu Punkt Datennetzen mittels 20 mA Current Loop Systemen hat der Anwender einen beachtlichen Kostenvorteil. Ggfs. können freie Adern eines Telefonkabels benutzt werden. Bereits vorhandene EDV-Geräte werden netzwerkfähig, womit der Ankauf neuer Geräte mit teueren Netzwerkinterfaces entfällt.

INGENIEURBÜRO
STECKER

5000 Köln 60 (Niehl)
Postfach 60 07 66
Delmenhorster Str. 20
Tel. (02 21) 712 40 18

IBM

IBM-PC HANDBUCH

Als **EINFÜHRUNG** beschreibt der Text alle Schritte zwischen Inbetriebnahme, Tastaturerläuterung, Peripherieanschluß, Laden der verschiedenen BASIC-Versionen bis zu ersten Einsätzen für BASIC-Programme, Graphik, Akustik, Datenspeicherung, Ausdruck.
Als **BASIC-LEHRBUCH** wird dieser Text in den USA, der Schweiz und Deutschland für berufsbildende Kurse eingesetzt.
Als **SYSTEMBUCH** enthält es selten behandelte Informationen zu Themen wie Datenfernübertragung und LANs mit IBM-PCs, Erweiterungskarten, Speicherbelegungen, Kombination BASIC/Maschinenprogramme, BASIC-Erweiterungen, Hinweise zu IBM-Software wie SYMPHONY usw.
Ein Buch für Einstieg und Referenz.

Von Lyle. J. Graham, 416 Seiten, Softcover, DM 59,–

IBM-PC-/XT ASSEMBLER-PROGRAMMIERUNG CPU 8088
Ein neues Buch für professionelle und nichtprofessionelle IBM-PC-Anwender. Mit umfangreichen Beispielprogrammen und bildlichen Darstellungen. Sie erfahren z. B., welche Systemarchitektur der IBM-PC/XT hat; wie die INTEL-CPU 8088 im PC/XT arbeitet, wie er in Assemblersprache programmiert wird, wie man mit DOS, BIOS und MACROASSEMBLER zu arbeiten hat. Die Peripherie kommt natürlich auch nicht zu kurz.

Von Willen/Krantz, 416 Seiten, Softcover, DM 66,–

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

te-wi

MICRO MAG

HERAUSGEBER/EDITOR:
DIPL.-VOLKSWIRT ROLAND LÖHR
HANSDORFER STRASSE 4
D-2070 AHRENSBURG
☎ (04102) 5 58 16

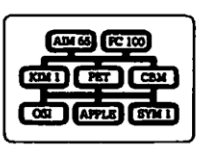
MICRO MAG (vormals 65xx MICRO MAG) erscheint zweimonatlich, jeweils Mitte Februar, April usw.. COPYRIGHT 1985 by Roland Löhr. Alle Rechte vorbehalten, auch die des auszugsweisen Nachdruckes, der Übersetzung, der fotomechnischen Wiedergabe und die der Verbreitung auf magnetischen und sontigen Trägern. Von den veröffentlichten Programmen, Schaltungen und Angaben wird ohne eine Gewährleistung von hier aus angenommen, daß sie fehlerfrei und frei von den Schutzrechten Dritter sind. Beiträge, die nicht besonders gekennzeichnet sind, stammen vom Herausgeber. Offsetdruck: L & L Druckservice, Hamburg 73

Bezugsbedingungen: Abonnement ab laufender Ausgabe für 6 Hefte DM 54,-- im Inland, bzw. DM 59,-- im Ausland (surface mail). Luftpostzustellung auf Anfrage. Abonnements laufen bis auf Widerruf mit Kündigungsmöglichkeit bis zu 4 Wochen vor deren Ablauf. - Nachliefermöglichkeiten siehe unten.

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

Leser-Service des Herausgebers

Das Buch 7-13 des 65.. MICRO MAG



340 Seiten, DM 42,-

Nachliefermöglichkeiten:

Vergriffen sind 'Das Buch 1-6 des 65xx MICRO MAG' sowie die Hefte 1-13 und 16 der Zeitschrift. Es erfolgt keine Neuauflage.

Lieferbar: 'Das Buch 7-13 des 65xx MICRO MAG' zu DM 42,-- sowie die Hefte 14, 15 und 17-44 zu DM 7,80/St. (ab 10 St. in einer Sendung: DM 6,-/St.).

Mathe-ROM nach P. Rix für FORTH und Einbindung in Assembler-Programme, mit Dokumentation DM 124,30.

Assembler für 6502/65C02/65802/65816 (drei Befehlssätze!) sowie für MC6805 und für 6800/6802/6303 (drei Befehlssätze): Siehe im Heftinneren.

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