

65_{xx} MICRO MAG

COMPUTING · SOFTWARE · HOBBY

DM 7,80

Nr. 11

F E B R U A R 1 9 8 0

VIELE NÜTZLICHE DIENSTPROGRAMME

und Erfahrungsberichte wurden aus der Leserschaft zu diesem Heft beigesteuert oder schon für künftige Hefte angekündigt. Mit Freude ist zu betrachten, wie sich das Arsenal der Hilfen für die Programmentwicklung abrundet und wie die in den Prozessoren enthaltenen Festwertprogramme (Betriebssystem, Assembler, BASIC) durchleuchtet und erweitert nutzbar gemacht werden. Gleichzeitig stellt die Hardwareseite ein immer leistungsfähigeres Angebot preiswert zur Verfügung. - Die Mikros werden damit mehr und mehr 'erwachsen'

I N H A L T S V E R Z E I C H N I S

MLIST (AIM)	3
Disk Utility Program (CBM)	6
6502 Direct Assembler (PET)	12
Schaufel (Relocate, PET)	16
VARLIST - Variablenausdruck (PET)	17
AIM Spezial (6)	22
Ein Printer für den AIM	23
CBM schießt sich eigene EPROMs	24
ROM-Vergleichsliste CBM-PET	28
Anschluß von numerischer Anzeige und Tastatur	31
USCOM - User Defined BASIC Commands (AIM)	37
Erweiterung des AIM auf den S 100-Bus	39
Das VIDEO PLUS	41
Superboard Challenger II	44
English Summary	45

CPU-Karte NICO 65

CPU: 6502, 1 oder 2 MHz, 2 kB RAM, max. 8 kB EPROM beliebigen Typs (24-polig), 2 VIAs = 40 Pin I/O über Pfostenleiste nach vorne ausgeführt, alle Busse gepuffert, VG 64 Steckerleiste, Power-On-Reset, alle Interruptmöglichkeiten, Europakartenformat.

Preis: DM 450,-
in Grundausstattung = 1 kB RAM, 1 VIA DM 390,-

CPU-Karte NICO 69

CPU: 6809, Beschreibung wie oben. Die CPU hat eine interne 16-Bit-Struktur und arbeitet nach außen hin auf 8 Bit Datenbusbreite. Die Busse sind zum 6502 hin kompatibel.

Preis: DM 510,-
in Grundausstattung = 1 kB RAM, 1 VIA DM 450,-

Combo-Karte

8 kB RAM statisch, 2 VIAs, 40 Pin I/O über Pfostenleisten nach vorne ausgeführt, RAM und VIAs getrennt selektierbar, busgepuffert.

Preis: DM 560,-
nur mit RAMs bestückt DM 470,-
nur mit VIAs bestückt DM 140,-

Video-Karte

Bildschirmformat beliebig (max. 80 Zeichen x 24 Zeilen), bis zu 16 Rasterzeilen pro Zeichenzeile, daher echte Unterlängen, Potenzschreibweise und Indices möglich. Zeichengenerator für 128 Zeichen frei programmierbar, ausbaubar auf 256 Zeichen on board, Betriebsprogramm im EPROM on board, 2 k Bildwiederholpeicher, Grafik bzw. Semigrafik möglich. Durch Erweiterung halbe Helligkeit, Blinken, 4 k Bildwiederholpeicher, Tongenerator, Strichgrafik für Formulare zwischen den Zeichen bzw. Zeilen.

Preis: DM 660,-

Buskarte

8 Steckplätze für alle Normen mit BG 64 Leisten DM 130,-

Matrixdrucker

Friktionsantrieb für normales Rollenpapier, 80 Zeichen pro Zeile, Zeilenabstand 4,23 mm, 100 Zeichen pro Sek., Zeichenvorrat beliebig; normal: 64 Standard ASCII-Zeichen, Betriebssoftware incl.

Preis: DM 745,-

Interfacekarte dafür

Treiber, Schutzschaltung für Druckkopf, Motorsteuerung und Stromversorgung, ohne Trafo (2x24 V, 1 A AC).

Preis: DM 73,-

Alle Preise zuzüglich 13 % MWSt.

65xx MICRO MAG

Ing. (grad.) Horst Steder, Frankfurt

MLIST (AIM)

Bei der Programmentwicklung mit einem AIM 65 und angeschlossenem TV-Display entstand der Wunsch, die Memory-List-Routine des AIM so zu erweitern, daß nicht nur der Inhalt von jeweils 4, sondern von jeweils 16 Speicherzellen zur Anzeige gebracht werden kann. Zusätzlich sollte es möglich sein, vorwärts und rückwärts durch den Speicher zu gehen und natürlich auch Änderungen in einer Zeile vornehmen zu können. Zur Bedienung des Programmes MLIST, das diese Funktionen bereitstellt, ist hinsichtlich der Tastenbelegung folgendes zu bemerken:

"SPACE" oder "D" Diese Tasten erlauben ein zeilenweises Vorwärtsschreiten, erlaubt Eingabe einer neuen Ausgangsadresse,
 "M" erlaubt zeilenweises Rückwärtsschreiten,
 "U" ermöglicht innerhalb der Zeile Änderung an beliebigen Plätzen,
 "/" führt zum Aussprung in den Monitor.
 "ESCAPE"

Wer sich übrigens nicht an einem "=" zu Beginn einer Zeile stört, kann durch nachstehende Befehlsfolge 14 Bytes sparen: Die Befehle in den Speicherplätzen 0200 bis einschl. 0213 ("PLA") werden ersetzt durch

MLIST JSR CRLF (in E9F0)
 JSR PATCH8 (in FEBC). Der Folgebefehl ist dann wie vorher CMP #'M.

```

0000 ; *****
0000 ; *
0000 ; *   PROGRAM   < T V - M L I S T >   *
0000 ; *
0000 ; *           H.STEDER 10/79 *
0000 ; *****
0000 ; THIS ROUTINE SHOWS 16 MEMORY LOCATIONS IN ONE LINE
0000 ; <SPACE>- OR <D>- KEY SHOWS FOLLOWING 16 LOCATIONS
0000 ; <U>- KEY SHOWS PREVIOUS 16 LOCATIONS
0000 ; </>- KEY ALLOWS MEMORY ALTER ANYWHERE IN THE LINE
0000 ; <ESC>- KEY RETURNS CONTROL TO MONITOR
0000
0000 ADDR      =#A41C      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXX
0000 OUTPUT    =#EF02 <---- ;X DILINK VECTOR MUST BE SET X
0000 NXTADD    =#E2CD      ;X TO USER OUTPUT ROUTINE ! X
0000 WRITAZ    =#E2DB      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXX
0000 OM        =#E7D4
0000 BLANK     =#E83E
0000 NUMA      =#EA46
0000 RD2       =#EA5D
0000 ADDIN     =#EAAE
0000 MEMERR    =#EB33
0000 LDAY      =#EB58
0000 SADDR     =#EB78
0000 RED1      =#FE96
0000
0000          =#0112      ;FUNCTION KEY <F3>
0112 4C0002 JMP MLIST
0115

```

65xx MICRO MAG

```

0115                      **=$0200
0200
0200 MLIST      A90D      LDA  #0D
0202            2002EF    JSR  OUTPUT
0205            A93C      LDA  #<<
0207            2002EF    JSR  OUTPUT
020A            2096FE    JSR  RED1      ;GET COMMAND WITHIN CARROTS
020D            48        PHA
020E            A93E      LDA  #>
0210            2002EF    JSR  OUTPUT
0213            68        PLA
0214            C94D      CMP  #'M'      ;<M>-KEY FOR START ADDRESS?
0216            F016      BEQ  ML1        ;YES, GET START ADDRESS
0218            C920      CMP  #' '      ;<SPACE>-KEY?
021A            F018      BEQ  ML2        ;YES, SHOW NEXT 16 MEMORY LOCATIONS
021C            C944      CMP  #'D'      ;<D>-KEY FOR DOWN?
021E            F014      BEQ  ML2        ;YES, SHOW NEXT 16 LOC ALSO
0220            C955      CMP  #'U'      ;<U>-KEY FOR GOING BACK?
0222            F06E      BEQ  LSTADD      ;YES, SHOW PREVIOUS 16 MEM LOC
0224            C92F      CMP  #'/'      ;CHANGE DESIRED?
0226            F03C      BEQ  MRALTR      ;YES, GO TO MEMORY ALTER ROUTINE
0228            20D4E7    JSR  QM          ;ON ANY OTHER KEY SHOW '?'
022B            4C0002    JMP  MLIST      ;AND GO BACK FOR NEW INPUT
022E
022E ML1        204B02    JSR  MEM
0231            4C0002    JMP  MLIST
0234 ML2        203A02    JSR  NXT16
0237            4C0002    JMP  MLIST
023A NXT16      203EE8    JSR  BLANK      ;SAME ROUTINE AS USED
023D            A010      LDY  #16        ;IN AIM-65 MONITOR, BUT 16 MEM LOCS
023F            20CDE2    JSR  NXTADD      ;INSTEAD OF 4 ARE DISPLAYED
0242            20DBE2    JSR  WRITAZ

0245            203EE8    JSR  BLANK
0248            4C5002    JMP  MEIN
024B MEM        20AEEA    JSR  ADDIN
024E            B013      BCS  MEM3
0250 MEIN       A210      LDX  #16
0252            A000      LDY  #0
0254 MEM2       203EE8    JSR  BLANK
0257            A91C      LDA  #1C
0259            2058EB    JSR  LDAY
025C            2046EA    JSR  NUMA
025F            C8        INY
0260            CA        DEX
0261            D0F1      BNE  MEM2
0263 MEM3       60        RTS
0264
0264            ; MEMORY ALTER ROUTINE
0264 MRALTR      203EE8    JSR  BLANK
0267            20DBE2    JSR  WRITAZ      ;WRITE ADDRESS
026A            203EE8    JSR  BLANK
026D ALTR1       203EE8    JSR  BLANK
0270            205DEA    JSR  RD2        ;GET NEW VALUE
0273            900A      BCC  ALTR2      ;IS NOT SKIP OR DONE
0275            C920      CMP  #20        ;<SPACE>?
0277            D013      BNE  ALTR3      ;NOT <SPACE>, SO MUST BE DONE
0279
0279            ; SKIP THIS LOCATION
0279            203EE8    JSR  BLANK
027C            4C8702    JMP  ALTR16
027F

```


65xx MICRO MAG

```

027F      ; IS ALTER MEMORY
027F ALTR2 2078EB JSR SADDR      ;STORE ENTERED VALUE INTO MEMORY
0282      F003 BEQ ALTR16      ;NO ERROR IN STORE
0284      4C33EB JMP MEMERR     ;MEMORY WRITE ERROR
0287 ALTR16 C8      INY
0288      C010 CPY #16
028A      D0E1 BNE ALTR1      ;GET NEXT VALUE
028C
028C      ; LINE DONE OR <CR>
028C ALTR3 20CDE2 JSR NXTADD     ;UPDATE THE ADDRESS
028F      4C0002 JMP MLIST      ;BACK TO ROUTINE START
0292
0292      ; GET PREVIOUS 16 MEMORY LOCATIONS
0292 LSTADD 38      SEC          ;SUBTRACT 32 FROM ADDRESS...
0293      AD1CA4 LDA ADDR        ;<BECAUSE 16 WILL BE ADDED AGAIN>
0296      E920 SBC #32          ;TO GET PREVIOUS LINE
0298      8D1CA4 STA ADDR
029B      B097 BCS ML2
029D      CE1DA4 DEC ADDR+1
02A0      9092 BCC ML2
02A2
02A2
02A2
02A2      .END
02A2      ERRORS= 0000

```

** SYMBOL TABLE **

ADDIN	EAAE	ADDR	A41C	ALTR1	026D	ALTR16	0287
ALTR2	027F	ALTR3	028C	BLANK	E83E	LDAY	EB58
LSTADD	0292	MEIN	0250	MEM	024B	MEM2	0254
MEM3	0263	MEMERR	EB33	ML1	022E	ML2	0234
MLIST	0200	MRALTR	0264	NUMA	EA46	NXT16	023A
NXTADD	E2CD	OUTPUT	EF02	QM	E7D4	RD2	EA5D
RED1	FE96	SADDR	EB78	WRITAZ	E2DB		

ANWENDUNGSBEISPIELE FÜR MLIST:

```

<M>=0400 A6 A6 AA B0 23 1C A5 04 A6 A3 A0 03 0C B0 04 B0 "M" für Start-Adi
< > 0410 55 79 64 00 51 60 C8 40 02 B0 D6 70 A6 73 F5 F0 "SPC" ↓
< > 0420 98 A5 DC 45 98 B4 55 FC 4A 10 A3 A5 D8 69 7F 45
<D> 0430 98 A5 5C 89 7F 25 DC F8 78 A2 03 E5 E8 D5 DC 95 " D " ↓
<D> 0440 86 CA 00 97 98 90 8E C8 A5 A2 89 01 85 82 28 98
<D> 0450 CB A0 01 B1 A0 99 9A 00 88 C0 02 B0 C6 20 82 00
< > 0460 84 AA A0 BC A9 00 95 80 C8 D0 F9 00 91 04 60 BA
< > 0470 84 20 C5 03 A0 88 83 80 00 A9 A0 C9 FF 90 EB 84
<U> 0460 84 AA A0 BC A9 00 95 80 C8 D0 F9 00 91 04 60 BA " U " ↑
<U> 0450 CB A0 01 B1 A0 99 9A 00 88 C0 02 B0 C6 20 82 00
<U> 0440 86 CA 00 97 98 90 8E C8 A5 A2 89 01 85 82 28 98
<U> 0430 98 A5 5C 89 7F 25 DC F8 78 A2 03 E5 E8 D5 DC 95
< > 0440 86 CA 00 97 98 90 8E C8 A5 A2 89 01 85 82 28 98
</> 0440 11 22 AC EF DA 55 EB ←Change Memory
<M>=0440 11 22 00 97 AC 90 EF DA A5 A2 89 55 EB 82 28 98 ←Neuer Inhalt
< > 0450 CB A0 01 B1 A0 99 9A 00 88 C0 02 B0 C6 20 82 00
<M>=D000 20 EC DF A2 32 20 E1 DF 20 EC DF A9 00 85 23 85 Assembler-ROM
< > D010 0B 85 C0 20 A3 E7 B0 FB AD 1E A4 D0 0D AD 1C A4
< > D020 85 3A AD 1D A4 85 3B 4C 31 D0 A5 3B A6 3A 20 42
< > D030 EA 20 FA DF 20 A7 E7 B0 FB AD 1E A4 D0 0D AD 1C
</> D030 55

```

MEM FAIL D032

Abbruch bei Versuch, ins ROM zu schreiben

Dipl.-Ing. Uwe Kornnagel, Raunheim

DISK UTILITY PROGRAM

Dieses Disk Utility Program bietet dem Anwender die Möglichkeit, einfacher mit der cbm 3040-Diskette zu arbeiten. Es ist der zweite Teil zum Commodore DUM und erlaubt es, eine Diskette neu zu formatieren, zu löschen und mit gleicher ID umzubenennen. Testlesen einer Diskette oder nur bestimmter Spuren mit Ausgabe einer Fehlerliste auf dem Bildschirm ist möglich.

Weitere Leistungen sind: Gezieltes Kopieren von einer Diskette auf eine andere und das Zusammenbauen von zwei Programmen auf der Diskette unter neuem Namen (Append-Statement). Das Statement 2360 DATA nimmt die Programme oder Dateien auf, die nach einem Neuformatieren oder Löschen auf jeder Diskette stehen sollen. Sie sind vom Anwender frei wählbar.

```

1000 REM CBM 3001 DUP
1005 IFPEEK(826)<>1690RPEEK(827)>>0THENPRINT"***INIT DUP***":RUN 3690
1010 PRINT"CBM 3001 DISK UTILITY PROGRAM 3001":PRINT
1020 PRINT"(C) 1979 DIPL. ING. U.KORNNAGEL"
1030 PRINT:PRINT
1040 PRINT:PRINT"FORMATIERE DISK"
1050 PRINT:PRINT"KOPIERE FILES VON DISK X NACH DISK Y"
1060 PRINT:PRINT"TESTLESE EINE DISKETTE"
1070 PRINT:PRINT"ZUSAMMENBAU VON 2 PROGRAMMEN"
1080 PRINT:PRINT"PROGRAMM ENDE."
1090 PRINT:PRINT:PRINT"BITTE 0, 1, 2, 3 ODER 4 EINGEBEN ";
1100 SYS990:GETQ$
1110 IFQ$="F"THENRUN2070
1120 IFQ$="C"THENRUN1180
1130 IFQ$="T"THENRUN2400
1140 IFQ$="P"THENRUN3320
1150 IFQ$="*"THENEND
1160 GOTO1100
1170 END
1180 REM CBM 3040 DISK FILE COPY
1190 REM MODIFY 1.0 BY UWE KORNNAGEL
1200 PRINT"CBM 3040 DISK - FILE - COPY"
1210 PRINT"MODIFIZIERTE VERSION 1979 BY"
1220 PRINT"DIPL. ING. UWE KORNNAGEL"
1230 PRINT" "
1240 DIML$(100),N$(100),T$(100),ER$(100),CP$(100):PRX=826:GOTO1510
1250 PRX=871
1260 POKE216,LNX:POKE198,SPX:SYSPRX:PRX=826:RETURN
1270 OPEN15,8,15:N0%=0:PRINT" CATALOG EINLESEN."
1280 B=30:PRINT#15,"M-E"CHR$(212)CHR$(237)
1290 OPEN 2,8,0,"$"+D$+" "+I$
1300 GET#2,A$:GET#2,A$:I$=""
1310 IF ST<0 GOTO 1410
1320 FOR A=1TOB:GET#2,A$
1330 IF A$="" THEN A$=CHR$(0)
1340 I$=I$+A$:NEXT I$:I$=I$+" "
1350 B=ASC(MID$(I$,3,1))
1360 B=B+ASC(MID$(I$,4,1))*256
1370 LZ(N0%)=B:N$(N0%)=MID$(I$,7,29):T$(N0%)=MID$(I$,26):TY$=T$(N0%):N0%=N0%+1
1380 TY$=MID$(TY$,2,1)

```

65xx MICRO MAG

```

390 IFTV$<"P"ANDTV$<"R"ANDTV$<"U"ANDTV$<"S"THENN0%=N0%-1
400 I$="" :B=32 :GOTO 1310
410 CLOSE2:CLOSE15
420 POKE216,18:POKE198,2:SYS871:PRINT"AUFBAU DES CATALOGS "
430 FORI0=0TON0%-1:FF$=""
440 POKE216,18:POKE198,30:SYS826:PRINTI0+1
450 FORJ0=1TOLEN(N$(I0))
460 IFMID$(N$(I0),J0,1)<>CHR$(34)THENNEXTJ0
470 FORJ1=J0+1TOLEN(N$(I0))
480 IFMID$(N$(I0),J1,1)=CHR$(34)GOTO1500
490 FF$=FF$+MID$(N$(I0),J1,1):NEXT J1
500 N$(I0)=FF$:NEXTI0:RETURN
510 PRINT"COPIEREN VON DRIVE "
520 SYS990:GET0$:IFASC(Q$)<48ORASC(Q$)>49GOTO1520
530 VO%=VAL(Q$):VO$=Q$:BI$=NOTVO%AND1:BI$=RIGHT$(STR$(BI%),1):PRINTVO$:" NA
BI$
540 LN%=10:SP%=0:GOSUB1260:PRINT"ANFANGSKETTE (J/N) "
550 SYS990:GET0$:IFQ$="N"THENI$="*":PRINT"NEIN":GOTO1580
560 IFQ$<>"J"GOTO1550
570 OPEN1,0,0:LN%=10:GOSUB1250:PRINT"ANFANGSKETTE " :INPUT#1,I$:CLOSE1:I$=I$
"
580 LN%=12:GOSUB1260:PRINT"TYPE SP$FROM 33EQ, 33SR ODER 33ELLES "
590 SYS990:GET0$:PRINT" Q":Q$=I$+Q$:I$=VO$
600 LN%=16:GOSUB1260:GOSUB1270
610 FORI=24TO11STEP-1:LN%=I:GOSUB1250:NEXT
620 LN%=12:SP%=0:GOSUB1250:PRINT"LAENGE":SP%=10:GOSUB1260:PRINT"NAME":SP%=2
UB1260
630 PRINT"TYPE"
640 FORI=0TON0%-1:LN%=14:SP%=0:GOSUB1250
650 PRINTLX(I):SP%=10:GOSUB1260:PRINTN$(I):SP%=26:GOSUB1260:PRINTT$(I)
660 LN%=16:SP%=3:GOSUB1250:PRINTCP%+1"COPIEREN (J/N) "
670 SYS990:GET0$:IFQ$="N"GOTO1700
680 IFQ$<>"J"GOTO1670
690 CP$(CP%)=N$(I):CP%=CP%+1
700 LN%=16:GOSUB1250:NEXT
710 PRINT"Q":CP%+1"■ COPIEN ALPHABETISCH (J/N) "
720 SYS990:GET0$:IFQ$="N"THEN1780
730 FORI=11TO16:LN%=I:GOSUB1250:NEXT:LN%=14:GOSUB1260:PRINT"ALPHABETISCH SO
EN ."
740 FORI=0TOCP%-2:FORJ=I+1TOCP%-1
750 IFCP$(I)<=CP$(J)GOTO1770
760 CT$=CP$(I):CP$(I)=CP$(J):CP$(J)=CT$
770 NEXTJ,I
780 PRINT"Q":POKE216,1:POKE198,21:SYS979:POKE216,0:SYS871:POKE216,1:POKE198
790 SYS826:PRINT"+":POKE216,2:POKE198,0:SYS826:PRINT"NAME"
800 :POKE216,2:POKE198,22:SYS826:PRINT"MELDUNG":POKE216,3:SYS875:POKE198,21
810 SYS826:PRINT"+":POKE216,4:POKE198,0:SYS826
820 PRINT"■ CBM 3040 COPY ROUTINE STATED"
830 POKE216,4:POKE198,1:SYS826
840 OPEN15,8,15,"I":ERX=0:ESX=0
850 FORI=0TOCP%-1:CC$="" :PRINTCP$(I):" "
860 POKE198,22
870 CC$="C"+BI$+"":+CP$(I)+"="+VO$+"":+CP$(I)
880 PRINT#15,CC$:INPUT#15,EC,EC$,TRX,SCX
890 IFEC=0THENJPM=0:PRINTEC$:GOTO1930
900 IFEC=63THENJPM=0:ERX=ERX+1:ERX(ERX)=I:PRINTEC$:GOTO 1930
910 IF(EC=62OREC(26)ANDJPM=0)THENPRINT#15,"I":PRINT#15,CC$:INFUT#15,EC,EC$,TR

```



```

1920 PRINT"*EC$*":PRINT
1930 NEXT
1940 IFER%=0THENCLOSE15:PRINT"***COPY END***":RUN
1950 SYS875:PRINT
1960 PRINTER%=" " ***COPY ERRORS ***
1970 PRINT:PRINT"FOLGENDE DATEIN EXISTIEREN SCHON:":
1980 FORI=1TOER%:CP%=ER%(I):PRINTCP$(CP%):PRINT
1990 PRINT"  DESCHEN,  BENENNEN,  IGNORIREN ";
2000 SYS990:GETQ$:PRINTQ$:IFQ$="L"THENC$="S"+BI$+" "+CP$(CP%):PRINT#15,CC$:GOTO2030
2010 IFQ$="I"GOTO2050
2020 OPEN1,0,0:PRINT:PRINT"NEUER NAME ";:INPUT#1,FM$:CLOSE1:GOTO2040
2030 CC$="C"+BI$+" "+CP$(I)+"="+VO$+" "+CP$(I):PRINT#15,CC$:GOTO2050
2040 CC$="C"+BI$+" "+FM$+"="+VO$+" "+CP$(I):PRINT#15,CC$
2050 NEXT
2060 PRINT:PRINT"COPY END":CLOSE15:RUN
2070 REM DISK NEW
2080 PRINT"  CBM 3040 DISK FORMATOR"
2090 LN%=2:SP%=4:GOSUB2370:PRINT"FORMAT DRIVE ";
2100 SYS990:GETQ$:IFASC(Q$)<48ORASC(Q$)>49GOTO2100
2110 PRINT"  DRIVE  ";Q$
2120 VO$=Q$:VO%=VAL(Q$):BI%=NOTVO%AND1
2130 OPEN15,8,15:I$="I"+VO$:PRINT#15,I$
2140 INPUT#15,EC,EC$,EC$,SE$:ID$=""
2150 IFEC=0GOTO2190
2160 LN%=4:GOSUB2370:PRINT"BITTE DISK ID ";
2170 SYS990:GETID$:SYS990:GETQ$:ID$=ID$+Q$:PRINT" ";ID$
2180 GOTO2220
2190 LN%=4:GOSUB2370:PRINT"DISK ID BEIBEHALTEN (J/N) ";
2200 SYS990:GETQ$:IFQ$="N"GOTO2160
2210 IFQ$<>"J"GOTO2190
2220 LN%=6:GOSUB2370:PRINT"DISK NAME ";
2230 OPEN1,0,0:INPUT#1,NN$:CLOSE1
2240 CC$="N"+VO$+" "+NN$:IFID$>" "THENC$=CC$+" "+ID$
2250 LN%=8:GOSUB2370:PRINTMID$(CC$,4)
2260 PRINT#15,CC$
2270 INPUT#15,EC,EC$,EC$,SE$
2280 LN%=10:GOSUB2370:PRINT"DISKETTE FORMATIERT ";EC$
2290 IFEC>0THENPRINT"LETZTER VERSUCH: ";:PRINT#15,CC$
2300 INPUT#15,EC,EC$,EC$,SE$:IFEC>0THENPRINT"DISKETTE DEFEKT !  ":END
2310 BI%=RIGHT$(STR$(BI%),1):RESTORE
2320 READA$:CC$=" ":IFA$=" "THENRUN
2330 CC$="C"+VO$+" "+A$+" "+BI$+" "+A$
2340 PRINT#15,CC$:GOTO2320
2350 RUN
2360 DATA DUM3.4,DUP,$$
2370 PR%=871:GOTO2390
2380 PR%=826
2390 POKE216,LN%:POKE198,SP%:SYSPR%:RETURN
2400 REM DISK CHECK READ
2410 PRINT"  CBM - 3001 DISK CHECK READ"
2420 PRINT"(C)1979 BY DIPL.ING. U.KORNNAGEL
2430 POKE216,6:POKE198,5:SYS871:PRINT"CHECK READ ON DRIVE ";
2440 SYS990:GETQ$:RE%=VAL(Q$):EX=ASC(Q$):IFEX<48OREX>49GOTO2440
2450 PRINTRE%
2460 Q0$="I"+Q$:OPEN1,8,15,Q0$
2470 Q0$="V"+Q$:PRINT#1,Q0$:CLOSE1
2480 LN%=8:GOSUB3020:PRINT"BESTIMMTER BEREICH (J/N)";

```

65xx MICRO MAG

```

2490 SYS990:GETQ$:IFQ$="N"THENSVM=1:SBZ=35:GOTO2520
2500 LN2=8:GOSUB3020:OPEN1,0,0:PRINT"VON SPUR ":INPUT#1,SVZ:LN2=8:GOSUB3020
2510 PRINT"BIS SPUR ":INPUT#1,SBZ:CLOSE1
2520 POKE216,8:POKE198,3:SYS871:PRINT"SPUR ":SVZ;" - ";SBZ
2530 DIMDI$(35,20)
2540 OPEN15,8,15:OPEN3,8,4,"#"
2550 FORTR=SVZTOSBZ:S0=20
2560 IFTR>17THENS0=19
2570 IFTR>24THENS0=17
2580 IFTR>30THENS0=16
2590 FORSC=0TOS0:PRINT#15,"B-R";4;REZ;TR;SC
2600 INPUT#15,EC,EC$,EC%,SC%
2610 IFEC=22THENECS$="FREI":FRZ=FRZ+1
2620 IFEC=0THENECS$="BELEGT":BLZ=BLZ+1
2630 IFEC=20THENECS$="KEIN HEADER":KHZ=KHZ+1
2640 IFEC=21THENECS$="KEIN SYNCH":KSZ=KSZ+1
2650 IFEC=23THENECS$="BLOCK CHECKSUM":BCZ=BCZ+1
2660 IFEC=24THENECS$="DECODIER FEHLER":DCZ=DCZ+1
2670 IFEC=27THENECS$="HEADER CHECKSUM":HCZ=HCZ+1
2680 SKZ=SKZ+1
2690 POKE216,12:POKE198,0:SYS871:PRINT"SPUR";TR;" SECTOR";SC;" ";EC$
2700 DI$(TR,SC)=EC:NEXTSC,TR
2710 LN2=8:GOSUB3020:PRINT"GESAMT":GOSUB3030:PRINTSKZ:GOSUB3010
2720 LN2=10:GOSUB3020:PRINT"BELEGT":GOSUB3030:PRINTBLZ:GOSUB3010
2730 LN2=11:GOSUB3020:PRINT"FREI":GOSUB3030:PRINTFRZ:GOSUB3010
2740 LN2=12:GOSUB3020:PRINT"KEIN HEADER":GOSUB3030:PRINTKHZ:GOSUB3010
2750 LN2=13:GOSUB3020:PRINT"KEIN SYNCH":GOSUB3030:PRINTKSZ:GOSUB3010
2760 LN2=14:GOSUB3020:PRINT"BLOCK CHECKSUM FEHLER":GOSUB3030:PRINTBCZ:GOSUB3010
2770 LN2=15:GOSUB3020:PRINT"DECODIER FEHLER":GOSUB3030:PRINTDCZ:GOSUB3010
2780 LN2=16:GOSUB3020:PRINT"HEADER CHECKSUM FEHLER":GOSUB3030:PRINTHCZ:GOSUB3010
2790 LN2=18:GOSUB3020:PRINT"TOTALE FEHLERRATE":GOSUB3030
2800 PRINTKHZ+KSZ+BCZ+DCZ+HCZ:GOSUB3010
2810 CLOSE3:PRINT#15,"I":CLOSE15
2820 LN2=22:GOSUB3020:PRINT"SPUR - SECTOR ERROR - LISTE (J/N) ";
2830 SYS990:GETQ$:IFQ$="N"GOTO3000
2840 IFQ$<"J"GOTO2830
2850 PRINT"J","- ERROR-LISTE -":PRINT
2860 LN2=3:GOSUB3020:PRINT"01 BELEGTE SECTOREN"
2870 LN2=5:GOSUB3020:PRINT"02 FREIE SECTOREN"
2880 LN2=7:GOSUB3020:PRINT"03 SEKTOREN OHNE BLOCK HEADER"
2890 LN2=9:GOSUB3020:PRINT"04 SEKTOREN OHNE SYNCH ZEICHEN"
2900 LN2=11:GOSUB3020:PRINT"05 SEKTOREN MIT BLOCK CHECKSUM FEHLER"
2910 LN2=13:GOSUB3020:PRINT"06 SEKTOREN MIT DECODIER FEHLER"
2920 LN2=15:GOSUB3020:PRINT"07 SEKTOREN MIT HEADER CHECKSUM FEHLER"
2930 LN2=19:GOSUB3020:PRINT"BITTE CODE 01-07 EINGEBEN ";
2940 SYS990:GETQ$:QZ=VAL(Q$):IFQZ<10RQZ>7GOTO2940
2950 LN2=22:GOSUB3020
2960 ONQZGOSUB3040,3060,3080,3100,3120,3140,3160
2970 ONJPZGOTO3180
2980 LN2=22:GOSUB3020:PRINT"NICHT VORHANDEN":POKE216,19:POKE198,25:SYS826
2990 GOTO2940
3000 RUN
3010 POKE216,LN2:POKE198,32:SYS826:PRINT"BLOCKS":RETURN
3020 POKE216,LN2:POKE198,1:SYS871:RETURN
3030 POKE216,LN2:POKE198,25:SYS826:RETURN
3040 JPZ=0:IFBLZ>0THENJPZ=1:VCZ=0
3050 TX$="BELEGTE SECTOREN":RETURN

```

65xx MICRO MAG

```

3060 JPZ=0:IFFRZ>0THENJPZ=1:VCZ=22
3070 TX$="FREIE SECTOREN":RETURN
3080 JPZ=0:IFKHZ>0THENJPZ=1:VCZ=20
3090 TX$="SECTOREN MIT BLOCK OHNE HEADER":RETURN
3100 JPZ=0:IFKSZ>0THENJPZ=1:VCZ=21
3110 TX$="SECTOREN MIT BLOCK OHNE SYNCH ZEICHEN":RETURN
3120 JPZ=0:IFBZ>0THENJPZ=1:VCZ=23
3130 TX$="SECTOREN MIT BLOCK CHECKSUM FEHLER":RETURN
3140 JPZ=0:IFDCZ>0THENJPZ=1:VCZ=24
3150 TX$="SECTOREN MIT DECODIER FEHLER":RETURN
3160 JPZ=0:IFHCZ>0THENJPZ=1:VCZ=27
3170 TX$="SECTOREN MIT HEADER CHECKSUM FEHLER":RETURN
3180 PRINT"J",TX$:LNZ=3:GOSUB3020
3190 FORTR=SVZTOSBZ:S0=20
3200 IFTR>17THENS0=19
3210 IFTR>24THENS0=17
3220 IFTR>30THENS0=16
3230 FORSC=0TOS0:IFDI$(TR,SC)<>VCZGOTO3300
3240 PRINT"SPUR ";TR;" SECTOR ";SC:LNZ=LNZ+1
3250 IFLNZ<19THENGOSUB3020:GOTO3300
3260 LNZ=22:GOSUB3020:PRINT"WEITER BITTE >SPACE< ";
3270 SYS990:GETQ$:IFQ$<>" "THEN3270
3280 FORI=22TOSTEP-1:POKE216,I:SYS871
3290 NEXT LNZ=3:GOSUB3020
3300 NEXTSC,TR
3310 GOTO2820
3320 REM PUZZEL
3330 PRINT"J CBM 3001 -DISK APEND-XXX"
3340 INPUT"NEUER PROGRAMMNAME";NN$
3350 INPUT"1.) PROGRAMMNAME";P1$
3360 INPUT"2.) PROGRAMMNAME";P2$
3370 INPUT"0) NEUES PROGRAMM AUF DISK #";DR$
3380 DIMN$(2)
3390 N$(0)="0"+DR$+" "+NN$+"",PRG,WRITE"
3400 N$(1)=P1$+"",PRG,READ"
3410 N$(2)=P2$+"",PRG,READ"
3420 OPEN1,8,3,N$(0):PRINT:PRINTN$(0)
3430 OPEN2,8,4,N$(1):PRINT:PRINTN$(1)
3440 PRINT"LINE#:"
3450 GOSUB3630:GOSUB3660:GOSUB3630:GOSUB3660
3460 GOSUB3630:HL=AZ:GOSUB3630:HH=AZ:HE=256*HH+HL
3470 IFHE=0GOTO3530
3480 GOSUB3630:LL=AZ:GOSUB3630:LH=AZ:LN=256*LH+LL
3490 PRINTLN:IFPOS(0)>32THENPRINT
3500 GOSUB3670
3510 GOSUB3630:GOSUB3660:IFAZ>0GOTO3510
3520 GOTO3460
3530 CLOSE2:PRINT:PRINT:OPEN2,8,4,N$(2):PRINTN$(2)
3540 PRINT"LINE#:"GOSUB3630:GOSUB3630
3550 GOSUB3630:HL=AZ:GOSUB3630:HH=AZ:HE=256*HH+HL
3560 IFHE=0GOTO3620
3570 GOSUB3630:LL=AZ:GOSUB3630:LH=AZ:LN=256*LH+LL
3580 PRINTLN:IFPOS(0)>32THENPRINT
3590 GOSUB3670
3600 GOSUB3630:GOSUB3660:IFAZ>0GOTO3600
3610 GOTO3550
3620 PRINT#1,CHR$(0);CHR$(0);CHR$(0);CHR$(0):CLOSE1:CLOSE2:END
3630 GET#2,A$:IFA$=""THENA$=CHR$(0)

```

65xx MICRO MAG

```

540 IFST=64THENA$=CHR$(0)
550 A$=ASC(A$):RETURN
560 PRINT#1,A$:RETURN
570 A$=CHR$(HL):GOSUB3660:A$=CHR$(HH):GOSUB3660
580 A$=CHR$(LL):GOSUB3660:A$=CHR$(LH):GOTO3660
590 REM INIT CBM VIDEO 2
700 DATA169,0,162,128,164,216,240,9,24,105,40,144,1,232,136,208,247,133,196
710 DATA134,197,96,165,216,133,90,165,198,133,91,169,24,133,216,96,165,90,13
720 DATA216,165,91,133,198,16,211,169,32,208,7,169,64,208,3,173,255,3,41,127
730 DATA133,94,32,58,3,160,39,165,94,145,196,136,16,251,96,32,58,3,160,39,17
740 DATA196,73,128,145,196,136,16,247,96,32,80,3,32,58,3,164,198,177,196,73
750 DATA128,145,196,198,216,16,241,48,183,32,80,3,32,131,3,198,216,16,249,48
760 DATA171,169,32,208,7,169,93,208,3,173,255,3,41,127,133,94,32,80,3,32,58
770 DATA3,165,94,164,198,145,196,198,216,16,243,48,138,32,182,3,32,107,3,169
780 DATA219,76,210,255,162,16,165,143,164,158,208,17,197,143,240,248,202,208
790 DATA243,164,198,177,196,73,31,145,196,208,231,96,32,50,247,231,0,75,75,-
800 SD=826:RESTORE
810 READA$:IFA$<>"$":GOTO3810
820 READCD$:IFCD$<>0THENRUN1000
830 POKESD,CD$:SD=SD+1:GOTO3820

```

BUCHBESPRECHUNG

R.C. Camp, T. A. Smay und C. J. Triska: *Microprocessor Systems Engineering*, Verlag Matrix Publishers, Inc., Portland, OR, 1979, 641 S., ISBN: 0-916460-26-6, DM 64,- (englisch).

Das aus der Lehrtätigkeit der Autoren heraus entwickelte Fachbuch berücksichtigt in erster Linie die 65xx-Systemfamilie einschl. Interfacebausteinen und geht besonders auf den AIM 65 mit seiner Hardware und seinem Betriebssystem ein und gibt für ihn in besonderen Schritt-für-Schritt-Darstellungen Anleitungen für das richtige Handling, sei es nun im Monitorprogramm, im Texteditor oder im Assembler. Ein besonderes Kapitel behandelt Rockwells System 65 und zeigt Parallelen zum AIM auf. Zahlreiche Programmierbeispiele beziehen sich auch auf den KIM-1. Eigene kürzere Kapitel behandeln die CPUs 6800 und 8080 einschl. Befehlssätzen, um Gemeinsamkeiten und Unterschiede zu 65xx herauszustellen. Ein etwa 50-seitiges Kapitel stellt Interfacetechniken auch in Programmierbeispielen dar.

Dieses Buch dürfte vor allem für den AIM-Betreiber wertvoll sein, denn außer den Hinweisen für das Handling findet er auf etwa 140 Seiten sein System erklärt, und zwar in Schemazeichnungen für die Hardware, daneben Blockdiagramme für das sie bedienende Betriebssystem und in sachlichen Zusammenhängen die dazu gehörigen Listings aus dem Monitorprogramm. Das ganze wird durch didaktische Fragen und ausprogrammierte Anwendungsbeispiele gestützt.

Das sorgfältig aufgemachte Buch ist damit nicht spezifisch ein Programmierbuch für den Anfänger aber ein Buch zum besonderen Verständnis des AIMS und des dichten Zusammenhanges zwischen hardwaremäßigem Layout und durch sie bedingter Programmiermöglichkeit. Bezug durch Interface Age Europe, 8011 Vaterstetten, durch den Buchhandel und durch den Herausgeber.

Diesem Heft ist ein Prospekt der Firma Siemens AG beigelegt.

Martin Görnitz, Aldegundisstraße 54, 5400 Koblenz

6502 DIRECT ASSEMBLER

Zum in Heft 6 dieser Zeitschrift von K. Lehner veröffentlichten PET 6502-Assembler wurden verschiedene Ergänzungen und Verbesserungen nachgetragen. Es blieb seine Empfindlichkeit gegen Fehlbedienungen, die einen Abbruch der Assemblierung und den Verlust bisher erklärter Symbole und benutzter Labels verursachten. Zahlreiche Leser baten zwischenzeitlich um zusammenfassenden Abdruck des erreichten Standes. Das geschieht hiermit. Die erzeugten Maschinenprogramme können als selbstladende Files abgespeichert werden. Herr Görnitz hat weitere Leistungsmerkmale eingearbeitet. (Anm. d. Hrsg.)

Der PET-Assembler erwies sich hier als ein sehr nützliches Entwicklungsinstrument. Es wurden folgende Bearbeitungen vorgenommen: Die Speicheradressen werden hexadezimal ausgegeben und der erzeugte Opcode wird ebenfalls hexadezimal in die Befehlszeile geschrieben. Diese Maßnahme ist für eine Dokumentation wichtig, denn nun kann man auch Programme assemblieren, die auch auf anderen Maschinen laufen sollen und die nicht von der Programmcassette des PET aus einlesen können.

In den Zeilen 1003-1005 wird ein Printerausdruck über IEC-Bus durchgeführt. Es wird eine Zeile des Bildschirms in einen String umgewandelt und ausgegeben. Enthält die Zeile mehr als 40 Zeichen, so wird der Druck entspr. verlängert. Das Ergebnis sieht dann wie ein 'richtiges' Assemblerlisting aus.

Der Autor konnte allerdings ein Problem noch nicht lösen und bittet die Leser um Hinweise: Wenn man in den INPUT für eine Stringvariable mehr als 40 Zeichen schreibt, dann geht auch das Fragezeichen des INPUT als quasi-Eingabe in die Variable ein. Hieran stürzt der Assembler ab, wenn z.B. durch einen Kommentar mehr als 40 Zeichen eingegeben werden. Die Speicherstelle 146 (dez.) enthält nach INPUT die Zahl der Zeichen +10, und hier ist der Fehler schon drin.

0 REM***MICROMAG'S ASSEMBLER, MIT OPCODE UND ADRESSEN IN HEXADEZIMAL*****

```

1 AN=65530:I$="":OPEN2,6
2 INPUT"AUSDRUCK ";I$:IFI$="J"THENPR=1
3 I$=""
10 PRINT"";TAB(13);"6502 ASSEMBLER":PRINT
20 PRINT"SOURCECODE ZEILENWEISE EINGEBEN;"
30 PRINT"SPEICHERRAUM: 7168-8191":PRINT
40 PRINTTAB(27);""
49 IFFR=0THEN100
50 PRINT#2,"          6502 DIRECT ASSEMBLER BY M. GOERLITZ"
51 PRINT#2,"          =====":PRINT#2:PRINT#2
100 DEF FNB(V)=V-256:FNA(V)
101 DEF FNA(X)=INT(X/256)
150 DIMLB$(59),LB(59),UL$(29),UL(29)
500 DATA,ADC,105,109,101,97,113,117,125,121,AND,41,45,37,33,49,53,61,57,ASL
501 DATA10,14
510 DATA6,,,22,30,BCC,144,BCS,176,BEQ,240,BIT,,44,36,BMI,48,BNE,208,BPL,16,BRC
0
520 DATABVC,80,BUS,112,CLC,24,CLD,216,CLI,88,CLV,184,CMP,201,205,197,193,209,2
3

```


65xx MICRO MAG

```

530 DATA221,217,CPX,224,236,228,CPY,192,204,196,DEC,,206,198,,,214,222,DEX,202
540 DATADEV,136,EOR,73,77,69,65,81,85,93,89,INC,,238,230,,,246,254,INX,232,INY
550 DATA200,JMP,108,76,JSR,,32,LDA,169,173,165,161,177,181,189,185,LDX,162,174
560 DATA166,,,182,190,LDY,160,172,164,,,180,188,LSR,74,78,70,,,86,94,NOP,234,0
RA
570 DATA9,13,5,1,17,21,29,25,PHA,72,PHP,8,PLA,104,PLP,40,ROL,42,46,38,,,54,62,R
OR
580 DATA106,110,102,,,118,126,RTI,64,RTS,96,SBC,233,237,229,225,241,245,253,249
590 DATASEC,56,SED,248,SEI,120,STA,,141,133,129,145,149,157,153,STX,,142,134,,,
600 DATA150,STY,,140,132,,,148,TAX,170,TAY,168,TSX,186,TXA,138,TXS,154,TYA,152
1000 PRINT":":IFLEN(A$)>21THENPRINT"";
1001 PRINTI$:I$="":POKE242,39
1002 IFFR=0THEN1006
1003 J=32768+40*(PEEK(245)-1):I1=39:IFLEN(A$)>21THENJ=J-40:I1=71
1004 DU$=""":FORI=0TOI1:DU=PEEK(I+J):IFDU<32THENDU=DU+64
1005 DU$=DU$+CHR$(DU):NEXT:PRINT#2,DU$:NN=NN+1:IFNN>54THENPRINT#2,CHR$(12):NN=0
1006 DI=FNA(PC):GOSUB30000:I$=LEFT$(I$,2):DI=FNB(PC):GOSUB30000:PRINTI$:TAB(18)
1008 I$="":OPEN1,0,1:INPUT#1,A$:FORX=10TO50:IFPEEK(X)=0THENL=X-1:X=50
1010 NEXT:A$="":FORX=10TO1:A$=A$+CHR$(PEEK(X)):NEXT:CLOSE1:F=0
1015 PRINT:RESTORE:R$="":
1020 IFBG=0ANDMID$(A$,10,2)<>"*"THENPRINT"PC = 0 ERROR":GOTO1000
1025 IFBG=1ANDPC=0ANDPC<ANTHENAN=PC
1026 IFEN<PCTHENEN=PC
1030 BG=1:T=1:GOSUB9000:T$=LEFT$(CB$,1)
2020 IFT$="":THEN1000
2030 IFT$="":THEN7000
2040 IFT$="*"THENP$=MID$(CB$,3):GOSUB20000:PC=U:GOTO1000
3000 IFLEN(CB$)<>3THEN3050
3020 READR$:IFR$="152"THENRESTORE:R$="":GOTO3060
3030 IFCB$=R$THENOP$=CB$:GOTO5000
3040 GOTO3020
3050 IFCB$="A"THENPRINT"LABEL FALSCH":GOTO1000
3060 K=0
3070 IFLB$(K)=CB$THENPRINT"LABEL DOPPELT":GOTO1000
3080 K=K+1:IFK<NTHEN3070
3090 LB$(N)=CB$:LB(N)=PC:N=N+1:GOSUB9000:IFLEFT$(CB$,1)<>"*"THEN4030
4020 P$=MID$(CB$,5):GOSUB20000:PC=PC+U:GOTO1000
4030 IFLEFT$(CB$,1)<>"*"THEN4050
4040 P$=MID$(CB$,2):GOSUB20000:LB(N-1)=U:GOTO1000
4050 IFLEFT$(CB$,1)="":THEN7000
4060 OP$=CB$
5000 GOSUB9000:AD$=CB$
5020 IFLEFT$(AD$,1)="#"THEN5700
5025 IFLEFT$(AD$,1)="#"THENAD$=""
5030 IFAD$="A"ORAD$=""THENAD=0:GOSUB5600:GOTO1000
5040 IFRIGHT$(AD$,3)=",X"THENAD=3:P$=MID$(AD$,2,LEN(AD$)-4):GOTO5900
5050 IFRIGHT$(AD$,3)=",Y"THENAD=4:P$=MID$(AD$,2,LEN(AD$)-4):GOTO5900
5060 IFRIGHT$(AD$,1)=":"THENAD=0:P$=MID$(AD$,2,LEN(AD$)-2):GOTO5920
5070 IFRIGHT$(AD$,2)=",X"THEN5200
5080 IFRIGHT$(AD$,2)=",Y"THEN5300
5090 P$=AD$:GOSUB20000:IFLEFT$(OP$,1)="#"ANDOP$<>"BIT"ANDOP$<>"BRK"THEN5400
5100 IFLEFT$(OP$,1)="#"ANDF=1THEN5500
5110 IFU<256ANDLEFT$(OP$,1)<>"J"THENAD=2:GOTO5910
5120 AD=1:GOTO5930

```

```

5200 P$=LEFT$(AD$,LEN(AD$)-2):GOSUB20000:IFV(256THENAD=5:GOTO5910
5210 AD=6:GOTO5930
5300 P$=LEFT$(AD$,LEN(AD$)-2):GOSUB20000:IFV(256THENAD=6:GOTO5910
5310 AD=7:GOTO5930
5400 AD=0:GOSUB5600:IFF=1THENUL$(M)=P$:UL$(M)=PC:M=M+1:PC=PC+1:GOTO1000
5405 DI=V-PC-1:IFABS(DI)>127THENPRINT"DISTANZFEHLER:";AD$:GOTO1000
5410 IFDI<0THENDI=DI+256
5420 POKEPC,DI:PC=PC+1:GOSUB30000:GOTO1000
5500 AD=1:GOSUB5600:UL$(M)=P$:UL$(M)=-PC:M=M+1:PC=PC+2:GOTO1000
5600 IFR$="152"THENPRINT"KEIN OPCODE !":GOTO1000
5620 IFR$>OP$THENREADR$:GOTO5600
5630 FORK=0TODR:READR:NEXT:POKEPC,R:PC=PC+1:DI=R:GOTO30000
5700 AD=0:GOSUB5600
5710 IFMID$(AD$,2,1)=">"THENGOSUB5750:DI=FNA(U):POKEPC,DI:GOSUB30000:GOTO5740
5720 IFMID$(AD$,2,1)("<"THENGOSUB5750:DI=FNB(U):POKEPC,DI:GOSUB30000:GOTO5740
5730 P$=MID$(AD$,2):GOSUB20000:POKEPC,U:DI=U:GOSUB30000:IFF=1THEN7515
5740 PC=PC+1:GOTO1000
5750 P$=MID$(AD$,3):GOSUB20000:IFF=1THEN7515
5760 RETURN
5900 GOSUB20000
5910 IFF<>1THENGOSUB5600:POKEPC,U:DI=U:GOSUB30000:GOTO5740
5915 GOTO7515
5920 GOSUB20000
5930 IFF=1THEN5935
5931 GOSUB5600:DI=FNB(U):POKEPC,DI:GOSUB30000:PC=PC+1:DI=FNA(U):POKEPC,DI
5932 GOSUB30000:GOTO5740
5935 GOTO7515
7000 F$=MID$(CB$,2,3):IFF$="END"THEN8000
7020 IFF$="BYT"THEN7500
7030 IFF$="WOR"THEN7600
7500 GOSUB9000:IFLEFT$(CB$,1)="/"THEN7510
7501 P$=CB$:GOSUB20000:POKEPC,U:DI=U:GOSUB30000:PC=PC+1:GOTO7515
7510 FORK=2TOLEN(CB$)-1:DI=ASC(MID$(CB$,K,1)):POKEPC,DI:GOSUB30000:PC=PC+1:NEXT

7515 IFF=1THENPRINT"REFERENZ FALSCH:";P$:GOTO1000
7520 IFT>LEN(A$)THEN1000
7530 GOTO7500
7600 GOSUB9000:P$=CB$:GOSUB20000:DI=FNB(U):POKEPC,DI:GOSUB30000
7601 PC=PC+1:DI=FNA(U):POKEPC,DI:GOSUB30000:PC=PC+1
7605 IFF=1THEN7515
7610 IFT>LEN(A$)THEN1000
7620 GOTO7600
8000 IFM=0THEN8100
8015 FORK=0TOM-1:L=0
8020 IFL>NTHENPRINT"LABEL UNBEKANNT:";UL$(K):END
8030 IFLB$(L)>>UL$(K)THENL=L+1:GOTO8020
8035 IFUL$(K)<0THENPOKE-UL$(K),FNB(LB(L)):POKE-UL$(K)+1,FNA(LB(L)):GOTO8055
8040 DI=LB(L)-UL$(K)-1
8041 IFABS(DI)>>127THENPRINT"DISTANZFEHLER:";UL$(K):END
8050 POKEUL$(K),DI
8055 NEXT
8100 PRINT"END OF ASSEMBLY":PRINT
8102 IFPR=0THEN8110
8104 PRINT#2,"                                .END":PRINT#2
8106 PRINT#2,"          *** END OF ASSEMBLY ***":PRINT#2:PRINT#2
8110 PRINT"NUMBER OF LABELS:";TAB(24);N
8120 PRINT"NUMBER OF FORWARD REF.:";TAB(24);M
8130 INPUT"ABSPEICHERN";W$:IFW$="N"THENEND
8135 IFW$<>"J"THEN8130

```

```

8140 INPUT "NAME";W$:AC%=256+PEEK(125)+PEEK(124):WH%=INT(AN/256)
8142 WL%=AN-256*WH%:POKE247,WL%:POKE248,WH%:POKE229,FNB(EN+1):POKE230,FNA(EN+1)

8144 LN%=PEEK(AC%+2):WL%=PEEK(AC%+3):WH%=PEEK(AC%+4)
8146 POKE238,LN%:POKE249,WL%:POKE250,WH%:OPEN3,1,1,W$:SYS(63153):END
9000 CB$=""
9015 T$=MID$(A$,T,1):IFT$="" THEN T=T+1:GOTO9015
9020 IFT>LEN(A$) THEN RETURN
9030 T$=MID$(A$,T,1):IFT$="" THEN RETURN
9040 CB$=CB$+T$:T=T+1:GOTO9020
20000 R=1:W=0:F=0:AR=0
20010 L$="":U=0:BA=0
20020 P1$=MID$(P$,R,1):IF P1$>="0" AND P1$<="9" THEN BA=10:GOTO25010
20030 IF P1$=" $" THEN BA=16:GOTO25000
20040 IF P1$="@" THEN BA=8:GOTO25000
20050 IF P1$="%" THEN BA=2:GOTO25000
20060 IF P1$="/" THEN U=ASC(MID$(P$,R+1,1)):RETURN
20070 L$=L$+P1$:R=R+1:P1$=MID$(P$,R,1):IF R>LEN(P$) THEN 26000
20080 IF P1$="+" THEN AR=1:GOTO20200
20090 IF P1$="-" THEN AR=2:GOTO20200
20100 IF P1$="*" THEN AR=3:GOTO20200
20110 IF P1$="/" THEN AR=4:GOTO20200
20120 GOTO20070
20200 GOSUB26000:W=U:R=R+1:GOTO20010
25000 R=R+1:P1$=MID$(P$,R,1)
25010 IF R>LEN(P$) THEN 28000
25020 P1=ASC(P1$):U=U+BA-(P1<60)*(P1-48)-(P1>60)*(P1-55):GOTO25000
26000 K=0
26010 IF K=N THEN F=1:RETURN
26015 IFL$(<>LB$(K)) THEN K=K+1:GOTO26010
26020 U=LB$(K):IF R<=LEN(P$) THEN RETURN
28000 IF AR=0 THEN RETURN
28010 ON AR GOTO 28100,28110,28120,28130
28100 U=W+U:RETURN
28110 U=W-U:RETURN
28120 U=W*U:RETURN
28130 U=W/U:RETURN
30000 H=INT(DI/16):H1=DI-H*16:H=H+48:IF H>57 THEN H=H+7
30001 I$=I$+CHR$(H):H1=H1+48:IF H1>57 THEN H1=H1+7
30002 I$=I$+CHR$(H1)+" ":RETURN
READY.

```

CURSOR: IN ZEILE 1000 IN DEN ANFUEHRUNGSZEICHEN 2*DOWN, 3*UP UND
7*RIGHT !
IM ZWEITEN TEIL (IF-ANW.) NUR 1*UP!

DIES BRINGT DEN CURSOR AN DIE STELLE, WO DER OPCODE EINZUFUEGEN IST. ●

KLEINANZEIGE

PET 2001, 8 KB, EIN JAHR ALT, UND VIDEOKARTE XITEX SCT 100 MIT
128 DARSTELLBAREN ZEICHEN, BISHER AM AIM GELAUFEN, WEGEN UMRÜS-
TUNG ZU VERKAUFEN. ROLAND LÖHR, HANS DORFER STR. 4, 2070 AHRENS-
BURG, TEL.: 04102 - 55 816 (AUCH ABENDS).

SCHAUFELE

*** SCHAUFEL ***

65_{xx} MICRO MAG

65xx MICRO MAG

```

450 NEXT I:PRINT'ANZAHL DER UMGESETZTEN BEFEHLE:'U
460 REM
470 PRINT'↓↑↑↑ UMSETZUNG DER EINZELBYTE-ADRESSEN '
480 PRINT'↓ADRESSEN (ALT) HIGH-ORDER UND LOW-ORDER BYTE EINGEBEN!
490 INPUT'VH,L ,<<<<'H,L:IFH=0THEN560
500 H=H+D:L=L+D:Y=PEEK(H)*256+PEEK(L)
510 IFY>BORY<VTHENPRINT'UNGÜLTIGE ADRESSE! INHALT='Y:GOTO490
520 PRINT,'VON'YTAB(22)'IN'Y+D
530 Y=Y+D:Y%=Y/256:Y=Y-Y%*256
540 POKEL,Y:POKEH,Y%
550 GOTO490
560 REM
570 PRINT'↓↑↑↑UMSETZUNG DER EXTERNEN ADRESSEN! GGF. RESTART MIT 'GOT
0560'
580 INPUT'↓VON,BIS ,<<<<'VE,BE:IFVE=0THEN660
590 PRINT'↓FOLGENDE ADRESSEN WERDEN UMGESETZT: 'U=0
600 FORI=VETOBE:Z=PEEK(I):X=ZAND15
605 IF(ZAND25)=25THENGOSUB670:GOTO640
610 IFX=0THENGOSUB770:GOTO650
620 IFX=3ORX=8ORX=10THEN650
630 IFX>=12THENGOSUB670
640 I=I+1
650 NEXT I:PRINT'ANZAHL DER UMGESETZTEN BEFEHLE:'U
660 PRINT'↓↑↑↑ E N D O F J O B ' :END
670 REM *****
680 I=I+1:H=0:IFAV(H)=0THEN710
690 IFI>=AV(H)THENIFI<AB(H)THENRETURN
700 IFH<0THENH=H+1:GOTO690
710 Y=PEEK(I)+PEEK(I+1)*256
720 IFY>BORY<VTHENRETURN
730 PRINT'AUF ACK.'I-ITAB(17)'VON'YTAB(29)'IN'Y+D
740 Y=Y+D:Y%=Y/256:Y=Y-Y%*256
750 POKEL,Y:POKEI+1,Y%:L=U+1
760 RETURN
770 REM *****
780 IFZ=0ORZ=64ORZ=96ORZ=128THENRETURN
790 IFZ<>32THENI=I+1:RETURN
800 GOSUB670:I=I+1
810 RETURN
READY.

```

Peter W. Arps, Hamburg

VARLIST

Bei der Entwicklung und beim Test umfangreicher BASIC-Programme verliert man leicht die Übersicht über den momentanen Inhalt der Variablen. VARLIST listet den jeweiligen Stand aller BASIC-Variablen, der einfachen und der dimensionierten, mit Namen der Variablen und ihrem Inhalt.

VARLIST ist mit SYS(7800) zu starten. Die Ausgabe kann jederzeit durch Drücken der SHIFT-Taste abgebrochen werden. Die einzelnen Variablen erscheinen mit Namen und Inhalt. Bei dimensionierten Variablen (Arrays) erscheint der Name einmal in reverser Darstellung am Anfang der Gesamtvariablen. Vor den einzelnen Elementen wird ihre Index-Nummer angezeigt. Bei Arrays mit mehr als 256 Elementen beginnt nach der Indexnummer 255 wieder die Nummer 0.

65xx MICRO MAG

```

7800 A5 7C      START LDA VAR
7802 85 50      STA X1
7804 A5 7D      LDA VAR+1
7806 85 51      STA X1+1
7808 D8         CLD
7809 A5 51      LOOP1 LDA X1+1      ;ENDE DER EINF. VARIABLEN?
7811 C5 7F      CMP ARR+1
7813 90 09      BCC W1            ;NEIN
7815 A5 50      LDA X1
7817 C5 7E      CMP ARR
7819 90 03      BCC W1            ;NEIN
7821 4C C0 1E    JMP ARRAY        ;JA
7824           ;
7824 AC 04 02    W1 LDY SHIFT      ;ABBRUCH?
7827 D0 28      BNE RTN           ;JA
7829 20 98 1F    JSR NAME         ;VARIABLEN-NAME AUSGEBEN
7832 B0 1D      BCS W4            ;UEBERGEHEN
7834 A0 02      LDY #2
7836 A5 55      LDA X4            ;MERKER LADEN
7838 F0 12      BEQ W3            ;IST FLP.
7840 C9 24      CMP #'$
7842 F0 06      BEQ W2            ;IST STRING
7844 20 55 1F    JSR INTG         ;IST INTEGER
7847 4C B7 1E    JMP W4
7850 A0 04      W2 LDY #4
7852 20 7F 1F    JSR STRING
7855 4C B7 1E    JMP W4
7858 A0 06      W3 LDY #6
7860 20 6A 1F    JSR FLP
7863 20 E8 1F    W4 JSR INCX17
7866 4C 81 1E    JMP LOOP1
7869           ;
7869           ;
7869 4C 8B C3    RTN JMP BASIC     ;ZURUECK ZU BASIC
7872           ;
7872 A5 51      ARRAY LDA X1+1     ;ARRAY-ENDE ERREICHT?
7874 C5 81      CMP ARREND+1
7876 90 06      BCC W5
7878 A5 50      LDA X1
7880 C5 80      CMP ARREND
7882 B0 F1      BCS RTN           ;JA
7884 A5 E2      W5 LDA PRADR
7886 F0 03      BEQ W5A
7888 20 50 1F    JSR CR
7891 A9 12      W5A LDA #18       ;NAME REVERSE AUSGEBEN
7893 20 30 F2    JSR WRITE
7896 20 98 1F    JSR NAME
7899 20 50 1F    JSR CR
7902 A0 02      LDY #2
7904 B1 50      LDA (X1),Y        ;RELATIVEN POINTER FUER NAECHSTE ...

7906 18         CLC              ;...VARIABLE LADEN
7907 65 50      ADC X1            ;UND UM ADR DER VARIABLEN ERHOEHEN

7909 85 56      STA X5
7911 C8         INY
7912 B1 50      LDA (X1),Y
7914 65 51      ADC X1+1

```

65xx MICRO MAG

```

7916 85 57          STA X5+1
7918 20 E8 1F      JSR INCRX17    ;X1 AUF 1. ELEMENT VORSETZEN
7921 20 41 1F      LOOP2 JSR SETTAE
7924 A4 58          LDY DIM        ;ELEMENTZAEHLER LADEN
7926 A9 00          LDA #0
7928 20 78 D2      JSR FLOAT      ;FLOATEN
7931 20 AF CC      JSR ASCII      ;IN ASCII
7934 A9 01          LDA #1
7936 A8            TAY
7937 20 27 CA      JSR PUT        ;UND AUSGEBEN
7940 20 D8 1F      JSR TAB        ;TAB(5) ODER TAB(25) SETZEN
7943 A5 55          LDA X4
7945 F0 0D          BEQ W6        ;FLOATING
7947 C9 25          CMP #1%
7949 F0 12          BEQ W7        ;INTEGER
7951 A0 02          LDY #2
7953 20 7F 1F      JSR STRING
7956 A9 03          LDA #3
7958 D0 10          BNE W8        ;IMMER
7960 A0 04          W6 LDY #4
7962 20 6A 1F      JSR FLP
7965 A9 05          LDA #5        ;LAENGE DER VARIABLEN
7967 D0 07          BNE W8        ;IMMER
7969 A0 00          W7 LDY #0
7971 20 55 1F      JSR INTG
7974 A9 02          LDA #2
7976 AE 04 02      W8 LDX SHIFT    ;ABBRUCH?
7979 D0 90          BNE RTN        ;JA
7981 E6 58          INC DIM        ;ELEMENTZAEHLER ERHOEHEN
7983 20 EA 1F      JSR INCX1      ;POINTER VORSETZEN
7986 A5 51          LDA X1+1      ;ENDE DER VARIABLEN?
7988 C5 57          CMP X5+1
7990 90 B9          BCC LOOP2      ;NEIN
7992 A5 50          LDA X1
7994 C5 56          CMP X5
7996 B0 82          BCS ARRAY      ;JA
7998 4C F1 1E      JMP LOOP2
8001                ;
8001 A5 E2          SETTAE LDA PRADR    ;1. SPALTE?
8003 F0 0A          BEQ STE1        ;JA
8005 C9 14          CMP #20        ;2. HAELFTE
8007 F0 06          BEQ STE1        ;JA
8009 10 0E          BPL CR
8011 A9 14          LDA #20        ;TAB(20) SETZEN
8013 95 E2          STA PRADR
8015 60            STE1 RTS
8016                ;
8016                ;
8016                ;
8016 A9 0D          CR LDA #12
8018 4C 30 F2      JMP WRITE
8021                ;
8021                ;
8021                ;
8021 B1 50          INTO LDA (X1)+Y    ;MSB LADEN
8023 AA            TAX
8024 CB            INY
8026 B1 50          LDA (X1)+Y    ;LSB LADEN
8027 A2            TAY

```

65xx MICRO MAG

```

8028 8A          TXA
8029 20 79 D2   JSR FLOAT      ;FLOATEN
8032           ;
8032           ;
8032 20 AF DC   OUT    JSR ASCII  ;FLP-ACCU IN ASCII
8035 A9 00      LDA #0
8037 A0 01      LDY #1
8039 4C 27 CA   JMP PUT        ;AUSGEBEN
8042           ;
8042           ;
8042 A2 04      FLP    LDX #4
8044 E1 50      F1     LDA (X1),Y  ;VAR. IN DEN FLP-ACCU
8046 95 B0      STA ACC,X
8048 88         DEY
8049 CA         DEX
8050 10 F3      BPL F1
8052 A5 21      LDA ACC+1        ;1.BYTE DER MANTISSE LADEN
8054 85 25      STA ACC+5        ;INS VORZEICHEN-BYTE
8056 09 20      ORA #129        ;VORZEICHENBIT LOESCHEN
8058 95 E1      STA ACC+1        ;STORE MANTISSE
8060 4C 60 1F   JMP CLT
8062           ;
8062           ;
8063 A2 02      STRING LDX #2
8065 E1 50      S1     LDA (X1),Y  ;ADR. + LAENGE DES STRINGS LADEN
8067 95 52      STA X2,X        ;UND IN X2+X2 SETZEN
8069 88         DEY
8070 CA         DEX
8071 10 F3      BPL S1
8072 AA         TAX              ;LAENGE UNLADEN
8074 F0 03      BEQ S3          ;IST EIN NULL-STRING
8076 A0 00      LDY #0
8078 E1 50      S2     LDA (X3),Y  ;AUSGEBEN
8080 20 20 F2   JSR WRITE
8082 C2         INY
8084 CA         DEX
8086 D0 F7      BNE S2
8087 60         RTS

8088 20 41 1F   NAME    JSR SETTAB
8091 A0 01      LDY #1
8092 81 51      LDA (X1),Y        ;2.ZEICHEN HOLEN
8095 85 52      STA X2+1          ;UND SICHERN
8097 88         DEY
8098 84 55      STY X4            ;MERKER LOESCHEN
8100 84 56      STY D1M
8102 E1 50      LDA (X1),Y        ;1.ZEICHEN HOLEN
8104 95 52      STA X2            ;UND SICHERN
8106 10 0A      BPL N2           ;IST FLP. ODER STRING
8108 24 53      BIT X2+1
8110 30 02      BMI N1           ;INTEGER
8112 30         SEC              ;IST FUNKTION
8113 60         RTS              ;UEBERGEBEN
8114           ;
8114 A2 25      N1     LDX #1X
8116 D0 03      BNE N3           ;IMMER
8118 24 52      N2     BIT X2+1
8120 10 0A      BPL N4           ;FLP
8122 A2 24      LDX # 4          ;STRING

```

```

8124 86 55      N3      STX X4
8126 29 7F      N4      AND #127      ;BIT 7 LOESCHEN
8128 20 30 F2    JSR WRITE      ;1.ZEICHEN AUSGEBEN
8131 A5 53      LDA X2+1      ;2.ZEICHEN
8133 29 7F      AND #127
8135 F0 03      BEQ N5        ;IST LEER
8137 20 30 F2    JSR WRITE
8140 A5 55      N5      LDA X4
8142 F0 03      BEQ N6        ;KEIN ZUSATZ
8144 20 30 F2    JSR WRITE
8147 A9 3D      N6      LDA #'=
8149 20 30 F2    JSR WRITE
8152            ;
8152            ;
8152 A9 14      TAB      LDA #20
8154 C5 E2      CMP FRADR
8156 90 04      BCC T1
8158 A9 05      LDA #5
8160 D0 02      BNE T2
8162 A9 19      T1      LDA #25
8164 85 E2      T2      STA FRADR
8166 18         CLC
8167 60         RTS
8168            ;
8168            ;
8168 A9 07      INCX17 LDA #7
8170 18         INCX1 CLC
8171 65 50      ADC X1
8173 85 50      STA X1
8175 90 02      BCC RX1
8177 E6 51      INC X1+1
8179 60      RX1      RTS
8180            ;
8180            ;
8180            .END

```

Spezial-Netzgerät Typ MSB 3040

Technische Daten:

Netz 220V, -50/60 Hz

Spannung 1: +5V

Spannung 2: +24V

Spannung 3: +12V

Spannung 4: -12V

Nennstrom 1: -5A

Nennstrom 2: -1A

Nennstrom 3: -1A

Nennstrom 4: -1A

Preis: DM 415.- inkl. 13 % MwSt.

Made in Germany

Ideal für Ihren AIM 65, der bei uns auch ab Lager lieferbar ist.

MSB MICRO
POSTBIL 772
MARKTSTADT 5
D 7765 MARKT



MICRO-
SHOP
Bodensee

AIM SPEZIAL (6) Aus Platzgründen diesmal nur eine kurze Rubrik. Weitere Fehler des Monitorprogrammes sind bekannt geworden. Herr H.O. Hinrichs aus Bremen wies darauf hin, daß der Assembler den Quelltext STA 0000,X partout nicht als absolute Adressierung akzeptiert, sondern daraus einen Zeropagebefehl in 2 Bytes erzeugt. Weiter bemerkte er, daß der DUMP auf Cassette unter Monitor kein Wiederlesen mit erkanntem File-Ende ermöglicht, wenn die Menge der Bytes \$6C ist. Herr Georg Koch aus Ingolstadt machte die gleiche Beobachtung auch für die Mengen 6B und 6D. Erhöht man die Menge auf 6D, so wird ein zusätzlicher Datenblock übertragen, und der Fehler tritt nicht auf. Ob es sich um eine fehlerhafte Programmierung beim DUMP im Bereiche von DU2 (\$E47D ff.) oder beim Laden handelt, ist noch nicht aufgeklärt.

Prellende Tastaturen am AIM werden jetzt häufiger gemeldet. Abhilfe: Baden der Tastatur lauwarm mit Reinigungsmittel 'Meister Proper', Spülen mit PRIL, ausschütteln, trocknen, föhnen. Empfohlen von Herrn Palm, Hannover, hier erfolgreich nachvollzogen (ohne Gewährleistung!).

Frage: Wurde irgendwo schon die Commodore Floppy 3040 an den AIM angeschlossen?

Michael Zimmermann, Pfungstadt

Ein Printer für den AIM

Als häufiger Benutzer des AIM ist der Verfasser durchaus mit den Schwachstellen dieses Gerätes vertraut und ist immer auf der Suche nach Möglichkeiten, diese zu umgehen. Auf den ersten Blick mag es paradox klingen, aber der kleine Printer auf der Platine ist durchaus als eine derartige Störstelle anzusehen. Im Vergleich mit anderen Systemen zeichnet der eingebaute Drucker den AIM sicher aus, hier wurde jedoch sehr gespart und ein Modell genommen, das nicht voll auf befriedigen kann. Deshalb sann der Verfasser, bestärkt durch Ausfälle, auf Abhilfe und begann, sich nach einem 'richtigen' Drucker umzusehen.

Gewünscht wurde ein Normalpapierdrucker mit 80 Stellen. Leider waren alle angebotenen Fabrikate, die eine dauerhafte Lösung versprachen, jenseits von DM 2000,- und standen damit in keiner vernünftigen Relation zum Preis des AIM. Sicher werden gebrauchte Selectric-Schreibwerke oder 5-Kanal-Fernschreiber günstiger angeboten, ein sinnvoller Drucker zum AIM sollte aber ein Matrixdrucker sein, der mit einem Minimum von Hardware betrieben werden kann und der dabei gleichzeitig die im Monitor-ROM sowieso vorhandenen Dot-Tabellen für die Zeichengenerierung nutzt.

Der Autor wurde auf das von der Firma Dohmann in Gütersloh angebotene Druckwerk aufmerksam gemacht. Dieses erschien in Preis und Leistung einmalig dazustehen und den gewünschten Kriterien genau zu entsprechen. In den ersten Tagen des neuen Jahres erreichte den Verfasser ein Gerät, das seinerzeit noch Prototyp war.

Der Anschluß stellte sich einfach dar, es war lediglich eine 14-polige Verbindung zwischen Anwendungsstecker des AIM und der Treiberplatine erforderlich, die mit DIL-Steckern und Flachbandkabel aufgebaut wurde. Für die Stromversorgung wird ein 2x24 Volt Transformator benötigt, der direkt an die Treiberplatine anzuschließen ist. Diese Platine enthält die erforderliche Gleichrichtung und die Transistoren für die Ansteuerung von Motor und Solenoiden. Wegen dieser minimalen Hardware müssen alle Ansteuerungen per Software erfolgen. Die Schnittstelle ist hierbei das Anwender-VIA.

Das Betriebsprogramm wurde als Assembler-Listing mitgeliefert, es war in ca. 2 Stunden eingegeben und auf die besonderen Wünsche modifiziert. Eine Ansteuerung über das DILINK oder auch über User-Output ist problemlos möglich. Als Konsequenz hat der Verfasser den eingebauten Thermodrucker vollständig deaktiviert und arbeitet jetzt seit über einem Monat vollkommen störungsfrei über den neuen Normalpapier-Printer.

Vom mechanischen Aufbau mach das Druckwerk, ein schwedisches Fabrikat, einen sehr soliden Eindruck. Alle tragenden Teile sind aus 2 mm Stahlblech oder 6 mm Stabstahl aufgebaut. Überhaupt hat man den Eindruck, daß in stärkerem Maß Metall verwendet wurde, als bei vergleichbaren Fabrikaten. Lediglich nichttragende Teile sowie Zahnräder und Kurvenscheiben, letztere wegen Geräuscharm und Schmierung, sind aus Kunststoff. Alle erforderlichen Bewegungen, sowohl für den Druckkopf als auch für den Paplervorschub, werden von einem einzigen Motor abgenommen.

Auch im Betrieb enttäuschte der Drucker bis jetzt nicht und stellte eine deutliche Verbesserung gegenüber dem Thermoprinter in Hinsicht auf Leistung und Störsicherheit dar. In der Geräuschkentwicklung, die ja bei Matrixdruckern kritisch zu beobachten ist, zeichnet sich dieses Druckwerk positiv aus. Nach subjektivem Eindruck ist es nicht lauter, als ein IBM-Druckwerk im Gehäuse.

Es gab keinerlei thermische Probleme am Druckkopf. Alles in allem: Eine sinnvolle Anschaffung für jeden ernsthaften AIM-Betreiber.

Peter Trübger, Melissenweg 37, 2000 Hamburg 65

CBM SCHIESST SICH EIGENE EPROMs

Es wurde ein 2 kByte-EPROM-Programmiergerät entwickelt, das fast ohne zusätzliche Hardware auskommt und das vom PET/CBM über USER-Port und IEC-Bus gesteuert wird. Ein solches Gerät empfiehlt sich, hat der neue CBM 3032 doch immerhin 3 freie Sockel für 12 kByte und die CBM-Floppy einen für 4 kByte. Zwei EPROMs können in diese Sockel gesetzt werden und können dem PET z.B. Umlaute und das 'ß' beibringen, wenn sie als character-ROM eingesetzt werden. - Schaltung und Programmlisting finden sich am Schluß.

Da der User-Port beim Einschalten HIGH ist, dürfen die 25 Volt erst nach erstem RUN ans EPROM angelegt werden. Dann ist einfach noch einmal RUN einzugeben.

Zum Programmieren von 4 kByte-EPROMs muß die Hard- und Software geringfügig modifiziert werden (daher einfache Programmierung in BASIC). Nur der TMS 2532 von TI läßt sich direkt in die freien Sockel stecken. Er ist z.Zt. erst angekündigt (Einzelstückpreis um DM 132,-) und ist nicht kompatibel mit Intels 2732 (z.Zt. ca. DM 300,-, lieferbar).

EPROMs können leicht mit Hausmitteln gelöscht werden. UV-Lampen mit Glas- kolben sind allerdings unbrauchbar. Eine Höhensonne mit Quarzstrahler kann EPROMs in ca. 20 Minuten löschen, wenn sie 15-20 cm vom Strahler entfernt sind. Dabei ein Rat: Lieber größerer Abstand und längere Belichtung als geröstete (unschmackhafte) PROMs.

Zusammen mit leistungsfähigen Assemblern (z.B. der Macro-Assembler von C. Moser) und Monitoren (NEWTIM von R. Schineis) braucht Ihr CBM kein BASIC-Schattendasein zu führen, sondern wird zum leistungsfähigen Entwicklungssystem. Eine doppelseitige Platine zu diesem Prommer kann gegen geringe Kosten bezogen werden.

READY.

```

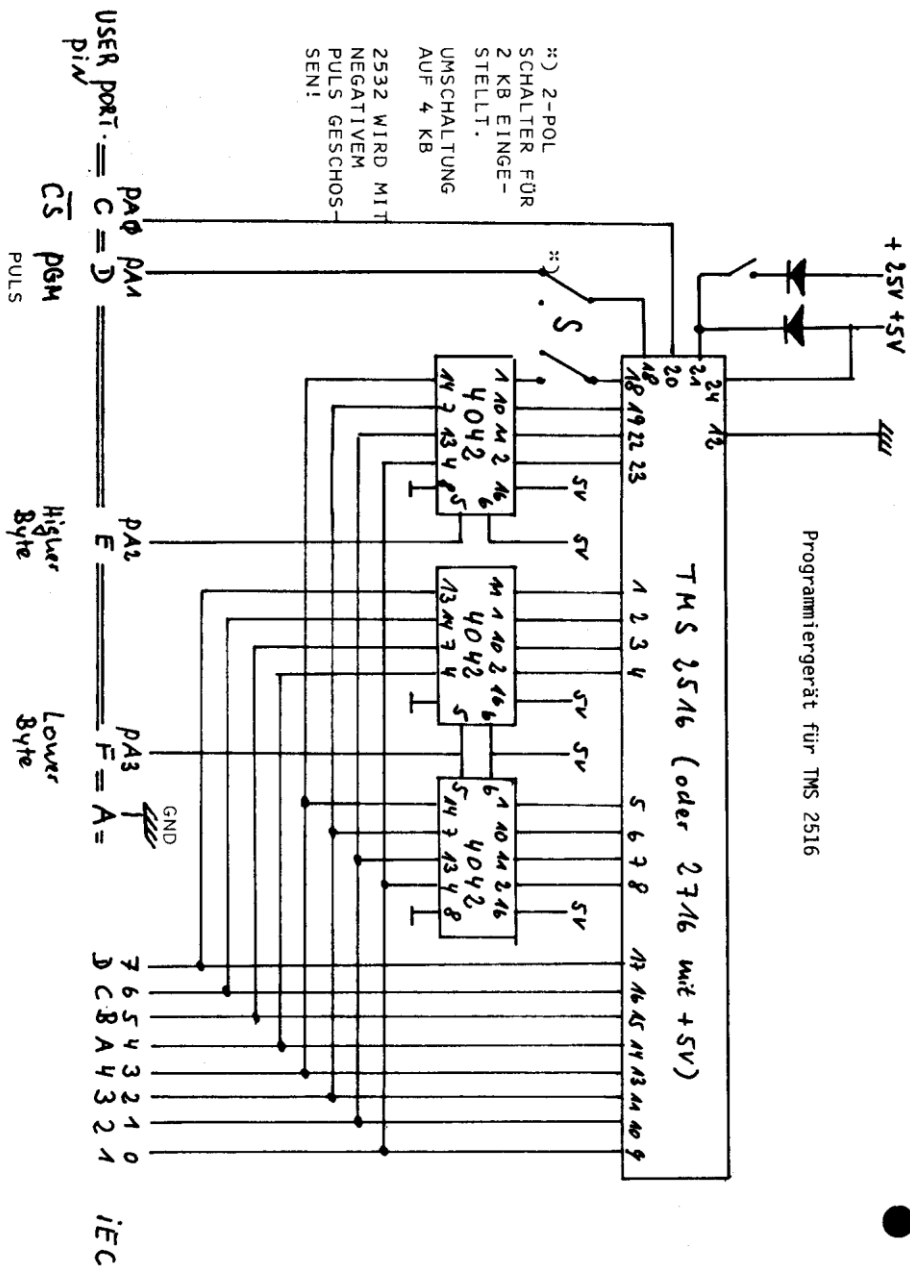
0 VER=0:INH=1:PRG=3:LOW=9:HIGH=5
1 FKT=59471:DOUT=59426:DIN=59424
2 TH=59465:REM TIMER ZUM MESSEN
3 POKE59459,255:REM** DATENRICHTUNGSREGISTER AUF AUSGANG
4 POKEFKT,INH :REM** INHIBIT
5 FORX=0TO2047
6 H=INT(X/256) :REM** HIGHER BYTE
7 L=X-H*256 :REM** LOWER BYTE
8 POKEDOUT,L:POKEFKT,LOW:POKEFKT,INH :REM*****LOWER BYTE
9 POKEDOUT,H:POKEFKT,HIGH:POKEFKT,INH:REM*****HIGHER BYTE
10 DATEN=PEEK(X+16384)
11 POKEDOUT,DATEN
12 POKETH,0:POKEFKT,PRG:REM
13 POKE 144,49
14 FORA=0TO30:NEXT :REM 50MS-PULS
15 POKE 144,46
16 POKE FKT,INH :REM
17 POKEDOUT,255:POKEFKT,VER:LESEN=PEEK(DIN):POKEFKT,INH
18 PRINTLESEN,DATEN,L,H:IF LESEN=DATEN THEN NEXT X:END
19 PRINT"PROGRAMMIERFEHLER":NEXTX:END

```

READY.

65_{xx} MICRO MAG

*) 2-POL
SCHALTER FÜR
2 KB EINGE-
STELLT.
UMSCHALTUNG
AUF 4 KB
2532 WIRD MIT
NEGATIVEM
PULS GESCHOS-
SEN!



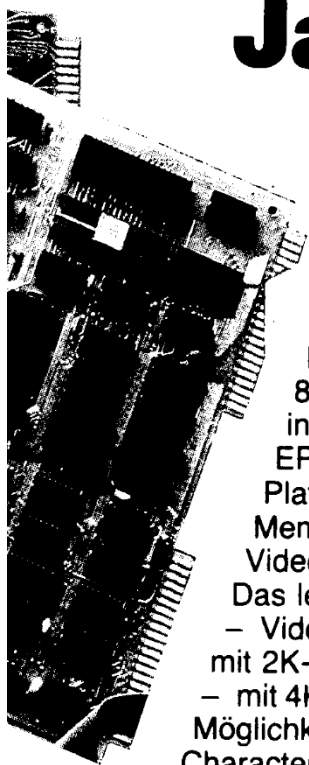


Bei Bestellung Pe

M.&R. Nedela
Tel. 07544 3575

MSB Mic
Sho
Boo

Leistung zum Jahresbeginn 1980



AIM (4K) im Gehäuse mit Netzteil
(Ausbau möglich) 1599,- DM
Systemerweiterung für AIM-65, SYM 1,
KIM-1 Speicher:

Memory Plus 755,- DM

8K-RAM frei adressierbar
in 4K- oder 8K-Schritten.

EPROM Programmierer (2716) und
Platz für 8K EPROM's

Memory Plus und ein 2716 EPROM 865,- DM
Video-Ausgabe:

Das leistungsfähigste Video Interface
– Video Plus

mit 2K-Bildwiederholtspeicher 755,- DM

– mit 4K-Bildwiederholtspeicher und Grafik

Möglichkeit in 2K-RAM-programmierbarem

Charactergenerator (alle Zeichen möglich) 875,- DM

EPROM mit Software für Video Plus 120,- DM

EPROM-Programmiergerät EP-ZA benötigt 8-bit-Port und

ein 4-bit-Port von AIM, KIM, für 2704/2708/2716/2732,

zwei Personality Module im Preis enthalten 708,- DM

Software in EPROM 145,- DM

/ Module angeben.

Die von vielen Betreibern des PET 2001 oder CBM 3001 seit langem gewünschte ROM-Vergleichsliste ist eine eigene Arbeit der Autoren, die unabhängig von einer ähnlichen Liste erfolgte, die von J. Butterfield und J. Russel veröffentlicht wurde.

Hans-Peter Görtz, Düsseldorf

Wolfgang Kölln, Delmenhorst

ROM-VERGLEICHSLISTE CBM 3001/PET 2001

CBM	PET	ERKLÄRUNG
CDEC	CE05	PRUEFT ARITHM. AUSDRUCK IN KLAMMERN
CD2	CE08	PRUEFT, OB RECHTE KLAMMER VORHANDEN
CD5	CE0E	PRUEFT, OB LINKE KLAMMER VORHANDEN
CD8	CE11-CE1B	PRUEFT, OB KOMMA VORHANDEN
CE03	CE1C-CE20	GIBT FEHLERMELDUNG 'SYNTAX ERROR'; VERLASST LAUFENDE ROUTINE
CE08	CE21-CE27	PRUEFT 'BASIC'-FUNKTION FÜR SPÄTERE AUSFÜHRUNG
CE0F	CE28-CE39	VORBEREITUNG ZUM SÜCHEN EINER VARIABLEN
CE2A	CE3B-CE96	UEBERPRUEFUNG DER RESERVIERTEN VARIABLEN TI, T\$ UND ST
CE89	CE97-CED5	UEBERPRUEFT ART DER FUNKTION
CEC8	CEDE-CF05	AUSFÜHRUNG DER 'AND'- UND 'OR'-FUNKTION
CEFB	CF06-CF6D	AUSFÜHRUNG VON VERGLEICHEN
CF60	CF6E-CF7A	VORBEREITUNG ZUR AUSFÜHRUNG DER 'DIM'-ANWEISUNG
CF6D	CF7B-D00E	SUCHT NACH EINER 'BASIC'-VARIABLEN
D001	D00F-D078	EINRICHTUNG EINER NEUEN VARIABLEN
D069	D079-D087	SPEICHERT ZEIGER AUF NEUE/GEFUNDENE VARIABLE
D078	D088-D098	HILFSROUTINE ZUM SÜCHEN IN EINEM FELD
D089	D099-D09C	FLIESSKOMMA-KONSTANTE 32768
D08D	D09D-D0B8	ZAHLENUMWANDLUNG VON 'REAL' IN 'INTEGER' MIT VORZEICHEN
D0AC	D0B9-D263	SÜCHEN ODER EINRICHTEN VON FELDERN
D259	D264-D277	AUSFÜHRUNG DER 'FRE'-FUNKTION
D26D	D278-D284	ZAHLENUMWANDLUNG VON 'INTEGER' IN 'REAL'
D27A	D285-D28A	AUSFÜHRUNG DER 'POS'-FUNKTION
D280	D28B-D294	AUSGABE VON 'ILLEGAL DIRECT ERROR', FALLS 'DIRECT MODE'
D28D	D295-D348	EINRICHTUNG UND UEBERPRUEFUNG EINER BENUTZERFUNKTION
D33F	D349-D36A	AUSFÜHRUNG DER 'STR\$'-FUNKTION
D361	D36B-D3D1	SUCHT NACH ELEMENTEN VON ZEICHENKETTEN
D3CE	D3D2-D403	ERRECHNET ZEIGER AUF ZEICHENKETTE
D400	D404-D5C3	'GARBAGE COLLECTION' - ENTFERNT NICHT BENÖTIGTE 'STRINGS'
D5C6	D5C4-D5D7	AUSFÜHRUNG DER 'CHR\$'-FUNKTION
D5DA	D5D8-D653	AUSFÜHRUNG DER 'LEFT\$-', 'RIGHT\$'- UND 'MID\$'-FUNKTION
D656	D654-D662	AUSFÜHRUNG VON 'LEN'
D665	D663-D672	AUSFÜHRUNG DER 'ASC'-FUNKTION
D675	D673-D684	LIEST ZAHLE (KLEINER 256) AUS 'BASIC'-TEXT
D687	D685-D6C3	AUSFÜHRUNG DER 'VAL'-FUNKTION
D6D6	D6C4-D6CF	LIEST ZWEI ZAHLEN AUS 'BASIC'-TEXT (16-BIT UND 8-BIT)
D6D2	D6D0-D6E5	PRUEFT, OB 0 < ZAHLE <= 65535
D6E8	D6E6-D701	AUSFÜHRUNG VON 'PEEK' UND 'POKE'
D710	D702-D71D	AUSFÜHRUNG DER 'WAIT'-ANWEISUNG
D72C	D71E-D890	AUSFÜHRUNG VON ADDITION UND SUBTRAKTION
D8C8	D891-D8BE	FESTKOMMA-KONSTANTEN
D8F6	D8BF-D8FC	AUSFÜHRUNG DER 'LOG'-FUNKTION
D934	D8FD-D95D	AUSFÜHRUNG DER MULTIPLIKATION
D998	D95E-D988	BELEGT FLIESSKOMMA-AKKUMULATOR #2
D9C3	D989-D9B3	KORREKTUR DER AKKUMULATOREN #1 UND #2
D9EE	D9B4-D9E0	MULTIPLIZIERT BZW. DIVIDIERT MIT BZW. DURCH 10
DA1B	D9E1-DA73	AUSFÜHRUNG DER DIVISION
DAAE	DA74-DA98	BELEGT AKKU #1
DAD3	DA99-DACD	BELEGT 'BASIC'-VARIABLE MIT AKKU #1
DB08	DACE-DADD	BELEGT AKKU #2 MIT AKKU #1
DB18	DADE-DAEC	BELEGT AKKU #1 MIT AKKU #2
DB27	DAED-DAFC	AKKU #1 WIRD AUFRUNDET
DB37	DAFD-DB29	AUSFÜHRUNG DER 'SGN'-FUNKTION
DB64	DB2A-DB2C	AUSFÜHRUNG DER 'ABS'-FUNKTION

65xx MICRO MAG

DB67	DB2D-DB6C	VERGLEICH 'BASIC'-VARIABLE MIT AKKU #1
DBA7	DB6D-DB9D	ZAHLENUMWANDLUNG VON 'REAL' IN 'INTEGER' OHNE VORZEICHEN
DBB8	DB9E-DBC4	AUSFUEHRUNG DER 'INT'-FUNKTION
DBFF	DBC5-DC4F	UMWANDLUNG EINER ZEICHENKETTE IN EINE ZAHL
DC8A	DC50-DC84	LIEST ASCII-ZEICHEN ('0'..'9') AUS 'BASIC'-TEXT
DCCE	DC94-DCBE	AUSGABE EINER ZEILENNUMMER
DCE9	DCAF-DDE2	UMWANDLUNG EINER ZAHL IN EINE ZEICHENKETTE (ABLADE AB \$0100)
DE1D	DDE3-DE23	UMWANDLUNGSKONSTANTEN
DE5E	DE24-DE2D	AUSFUEHRUNG VON 'SQRT'
DE68	DE2E-DE66	AUSFUEHRUNG DER POTENZFUNKTION
DEA1	DE67-DE71	VORZEICHENWECHSEL
DEA9	DEA0-DEF2	AUSFUEHRUNG DER 'EXP'-FUNKTION
DF2D	DEF3-DF3C	UEBERPRUEFUNG VON FOLGEN VON FUNKTIONEN
DF7F	DF45-DF9D	AUSFUEHRUNG DER 'RND'-FUNKTION
DFD8	DF9E	UEBERPRUEFUNG VON 'COS'
DFDF	DFA5-DEED	UEBERPRUEFUNG VON 'SIN'
E028	DFEE-E019	UEBERPRUEFUNG VON 'TAN'
E08C	E048-E077	UEBERPRUEFUNG VON 'ATN'
E0F9	E0B5-E0CC	'CHAR GET ROUTINE' - WIRD IN SEITE 0 TRANSFERRIERT
E116	E0D2-E173	ABSCHLUSS DER 'RESET'-ROUTINE, 'RAM'-PRUEFUNG USW.
	E19B-E1BB	GEHOERT ZU 'TI-', 'TI\$'-ROUTINE
	E1BC-E1E0	SELEKTOR FUEHRT 'INPUT', 'GET' UND 'READ'
E1DE	E1E1-E27C	INITIALISIERUNG DER 'I/O'-REGISTER
E285	E27D-E3C3	ERHAELT EINGABE VON BILDSCHIRM/TASTATUR
E3B4	E3C4-E3E9	EINRICHTUNG DER NAECHSTEN (NEUEN) ZEILE AUF DEM BILDSCHIRM
E3D8	E3EA-E52F	AUSGABE EINES ZEICHENS AUF DEM BILDSCHIRM
E519	E530-E5DA	BILDSCHIRMINHALT NACH 'OBEN' VERSCHIEBEN (FALLS NOETIG)
E527	E5DB-E66A	AUSGABE EINER NEUEN ZEILE
E61B	E66B-E67D	BEGINN DER 'INTERRUPT'-ROUTINE
E6E4	E67E-E683	BEENDIGUNG DER 'INTERRUPT'-ROUTINE
E6E2	E685-E73E	'INTERRUPT'-ROUTINE: 'CURSOR', TASTATUR-ABFRAGE, MOTOR USW.
E6F8	E73F-E7AB	UMWANDLUNG DER TASTATUR-MATRIX IN EIN ASCII-ZEICHEN
E6EA	E7AC-E7B9	AUSGABEROUTINE FUEHRT BILDSCHIRM
F156	E7DB-E7EB	AUSGABE EINER MELDUNG DES BETRIEBSSYSTEMS
F0B6	F0B6-F1CB	EROEFFNEN, PRUEFEN UND SCHLIESSEN EINES KANALS (IEEE-BUS)
F1D1	F1CC-F22F	EINGABE EINES ZEICHENS UEBER IEEE
F232	F230-F27C	AUSGABE EINES ZEICHENS UEBER IEEE
F272	F27D-F2A3	RUECKSETZEN DER 'I/O'-REGISTER
F26E	F2A4-F2AA	LOESCHT ALLE EROEFFNETEN 'FILES' (ACHTUNG: KEIN 'SCHLIESSEN')
F28D	F2AB-F2B7	SUCHT IN TABELLE DER EROEFFNETEN 'FILES'
F299	F2B8-F2C7	SENDET DATEI-PARAMETER ZUM PERIPHERIEGERAET
F2A9	F2C8-F329	SCHLIESST EINE DATEI
F301	F32A-F33E	PRUEFT, OB 'STOP'-TASTE GEDRUECKT WIRD
	F33F-F345	PRUEFUNG, OB 'DIRECT MODE'
F3C2	F346-F3FE	AUSFUEHRUNG DER 'LOAD'-ANWEISUNG
	F3FF-F421	AUSGABE 'SEARCHING'
	F422-F432	AUSGABE 'LOADING' BZW. 'VERIFYING'
	F433-F461	LIEST PARAMETER FUEHRT 'LOAD'/'SAVE'
	F462-F494	AUSFUEHRUNG VON 'LOAD'/'SAVE'/'OPEN' AUF IEEE-BUS
	F495-F4BA	SUCHT NACH BESTIMMTEN PROGRAMMNAMEN (AUF KASSETTE)
F4B7	F4BB-F4D3	AUSFUEHRUNG VON 'VERIFY'
	F4D4-F529	LIEST PARAMETER FUEHRT 'OPEN'/'CLOSE'
F521	F52A-F5AD	AUSFUEHRUNG VON 'OPEN'
F5A6	F5AE-F5E2	SUCHT NAECHSTEN PROGRAMMNAMEN (AUF KASSETTE)
	F5E3-F5EC	LOESCHT KASSETTENPUFFER
F5DA	F5ED-F64C	SCHREIBT KASSETTENVORSPEICH
F63C	F64D-F666	LIEST ANFANGS- UND ENDDRESSSE VON VORSPEICH
F656	F667-F67C	BESTIMMT PUFFER #1 BZW. #2 FUEHRT EINGABE/AUSGABE
F66C	F67D-F694	SETZT ZEIGER FUEHRT START- UND ENDDRESSSE DES PUFFERS
F684	F695-F69D	AUSFUEHRUNG VON 'SYS'
F69E	F69E-F71B	AUSFUEHRUNG VON 'SAVE'
	F71C-F735	BESTIMMT 'SECONDARY ADDRESS', FALLS NICHT ANGEZEIGT
F729	F736-F78A	BRINGT 24-STD-UHR AUF NEUESTEN STAND
F770	F78B-F7DB	SETZT GERAET FUEHRT EINGABE
F7BC	F7DC-F82C	SETZT GERAET FUEHRT AUSGABE
F806	F82D-F83A	INITIALISIERT ZAEHLER FUEHRT KASSETTENPUFFER
F812	F83B-F85D	WARTET, BIS 'PLAY' GEDRUECKT WIRD
F835	F85E-F870	PRUEFT 'TASTE-GEDRUECKT'-LEITUNG ZUM RECORDER
F847	F871-F87E	WARTET, BIS 'RECORD' UND 'PLAY' GEDRUECKT WIRD
F855	F87F-F8B8	LIEST VORSPEICH EINER DATEI VON KASSETTE
F886	F8B9-F8D1	SCHREIBT VORSPEICH EINER DATEI
F89E	F8D2-F912	BEENDET SCHREIBEN/LESEN

65xx MICRO MAG

F8E6	F913-F91D	WARTET AUF BEENDIGUNG EINGABE/AUSGABE
F8F0	F91E-F92D	PRUEFT, OB 'STOP'-TASTE GEDRUECKT; ABRUCH, FALLS NOETIG
F900	F92E-F95E	VORBEREITEN DER SYNCHRONISATION BEI LESEN VON KASSETTE
F931	F95F-FBFB	'INTERRUPT'-ROUTINE FÜR LESEN VON KASSETTE
	FBDC-FBE4	SPEICHERT ZEIGER AUF LETZTES GESCHRIEBENES ZEICHEN
	FBE5-FBE8	SETZT STATUSVARIABLE 'ST'
	FBE9-FBFF	HILFSROUTINE - ZAEHLT 8 BITS PRO BYTE
	FC00-FC1B	SCHREIBT EIN BIT AUF KASSETTE
	FC1C-FCFA	'INTERRUPT 1' BEI SCHREIBEN AUF KASSETTE (BEGINNT AB #FC21)
	FCFB-FD15	BEENDIGUNG VON EIN-/AUSGABE, RUECKSETZEN DER ZEIGER
	FD16-FD37	SETZT 'INTERRUPT'-VEKTOR
	FD38-FD47	'POWER-ON-RESET'-ROUTINE; PRUEFT OB 'DIAGNOSTIC'-MODUS
	FD48-FD7B	DIAGNOSTIC-ROUTINE
	FD7C-FD8F	'PARITY'-PRUEFUNG (QUERSUMME VON 8 BITS)
	FD90-FD9A	ZEIGERKORREKTUR
	FD9B-FFB1	FORTS. DIAGNOSTIC-ROUTINE

SPRUNGTABELLE

FFC0	OPEN
FFC3	CLOSE
FFC6	INPUT-DEVICE
FFC9	OUTPUT-DEVICE
FFCC	RESTORE I/O
FFCF	INPUT (ZEICHEN VON BILDSCHIRM)
FFD2	OUTPUT (ZEICHEN AUF BILDSCHIRM)
FFD5	LOAD
FFD8	SAVE
FFDB	VERIFY
FFDE	SYS
FFE1	STOP-KEY
FFE4	INPUT (ZEICHEN AUS TASTATURPUFFER)
FFE7	CLEAR (ALLE 'I/O'-KANAELE)
FFEA	24-HR-CLOCK
FFED-FFFA	TURN OFF CASSETTE MOTORS
FFFA-FFFF	NMI
FFFC-FFFF	RESET
FFFE-FFFF	IRQ

ABWEICHUNGEN PET-CBM

DF7F	DF45	NEU: 'RND(0)'
E054	E01A	FLIESSKOMMA-KONSTANTEN
E0BC	E078	FLIESSKOMMA-KONSTANTEN
E116	E0D2	NEUER ANFANGSWERT FÜR 'STACKPOINTER' (251)
E116	E0D2	ABWEICHUNGEN BEI INITIALISIERUNG
E1B7	E174	TEXT 'BYTES FREE, COMMODORE BASIC'
	E19B	SPEICHERT ZEILENNUMMER, AUSGABE 'READY.'
	E19F	TEIL VON 'TI'-ROUTINE (SIEHE CE2A/CE33)
	E1AB	DITO
	E1BC	TEIL VON 'INPUT', 'GET', 'READ' (SIEHE CB07/CB24)
	E1D9	DITO
E1DE	E1E1	ABWEICHUNG (INITIALISIERUNG)
E246	E236	BILDSCHIRM LOESCHEN
E229	E250	INITIALISIERUNG DER ZEIGER AUF BILDSCHIRMZEILEN
E257	E5DB	KORREKTUR DER ZEIGER AUF BILDSCHIRMZEILEN
E265	E27D	MELDUNG, DASS TASTATUR-PUFFER LEER
E29D	E297	EINGABE ZEICHEN, AUSGABE AUF SCHIRM, RUECKKEHR BEI 'RETURN'
E2F4	E2FA	EINGABE VOM BILDSCHIRM

MONITOR CBM

E76A	WROA
E775	WROB
E784	WRTWO
E78D	ASCII
E797	T2T2
E7A7	RDOA
E7B6	RDOB
E7E0	HEXIT
E7EB	RDOC
E7F7	ERROPR

Michael Zimmermann, Eberstädter Str. 170, 6102 Pfungstadt

ANSCHLUSS VON NUMERISCHER ANZEIGE UND TASTATUR AN EINEN 6500-MIKROCOMPUTER (1)

1. Einleitung

Sicher werden sich viele Leser fragen, warum am sich in der heutigen Zeit der Bildschirmprogrammierung und der intelligenten Terminals überhaupt noch mit so trivialen und vorgestrigen Dingen wie einfachen Siebensegmentanzeigen oder dekodierten Tastaturen beschäftigen kann. Sicher sind Bildschirm und Tastatur mit Editierfähigkeit Dinge, die erstrebenswert sind, für viele Anwendungen reichen aber einfachere Mittel für Ein- und Ausgabe. Man sollte sich hier als Designer nicht zum Maß aller Dinge machen und Fähigkeiten, die bei einem Entwicklungssystem nützlich sind, auch zur Richtschnur für Anwendungslösungen machen.

Wenn wir unsere Umwelt, soweit sie in der Datenverarbeitung ihren Niederschlag findet, betrachten, so ist die Vorherrschaft numerischer Informationen offensichtlich; von unserem Geburtsdatum über Kontonummer und -stand bis hin zu der Zahl der täglich mit dem Mikrocomputer verbrachten Stunden lassen sich die meisten Tatbestände als numerische Werte ausgeben. Deswegen ist es in vielen Fällen wenig sinnvoll, an der Schnittstelle dieser numerisch orientierten Welt zur Datenverarbeitung mit unpraktischen oder zu aufwendigen Hilfsmitteln für die Ein- und Ausgabe zu arbeiten. Aus diesem Grunde erscheinen auch heute noch Überlegungen zum Einsatz von numerischen Anzeigen und Tastaturen an Mikrocomputern durchaus angebracht.

Darüber hinaus gewinnt das Hobby Mikrocomputer gerade jetzt viele neue Freunde, die mit vergleichsweise komfortablen Maschinen arbeiten und die sich noch nicht mit diesen grundlegenden Techniken auseinandergesetzt haben, Techniken, die von Betreibern des KIM z.B. unbedingt beherrscht werden mußten.

Als Konsequenz dieser Überlegungen soll das Ziel unserer Betrachtungen eine numerische Anzeige - 8 Stellen sollten ausreichen - und eine Tastatur sein, wie sie z.B. in einem einfachen Datenverarbeitungsgerät Anwendung finden könnten. Alle dargestellten Schaltungen wurden vom Verfasser ausgetestet und betriebsfertig gemacht. Trotzdem erheben weder Schaltungen noch Programme Anspruch darauf, allgemeingültige Lösungen zu bieten, vielmehr können sie nur den Rahmen abstecken. Besonderes Augenmerk wurde auf die Darstellung der zugrunde liegenden Prinzipien gerichtet, auf besondere Tricks wurde deswegen verzichtet, und wo es sich ergab auf Erweiterungen hingewiesen. Darüber hinaus sollten die Leser diese Zeilen als Anregung begreifen und sich die Lösungen durch gezielte Versuche zu eigen machen oder den eigenen Anforderungen anpassen.

2. Überlegungen zur Hardware

Für den Anschluß von Anzeige und Tastatur werden seit einiger Zeit besondere Controller angeboten bzw. angekündigt. Für die 8080-Serie ist dies der 8279, der wegen allgemeiner Probleme beim Anschluß dieser Bausteinfamilie an die 6500 für unsere Betrachtungen keine größere Rolle spielen dürfte.

Aus der 6500-Familie ist der 6541 angekündigt, die Lieferungen dürften aber noch etwas auf sich warten lassen. Darüber hinaus scheint dieser Baustein für seine Aufgabe nicht besonders geeignet, da alle internen Register, auch der Display-Wiederholungspeicher, nur indirekt über ein Adreßregister ansprechbar sind, eine Technik, die bei einem CRT-Controller, der

meist nur einmal bei der Systeminitialisierung stärker benutzt wird, gangbar ist. Bei einem Display-Speicher, in den auch der Benutzer häufig Werte einstellt, ist dies eine große Behinderung und wenig anwendungsfreundlich.

Eine andere Lösung, die z.B. beim Elektor-Scamp-System benutzt wird, besteht darin, die Funktionen eines Display-Controllers mit Hardware nachzubilden. Hiergegen sprechen die 8 oder 9 Bauteile, die hierzu erforderlich werden. Von der Prozessorseite her erfolgt der einfachste Anschluß von Display und Tastatur über parallele Ports, wie sie z.B. im PIA realisiert sind. Hierbei ist es wenig sinnvoll, an einen Port jeweils eine Ziffernstelle zu schalten, sei es direkt oder über Dekoder, da hierdurch die Zahl der benötigten PIAs schnell steigen wird. Es ist eine bessere Lösung, Anzeige und Tastatur zu multiplexen. Leider ist es nicht möglich, die Anzeige direkt mit den Strömen und Spannungen der Ports zu treiben, da die Leuchtstärke zu gering würde, aber einfache 7404-Inverter können hier einen kostengünstigen Anschluß vermitteln. Als Anzeigeelemente werden hier Common-Cathode-Anzeigen vom Typ TIL 313 verwendet, die Tastaturen wurden aus Digitast-Schaltern aufgebaut.

3. Uncodierte Anzeige

Diese einfachste Form der Anzeige erfordert außer dem PIA einen minimalen Aufwand, die Ausgänge werden lediglich mit 7404 invertiert. Die Einzelheiten der Schaltung sind Abb. 1 zu entnehmen. Bei dieser Form der Anzeige ist jede einzelne Leuchtdiode als Element einer Matrix anzusehen, bei der durch den Port A die Zeile und durch B die Spalte bestimmt wird. Die Spalte repräsentiert hierbei die einzelnen Ziffern, die Zeile ein Leuchtelement über alle Ziffern hinweg. Durch Ansteuern einer speziellen Ziffer (Spalte) können die Segmente (Zeilen) in dieser Ziffer zum Leuchten gebracht werden.

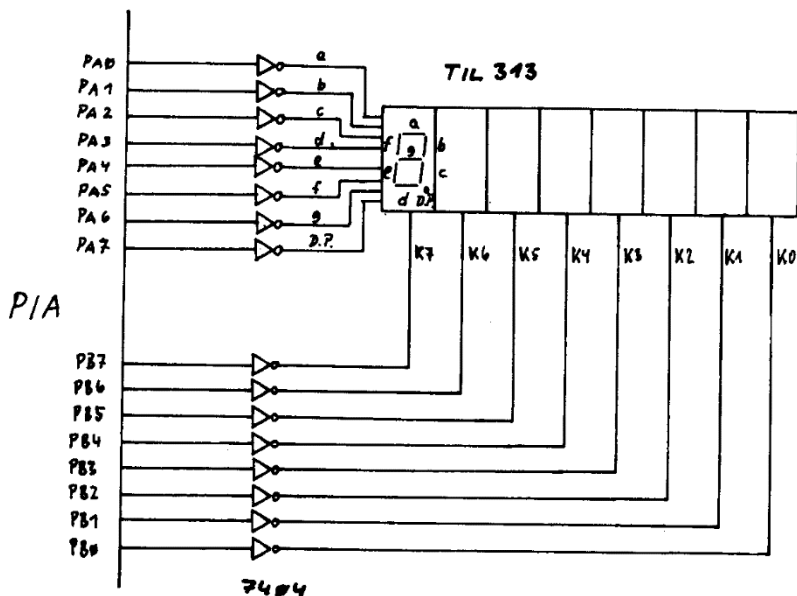


Abb. 1

65_{xx} MICRO MAG

Da es sich um Common-Cathode-Anzeigen handelt, erfolgt das Ansteuern durch Einschreiben eines Bits, das durch externe Schaltungen invertiert wird, in den B-Port. Hierdurch wird eine Ziffer angeschaltet, das eigentliche Leuchtmuster ergibt sich aus dem Komplement des Wertes, der in den A-Port gestellt wurde. Durch Verschieben des Bits durch den B-Port und gleichzeitiges Setzen der entsprechenden Leuchtmuster am A-Ausgang lassen sich nacheinander alle Ziffernstellen bedienen, wobei die jeweiligen Zustände eine zeitlang stabil gehalten werden müssen. Der Wechsel hat jedoch so schnell zu erfolgen, daß das Auge mit seiner Trägheit ihm nicht zu folgen vermag. Dann ist der Eindruck einer stehenden Anzeige gegeben. Einzelheiten der Bedienung sind dem kommentierten Programm 1 zu entnehmen. Dieses gliedert sich in 3 Hauptabschnitte:

1. Initialisieren des Ports
2. Übernehmen der anzuzeigenden Werte in den Ausgabepuffer, dabei Umwerten in Leuchtmuster
3. Anzeigen der Leuchtmuster aus dem Anzeigepuffer.

Nach jedem Durchlauf für das Display wird in die Zentralebene zurückverzweigt, um hier evtl. Eingaben abzufragen und zu bedienen.

Nachteil dieser vollständig uncodierten Anzeige ist es, daß ohne Uminitialisieren der PIA-Ports keine Eingaben bearbeitet werden können, da zumindest bei einer 8-stelligen Anzeige alle Ausgänge belegt sind. Wie hier mit geringem Aufwand Abhilfe zu schaffen ist, werden wir später sehen. Zuerst wollen wir uns aber der hier noch ausgeklammerten Erzeugung der Leuchtmuster widmen.

SIMPLE DISPLAY DRIVER

```
==0000 DRA
      =$9000
;DATA REGISTER A
==0000 DDRA
      =DRA
;DATA DIRECTION REGISTER A
==0000 CRA
      =DRA+1
;CONTROLLREGISTER A
==0000 DRB
      =DRA+2
;DATA REGISTER B
==0000 DDRB
      =DRB
;DATA DIRECTION REGISTER B
==0000 CRB
      =DRB+1
==0000 BUF
      =$00 DISPLAYBUFFER
==0000 PTR
      =$10 POINTER TO DISPLAY
```

MAIN PROGRAM

```
==0000
      *=$200
==0200
200F02 JSR INIT
;INITIALIZE PIA
202802 JSR GET
;GET STORAGE TO DISPLAY
```

```
==0206 M10
```

```
***
```

```
202902 JSR DISP
;DISPLAY BUFFER
204602 JSR KEY
;CHECK FOR KEYBOARD ENTRY
4C0602 JMP M10
```

INITIALIZE PIA

```
==020F INIT
```

```
***
```

```
A900 LDA #$00
;SET CRA/CRB TO DDRA/DDR B
8D0190 STA CRA
8D0390 STA CRB
A9FF LDA #$FF
;DIRECTION TO OUTPUT
8D0090 STA DDRA
8D0290 STA DDRB
==021F
```

```
A904 LDA #$04
;NOW ADDRESS DATAREGISTER
8D0190 STA CRA
8D0390 STA CRB
60 RTS
```

GET CHARACTERS TO BE DISPLAYED

```
==0228 GET
```

```
***
```

```
;SEE NEXT CHAPTER FOR DETAILS
```



```

60      RTS
-----
DISPLAY BUFFER
==0229 DISP
      ***
A207   LDX #07
;SET POINTER TO BUFFER
A901   LDA #01
;SET POINTER TO DISPLAY
8D0290 STA DRB
;AND ADDRESS DISPLAY
==0230 D20
      ***
B500   LDA BUF,X
;GET BUFFER CHARACTER
8D0090 STA DRA
;AND PUT INTO LEDS
A0FF   LDY #0FF
;NOW DELAY
==0237 D30

```

```

***
88      DEY
D0FD   BNE D30
CA      DEX
;NOW NEXT CHARACTER IN BUFFER
A9FF   LDA #0FF
;SIGN OFF LEDS
8D0090 STA DRA
0E0290 ASL DRB
;POSITION TO NEXT DISPLAY
90EB   BCC D20
;TO NEXT DISPLAY
60      RTS
; IF WE ARE THRU RETURN
-----
CHECK FOR KEYBOARD-ENTRY
==0246 KEY
      ***
;SEE FORTHCOMING CHAPTERS FOR DETAILS
60      RTS
      .END
ERRORS=0000

```

4. Erzeugung von Leuchtmustern auf der Anzeige

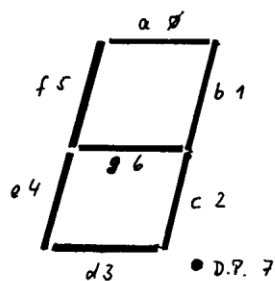
Beim vorstehenden Anzeigeprogramm wurde der Anzeigewert nur aus einem Puffer entnommen. Es fehlte die Angabe, wie das Leuchtmuster in diesen hineingestellt wurde. Grundsätzlich ist eine derartige Trennung in Erzeugung und Anzeige eines Leuchtmusters anzustreben, da in einem derartigen Vorgehen eine größere Flexibilität in der Bedienung der Anzeige durch den Benutzer begründet ist. Wir werden daher diese Trennung in den folgenden Programmen beibehalten.

Zuerst einige Anmerkungen zum darstellbaren Zeichenvorrat. Mit einer Siebensegmentanzeige kann man nicht den ganzen ASCII-Zeichensatz abbilden. Dies zeigen z.B. die Bemühungen von Texas Instruments, für alle Ziffern, Buchstaben und eine Reihe von Sonderzeichen Leuchtmuster zu definieren. Dies führt zwangsläufig zu nicht mehr erkennbaren Zeichenformen und damit zu Unklarheiten, so daß ein derartiges Alphabet nur mit einer Umwandlungstabelle lesbar wird.

In der Abb. 2 ist zuerst einmal eine genaue Definition der Siebensegmentanzeige mit eindeutiger Zuordnung der einzelnen Leuchtbalken gegeben. Weiterhin erscheint es erforderlich, für alle Hexadezimalzahlen Muster vorzusehen. Ein Vorschlag, der keine Rücksicht auf die Benutzung von Groß- oder Kleinbuchstaben nimmt, kann Abb. 3 entnommen werden. Die entsprechenden Verschlüsselungen werden hierbei mit angegeben. Ebenso sind in Abb. 4 sinnvoll darstellbare Groß- und Kleinbuchstaben mit ihren Leuchtmustern aufgeführt. Von der Verwendung weiterer Zeichen, insbesondere in der betrieblichen Praxis und damit in der Hand von ungeschultem Personal, kann der Verfasser nur abraten. In Fällen, in denen eine stärkere Bearbeitung von alphanumerischen Informationen erforderlich ist, sollte nicht aus falscher Sparsamkeit eine Siebensegmentanzeige mit ihrem hierfür unzureichenden Zeichenvorrat gewählt werden, diesem Anzeigentyp sollte vielmehr die Ausgabe rein numerischer Daten vorbehalten bleiben.

65_{xx} MICRO MAG

Abb. 4



Definition der Leuchtanzeige
und Bit-Position

Abb. 2

0	0	C0
1	1	F9
2	2	A4
3	3	B0
4	4	99
5	5	22
6	6	82
7	7	F8
8	8	80
9	9	93
A	A	88
B	b	83
C	C	C6
D	d	A1
E	E	86
F	F	8E

Abb. 3

Zeichen	groß		klein	
A	A	88		
B			b	83
C	C	C6	C	A7
D			d	A1
E	E	86		
F	F	8E		
G				
H	H	89	h	8B
I			i	7B
J	J	F1		
K				
L	L	C7		
M				
N			n	AB
O			o	A3
P	P	8C		
Q				
R			r	AF
S				
T				
U	U	C1	u	E3
V				
W				

65_{xx} MICRO MAG

Für die eigentliche Umsetzung von ASCII-Daten ist hier das Programm 2 abgedruckt, als Beispiel für die Wandlung gepackter Daten kann Programm 3 genommen werden. Beide Programme entnehmen die Werte dem Anwenderspeicher, auf den ein Pointer zeigt. Die eigentliche Umwandlung erfolgt über Tabellen, Einzelheiten können den Programmkommentaren entnommen werden.

X				
Y	51	91		
Z				

Abb. 4

PROGRAMM 2

GET CHARACTERS TO BE DISPLAYED

```

==0228 GET
      ***
A007 LDY #07
;LOAD COUNTER
==022A G10
      ***
B110 LDA (PTR),Y
;AND CHARACTER
293F AND #$3F
;MASK OFF BITS
AA TAX
;USE AS INDEX
BD3902 LDA TAB,X
;GET PATTERN
990000 STA BUF,Y
;AND STORE IN BUFFER
88 DEY
;DECREMENT COUNTER
10F2 BPL G10
;NOT DONE, REPEAT
60 RTS
;DONE, RETURN

```

PATTERN TABLE

==0239 TAB

```

.BYT $FF,$88,$83,$C6 ;a-g
.BYT $A1,$86,$8E,$FF
.BYT $89,$FB,$F1,$FF ;H-O
.BYT $C7,$FF,$AB,$A3
.BYT $8C,$FF,$AF,$FF ;P-W
.BYT $FF,$C1,$FF,$FF
.BYT $FF,$91,$FF,$FF ;X-<
.BYT $FF,$FF,$FF,$FF
.BYT $FF,$FF,$FF,$FF ;SP-
.BYT $FF,$FF,$FF,$FF
.BYT $FF,$FF,$FF,$FF ;(-/
.BYT $FF,$FF,$FF,$FF
.BYT $C0,$F9,$A4,$B0 ;0-7
.BYT $99,$92,$82,$F8
.BYT $80,$98,$FF,$FF
.BYT $FF,$FF,$FF,$FF

```

ERRORS=0000

Wird ausführlich fortgesetzt

FIRMENNACHRICHTEN

Die Firma GWK, Gesellschaft für technische Elektronik mbH in Herzogenrath rundet ihr Angebot für 65xx-Betreiber ab: Der Mikrocomputer Siemens PC 100 wurde neu in das Programm aufgenommen, ebenso der Video-Monitor Sanyo DM5912CX und der Matrixdrucker TX80 mit 95 ASCII- und 64 Grafikzeichen und speziell auf AIM 65/PC 100 zugeschnittenem Betriebssystem - AIMS werden jetzt mit garantierter und protokollierter 'burn in' in vom Besteller gewünschter Dauer geliefert. Hinzu kommen neben der bereits bekannten BASIC-ERweiterung für den AIM weitere Softwarehilfen, die 'Development Aid' mit Systemprogrammen wie MOVE, RELOCATE, Disassemble into Texteditor, FILL, Search Byte, Word, String usw. und ein BASIC-Texteditor mit beliebiger Zeilenlänge. Außer auf der Hobby-tronic 80 in Dortmund werden die Produkte auch auf der Hannover-Messe in Halle 12, 2. OG. Stand 21/22 vorgeführt.

Christian Streicher, Wittelsbacherstr. 24, 8034 Germering

U S C O M - USER DEFINED BASIC COMMANDS (AIM 65)

Mit diesem Programm ist die Möglichkeit gegeben, zusätzliche vom Anwender definierte BASIC-Kommandos abarbeiten zu lassen. Das Hauptproblem bestand darin, eine Stelle zu finden, an der der BASIC-Interpreter unterbrochen werden kann, ohne den natürlichen Programmablauf zu gefährden. Eine Veränderung des DILINK-Vektors stellte sich als unbrauchbar heraus, da hierdurch bestenfalls die SN-ERROR-Meldung unterdrückt werden konnte, nicht jedoch der Programmablauf. Nach längerem Suchen im Bereich der Zero Page fand der Autor zwischen hex BF und D6 eine Routine, die bei der Initialisierung des AIM-BASIC angelegt wird und die dem Interpreter dazu dient, den ':' als Delimiter für den nächsten Befehl zu erkennen und außerdem eine Zeilennummer von einem direkt eingegebenen Befehl zu unterscheiden. Innerhalb dieser Routine werden nun die Adressen C6 und C7 ständig verändert. Sie stellen den Vektor auf den abzuarbeitenden Befehl im BASIC-Input Buffer dar. In Zelle CA beginnt der Befehl BCS ØØD6. Der Sprung erfolgt, wenn der Befehl aus dem Input Buffer ein BASIC-Befehl oder ein Alphazeichen ist. Nachdem in Zelle D6 lediglich ein RTS steht, ist hier die Möglichkeit gegeben, in den Programmablauf einzugreifen. Die Adresse des Branch-Befehles wird zu diesem Zweck auf die Adresse des ATN-Vektors in Zelle BB gesetzt. Der ATN-Vektor wird auf die Anfangsadresse der Steuerprogramme gesetzt. Das Steuerprogramm entscheidet nun, ob ein vom Benutzer definierter Befehl vorliegt oder nicht. Wird der Befehl unter 'COMM' nicht gefunden, so erfolgt die SN-ERROR-Meldung. Anderenfalls wird ein REM (hex 8E) an den Interpreter übergeben um die Fehlermeldung zu umgehen. Nach dem Lesen des letzten Zeichens wird der Befehl ausgeführt.

Auf eine Einschränkung sollte jedoch an dieser Stelle hingewiesen werden: Da der Input Buffer nach der vollständigen Abarbeitung wieder in den BASIC-Textspeicher zurückkopiert wird, können die Befehle als Steuerbefehle, wie z.B. RESequenz oder RENumber verwendet werden, ausgenommen die ATN-Funktion. Eine Abspeicherung unter einer BASIC-Zeilenummer ist nicht möglich, da der Befehl nach dem ersten Lauf mit REM überschrieben ist. Außerdem sollte beachtet werden, daß vom Userkommando nur die ersten drei Zeichen interpretiert werden.

```

*=$CA          BCS EXTRA          SN-ERROR AUF ATN-VEKTOR
*=$BB
EXTRA JMP ANF   ATN VEKTOR AUF STARTADRESSE SETZEN
*=$1F00
;
;VARIABLEN
;
GOTO *$+2
SAVY *$+1
SAVX *$+1
SAVA *$+1
LASTX .BYT 0
V1 *$+1
V2 *$+1
;
; PROGRAMMANFANG
;

```

ANF	PHP	
	STA SAVA	REGISTERINHALTE RETTEN
	STX SAVX	
	STY SAVY	
	LDX LASTX	
	BEQ LOOP1	GÜLTIGER BEFEHL SCHON GEFUNDEN?
	INX	
	STX LASTX	
	CPX #2	
	BNE END	BUCHSTABE 2 IM INPUT BUFFER
	LDX #0	
	STX LASTX	PROGRAMMSTATUS LÖSCHEN
7	PLP	
LOOP1	JMP (GOTO)	
	CMP COMM,X	USERBEFEHL VEREINBART?
	BEQ NEXT1	DELIMITER FÜR LETZTEN USER-BEFEHL
	LDA #\$7F	PRUEFE DIE NÄCHSTEN 4 ZEICHEN
	LDY #3	
LOOP2	INX	
	CMP COMM,X	
	BEQ ERR	USER-BEFEHL NICHT GEFUNDEN
	DEY	
	BPL LOOP2	
	LDA SAVA	
	JMP LOOP1	NÄCHSTEN USER-BEFEHL TESTEN
NEXT1	STX V1	
	LDY #2	
	LDA (\$C6),Y	3. BYTE AUS DEM INPUT BUFFER RETTEN
	STA V2	
	DEY	
	LDA (\$C6),Y	
	INX	BYTE 2 MITEINANDER VERGLEICHEN
	CMP COMM,X	
	BEQ NEXT2	
	LDA SAVA	
	INX	WENN UNGLEICH: TESTE NÄCHSTEN USER-BEFEHL
	INX	
	INX	
	JMP LOOP1	
NEXT2	INX	
	LDA V2	BYTE 3 MITEINANDER VERGLEICHEN
	CMP COMM,X	
	BEQ GOOD	
	LDA SAVA	
	INX	WENN UNGLEICH: TESTE NÄCHSTEN USER-BEFEHL
	INX	
	JMP LOOP1	
ERR	LDX #0	
	STX LASTX	BEFEHL NICHT GEFUNDEN
END	LDA SAVA	REGISTER WIEDERHERSTELLEN
	LDX SAVX	
	LDY SAVY	
	PLP	
	RTS	
GOOD	LDA V1	
	CLC	USER-ADRESSE BERECHNEN
	LSR A	

65xx MICRO MAG

```

TAX                USER ADRESSE RETTEN
LDA USADD,X
STA GQTO
LDA USADD+1,X
STA GOTO+1
LDY #0             REM-ANWEISUNG AN BASIC-INTERPRETER
LDA #$8E
STA SAVA
STA ($C6),Y
LDA #1             USER-KOMMANDO GÜLTIG
STA LSTX
JMP END            USER DEFINED COMMANDS

COMM .BYT 'ATN','XXX',$7F
                USER-ADRESSEN
USADD .WOR ATN,XXX
;
ATN   RTS        ZU ERPROBUNGSZWECKEN NUR 1 BEFEHL IN DEN
XXX   RTS        USER COMMANDS

EXTRA 00BB      V2      1F07      END      1F72
GOTO 1F00      ANF      1F08      GOOD     1F7D
SAVVY 1F02      LOOP1   1F28      COMM     1FA0
SAVX  1F03      LOOP2   1F31      USADD    1FA9
SAVA  1F04      NEXT1   1F40      ATN      1FAD
LASTX 1F05      NEXT2   1F5C      XXX      1FAE
V1    1F06      ERR     1F6D

```

Hans-Joachim Regge, Fesenfeld 57, 2800 Bremen 1

ERWEITERUNG DES AIM 65 AUF DEN S100-BUS

Bei der Vielzahl der für den AIM angebotenen Erweiterungssysteme fällt die Entscheidung nicht leicht: Die Inkompatibilität der Systeme untereinander bindet den Kunden evtl. an einen einzigen kleinen Hersteller mit nur wenigen Platinen zu teilweise hohen Preisen. Schafft sich der Käufer evtl. später ein größeres System an, sind viele Bausteine nicht mehr einsetzbar.

Eine Alternative ist der S100-Bus. Diese Karten sind über preiswerte Interfacekarten an die meisten bekannten Geräte anschließbar (z.B. TRS 80, APPLE, PET, AIM, SYM, KIM usw.) und werden von einer großen Zahl von Herstellern nicht nur für Hobbyzwecke angeboten, so daß wohl auch in Zukunft kein Lieferengpaß zu erwarten ist. Wo Konkurrenz existiert, freut sich der Kunde: So kostet z.B. eine komplette Karte (Kit) für 16 EPROMs Typ 2708 oder 2716 ohne EPROMs nur 50 Dollar in den USA. Auch RAM-Karten sind erfreulich preiswert. US-Firmen liefern prompt gegen Verrechnungsscheck, der deutsche Zoll inkl. MWSt beträgt ca. 24 %. Garantieansprüche sind allerdings wohl nur umständlich durchzusetzen.

Als Mutterplatine verwendet der Verfasser die ursprünglich für den KIM konzipierte 'KIMSI'-Karte. Diese ist mit wenig Aufwand (anderer Adressbereich, R/W) durch Einsetzen und Verdrahten eines 7402 in die vorbereitete 'Prototyp-Fass' abzuändern und dekodiert dann 1000-7FFF. Wem dieser Adreßraum

Bernhard Kokula, Wredestraße 17, 6700 Ludwigshafen

DAS VIDEO PLUS - EINE BESCHREIBUNG

Das VIDEO PLUS ist eine Video-Controller-Karte in professioneller Ausführung von der 'The Computerist Inc.' (USA). Abmessungen 275x200 mm, zwei Anschlußleisten für KIM-Stecker (44-polig). Sie paßt auf das von der gleichen Firma für den AIM angebotene Motherboard. Alle ICs sind auf Stecksockeln. Stromversorgung 5 Volt, 1,2 A oder 8 V, ungeregelt.

Folgende Baugruppen sind vorhanden: Bild-RAM mit 2114, 2 kB bestückt, bis 4 kB erweiterbar, Grafik-RAM 1 kB bestückt (bis 2 kB). EPROM-Sockel für 2716 (5 Volt), EPROM als Option, CRT-Controllersystem mit Motorola 6845, Tastatur- und Lichtgriffelsystem mit 6522-VIA (16+4 I/O, 2 Timer). Es werden somit 8 kB Adressenbereich belegt, dieser kann mit einem Drehschalter in 8-k-Schritten gewählt werden. Das EPROM-Programm ist für \$9829-9FFF geschrieben, der Gesamtbereich ist damit normal \$8000-9FFF.

Alle wichtigen Verbindungen zur CPU, besonders der Datenbus, sind gepuffert. Die Unterbringung dürfte keine Schwierigkeiten bereiten (Gehäuse von Feltron oder vom Computerist). Durch die Pufferung ist der direkte Betrieb am AIM ohne Motherboard möglich.

Baugruppe Bild-RAM

Sie ist mit 2114-RAMs/450 ns bestückt und dient während der Taktphase Ø1 (CPU) zum Schreiben und Lesen. Das RAM kann auch im Video-Betrieb wie ein normaler Programm- oder Datenspeicher benutzt werden. Während des CPU-Taktes Ø2 wird das RAM vom CRT-Controller verwaltet und ausgelesen. Das Video-RAM liegt relativ zur schalterbestimmten Dekodierung auf den Adressen 0000-0FFF. Bei Lieferung ist der Bereich 0000-07FF bestückt.

Baugruppe Graphik-RAM

Auch hier werden 2114-RAMs verwendet. Bestückung bei Lieferung 1 kB. Je 16 Byte bilden ein Grafik-Feld von 8 Punkten breit und 16 Zeilen hoch. Bei 128 ASCII-Zeichen sind das genau 2 kB. Jedes Grafik-Feld kann wieder von der CPU (Ø1) oder vom Controller (Ø2) angesprochen werden. Die Zeichen können völlig frei gestaltet werden. Sie sind leicht zu adressieren: OFFSET (Z.B. \$8000) +1000 RELATIV-LAGE +10xASCII-WERT. - Ein Beispiel: GROSSES 'A'=ASCII \$41. DARAUS WIRD GEBILDET \$800+\$1000+\$410=\$9410 FÜR DIE OBERSTE REIHE, ODER +\$41F=\$941F FÜR DIE UNTERSTE REIHE.

Wegen des direkten Zugriffes auf das Grafik-RAM können die einmal entwickelten Zeichen auf Kassette gespeichert werden. Per Programm gelangt man automatisch zum Grafik-RAM, indem man zum ASCII-Wert des Zeichens durch ORA #80 hex 80 addiert. Das bei Lieferung enthaltene RAM von 1 kB erlaubt die Darstellung von 64 Zeichen, es wird mit relativ 1000-13FF dekodiert. Man kann aber auch normale Programme in das Grafik-RAM schreiben.

Baugruppe EPROM

Es ist ein Stecksockel für ein Intel 2716-Eprom vorhanden (zum Lesen sind nur 5 V erforderlich). Der relative Adreßbereich ist 1820-1FFF (hex). Die ersten 32 Zellen (dezimal) müssen mit FF gelöscht bleiben, denn dort liegen die Adressen des CRT-Controllers sowie des VIA 6522. Als Option kann das Betriebsprogramm für das Video Plus in der für den AIM geeigneten Version als EPROM bezogen werden. Man kann sich den Festwertspeicher auch leicht selbst 'schießen', z.B. mit dem Programm EPROMMER, das in CHIP veröffentlicht wurde und das von Rockwell stammt. Es arbeitet am User-Port des AIM und benötigt außer der EPROM-Fassung nur einen Kipp-Umschalter und eine Diode (Programmieren +25V, Lesen +5V).

Leicht läßt sich aber auch mit einem 24-poligen DIP-Stecker ein selbstgestricktes 2 kRAM anschließen, da die Adreßdekodierung ja schon vorhanden ist. Nur die R/W-Leitung muß man sich noch vom E/A-Stecker am Kontakt E-W holen. Das geht hervorragend, man kann dann hier auch normale Programme laden.

Baugruppe CRT-Controllersystem

Außer dem 6845 sind zahlreiche ICs für die Teilung der Quarzfrequenz (32 MHz) auf 0,5 MHz vorhanden. Die Bildpunktfrequenz kann für Monitore mit geringerer Bandbreite von original 16 MHz auf 8 oder 4 MHz per Lötbrücke umgeschaltet werden. Der 6845 steuert die Datenbus-Puffer, damit die gemeinsame Benutzung von Bild- und Grafik-RAM funktioniert. Während für die CPU Bild- und Grafik-RAM gleichwertig sind, steuert Bit 7 im Bild-RAM die Auswahl des Zeichengenerators (Bit 7=0: Zeichenerzeugung aus ROM des character generator, =1: aus Grafik-RAM). Eine eigene Logik mischt nun noch die Bildimpulse zusammen, nämlich für den Bildschirm den VSYNC und den HSYNC, den Cursor-Impuls und die Video Data. Alles zusammen bildet das BAS-Signal im CCIR-Standard (1 Veff, 60 Ohm). Ein Trimpoti erlaubt die Anpassung an das verwendete Display.

Per Initialisierung in Software erzeugt der CRT 6845 nach den Parametervorgaben nun die Zeichenzahl je Zeile, die Linienzahl je Zeile, die Zeilenzahl pro Bild, die Lage des Bildes im Bild-RAM, die Stellung des Cursors und die Art des Cursors. Ganz frei ist man in der Festlegung des Bildes nicht, die Bildfrequenz 50 Hz muß am Ende etwa erreicht sein. In der Grundsoftware (EPROM/Band) ist das berücksichtigt. Hinsichtlich der Bildschirmzeilen je Zeichenzeile: 12 sind gut für ASCII-Zeichen, 16 sind für Grafik nötig. Aus einem Video-RAM von 4 kB können je nach den Parametern 3-8 Bilder erzeugt werden.

Baugruppe VIA 6522

Die auf der Platine enthaltene VIA ist mit Ihren 16 Portpins und 4 Kontrollleitungen auf die Steckerleiste 'A' herausgeführt. Die Anschlüsse und die Timer sind frei verfügbar und sind für die Abfrage einer Tastatur im Multiplex und eines Lichtgriffels vorgesehen. Weiterhin ist der Platz für eine eigene CPU 6502 vorgesehen. Damit kann das VIDEO-PLUS zu einem unabhängigen intelligenten Terminal ausgebaut werden. Das sind allerdings Optionen für den OEM. Die Beschreibung sagt darüber nicht allzuviel, das Listing enthält allerdings Vorschläge.

Software

Zum VIDEO PLUS gibt es Software in verschiedenen Versionen. Sie ist auf jeden Fall nötig, ohne Programm geht gar nichts, weil der 6845 per Software initialisiert werden muß. Mit der Lieferung erhält der Anwender eine AIM-Cassette. Das ist die billigste Lösung, die allerdings RAM benötigt, meistens das Grafik-RAM. Gegen Mehrpreis erhält man das 2716-EPROM und hat damit nach dem Einschalten sofort folgende Kontroll-Befehle: Cursor Home/Ende, Cursor links/rechts, auf/ab, Schirm-Scroll auf/ab, lösche Zeile, lösche alles, fülle Zeile, fülle alles (beliebiges Zeichen), Zeichen ASCII/Grafik. Mit der Lieferung erhält man ferner ein Listing (für AIM, inkl. Grafik). Mit diesen kommentierten Listings kann man Betriebsprogramme selber entwickeln und hat damit die reizvollsten und komfortabelsten Möglichkeiten.

Die RAM-Version auf Band ist nicht so reichhaltig wie die EPROM-Version. Allerdings macht der AIM-Monitor bei einigen Funktionen Schwierigkeiten, da er immer Rückfragen stellt, wenn er ein Zeichen nicht kennt. Und diese Rückfragen kommen natürlich auch auf das Bild. Erste Abhilfe ist: Eingabe 'L' (LOAD des Monitors), dann 1 SPACE, danach kann man beliebig lange Zeichenkette geben (ca. 54). Der Editor nimmt natürlich auch nichts krumm, BASIC ist da schon empfindlicher, Kontrollzeichen will es nicht in den

65.xx MICRO MAG

Speicher übernehmen. Alle Softwareversionen opfern der DELETE-Funktion das Display, d.h. es läuft nicht parallel mit. Herr Regge, Bremen fand allerdings eine Möglichkeit, das Display auf der 'CURRENTLINE' in allen Betriebsebenen (Maschine, Assembler, Editor, BASIC) mitlaufen zu lassen. Dafür geht das DELETE nur bis zum 20. Zeichen einer Zeile im Bild, auf den Speicher bleibt es auch darüberhinaus wirksam.

Groß-/Kleinschreibung kann man sich leicht selbst erzeugen, im Bereich der Buchstaben (\$41-5A) ist mit ORA #\$20 der ASCII-Wert zu erhöhen. Die Steuerung mit der Shift-Taste des AIM wird noch untersucht. Um Vorschläge hierzu wird gebeten.

Softwarefunktionen

Die Software aller Versionen besteht aus mehreren 'Baugruppen':

Initialisierung des VIDEO PLUS. Die Parameter zur Bild- und Cursorbestimmung werden aus einer oder aus mehreren Tabellen in die Register des 6845 geladen, Aufruf z.B. mit Taste F3.

Initialisierung des AIM. Das DILINK des AIM (A406/07) wird auf die Schreibroutine des VIDEO PLUS gerichtet.

Kontrollroutine. Alle Zeichen <\$20 sind für die Steuerbefehle vorbehalten und werden in der Kontrollroutine ausgeführt, gelangen aber nicht zur Anzeige.

Schreibroutine. Sie bringt die Zeichen ≥\$20 zur Anzeige auf den Schirm und bedient sich der Kontrollroutine für Weiterschaltung des Cursors, Zeilenwechsel, Scroll.

De-Initialisierung des AIM. Zurückschalten des DILINK-Vektors auf OUTDIS (EF05), z.B. mit Taste 'CTRL-F3' (\$1E).

Zusammenfassung

Professionelle Auslegung mit viel Ausbaumöglichkeiten bis hin zum stand-alone-Terminal. Sowohl für Hobbyschirme (32x16) als auch für hochwertige (80x24 Zeichen) geeignet. Zusätzliche Ports und Timer nach Art des AIM 65. Die Bus-Schnittstelle ist gepuffert, ein Betrieb ist direkt am AIM möglich. Viel RAM, durch Adreßbereich-Schalter als 'NOT-RAM' verwendbar. Viele Möglichkeiten der Softwaregestaltung. Nur eine Betriebsspannung (+5 V) Spannungsregler 'on Board'. Eigene Tastatur und Lichtgriffel vorbereitet. Eine ganze Menge für's Geld (Preis ca. DM 780,-).

Anzeige

ACHTUNG PET-/ CBM-ANWENDER

GRAFIKZUSATZ FÜR PET / CBM TISCHRECHNER

ENORME ERWEITERUNG DER AUSGABEMÖGLICHKEITEN

DES BILDSCHIRMS, 64000 EINZELPUNKTE,

ZUSÄTZLICHE BASIC-BEFEHLE.

MICRO-COMPUTER-VERTRIEB, STOCKBREITE 9,

3501 FULDATAL 1, TEL. 0561 - 81 11 66

SUPERBOARD CHALLENGER II

Ein kleines BASIC-System, das viel von sich reden machte, betriebsfertig mit ASCII-Tastatur, BASIC-Interpreter in 8 kROM, 4 k RAM, Videosignalerzeugung und Cassetteninterface zum freundlichen Preis von ca. DM 850,-.

Das System benötigt nur eine stabilisierte Spannung von 5 Volt bei allerdings 3 A mind.. Es kann an einen Monitor oder über einen vorzuhaltenden HF-Modulator an ein Fernsehgerät angeschlossen werden. Die Platine enthält Steckplätze für bis zu 8 k RAM. Das Superboard hat sich in den vergangenen Monaten wegen dieser Ausstattung nun auch hier einen zunehmenden Freundeskreis geschaffen, z.T. bei Anfängern, aber auch bei professionellen Anwendern.

Nach der Erprobung sind folgende Beobachtungen mitzuteilen: Das TV-Bild entspricht amerikanischer Norm, man kann es mit den meisten Heimfernsehern einfangen, allerdings nicht immer ganz ruhig (am besten auf UHF). Eine Monitoreinspeisung führt zu einem gestochenen scharfen und ruhigen Bild von 24 Zeilen mit 24 Zeichen. Es handelt sich also um eine großformatige Schrift, die immer nur ein kleines Fenster zum Speicher öffnet. Zusätzlich kann eine große Zahl von grafischen Symbolen erzeugt werden, die sich vor allem für Spiele verwenden lassen (Schiffe, Männchen etc.). Der Bildschirm wird angenehm schnell gefüllt.

Viel Leistung also in der Grundausstattung. Der Preis zog aber sicher auch einige Einschränkungen nach sich: Das BASIC (Microsoft) ist eine abgemagerte Version, es kennt z.B. keinen GET-Befehl, ein SAVE oder LOAD mit FILENAME (Cassette) ist nicht möglich. Man muß also eine sorgfältige Dateiverwaltung z.B. mit einem Umdrehungszählwerk oder Benennung eines Programmes im ersten Statement vornehmen. Das Bandformat ist langsam und nicht immer störungsfrei (kein Timereinsatz) und hat auch eigenwillige Meldungen. - Fehler werden durch grafische Symbole für die Fehlerart angezeigt - eine Frage der Gewöhnung.

Die Tastatur ist nicht ganz konventionell belegt, die Shift-Tasten an den beiden Seiten wirken verschieden, was man ebenso beachten muß wie die für BASIC einzurastende Taste 'Shift Lock'.

Die Platine trägt kaum Beschriftungen. Da die beigelegte Dokumentation wenig augenfreundlich gedruckt und gezeichnet ist, hat man vor der ersten Inbetriebnahme (die dann auf Anhieb klappt) lieber einige sorgfältige Prüfungen für die externe Beschaltung vorzunehmen.

Das Monitorprogramm ist 'mini'. Es erlaubt die Anzeige von Speicherzelleninhalten und deren Veränderung. Systembetreiber, die weiter in die Maschinensprache eindringen wollen, müssen sich Dienstleistungsprogramme für das Programmieren noch weitgehend selber aneignen, was nach heutigen Standards zwar sehr lehrreich aber auch sehr mühsam ist. Es scheint jedoch, daß sich hier und da schon Betreiberkreise bilden. Handbücher in Deutsch sind angekündigt.

In der Sicht des Autors, der der Firma Heninger Micro Computer GmbH in München, Landwehrstr. 39 für die Hergabe des Gerätes für die Besprechung dankt: Die kleinen Systeme können nicht alle Leistungen enthalten, über die man in der Computerei spricht, immer sind einige Akzente besonders gesetzt. Dieses ist zunächst vorwiegend auf BASIC orientiert und hat eine beachtliche Grundausstattung mit einigen Einschränkungen. Es wird seine Verdienste sicher bei Anfängern finden, die ohne viel Umstände zunächst einmal in die Computerei und BASIC hineinfinden wollen. Viele wird der Appetit beim Essen packen.

R.L.

ENGLISH SUMMARY

MLIST resembles the AIM 'M' command but displays the contents of 16 memory cells at a time in an up and down mode by blocks. - The DISC UTILITY PROGRAM allows to reformat, rename, read for test, copy CBM floppies and to append programs. - The 6502 DIRECT ASSEMBLER is a (final?)

version for PET which won't hang up on entry errors and introduces print out among other features. - SCHAUFEL is a BASIC program to relocate machine language programs. - VARLIST serves testing in BASIC and displays the actual contents of all variables at any desired time. - CBM SCHIESST SICH ELGENE EPROMS is a prommer for 2716 and 2732 with a minimum of hardware, driven by a tested BASIC program. - ROM VERGLEICHSLISTE PET/CBM names machine language entry points into the executing routines of monitor and BASIC interpreter. -

ANSCHLUSS VON NUMERISCHER

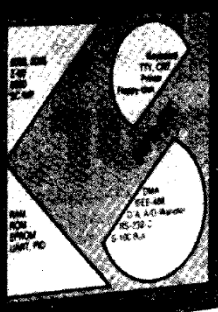
ANZEIGE UND TASTATUR shows in a series in hard- and software the several possibilities to interface LED displays and keyboards to a 65xx micro-computer. USCOM introduces user defined commands to AIM BASIC by manipulating the character get routine in the zero page. - Final articles describe the KIMS1 expansion board, the VIDEO PLUS from the Computerist and OHIO's Superboard Challenger II P.

THEMEN IN DEN FOLGENDEN HEFTEN:

Fortsetzungen für 'Anschluß von numerischer Anzeige und Tastatur' und für 'Leitfaden für die Programmierung'. Datenverbund zwischen PET/CBM und mehreren AIM 65 als Terminal mit Korrespondenz über die BASIC-Variablen. - Interruptgetriebene Cassettenein- und ausgabe. - Single stroke BASIC commands am PET. - Sortierte Symboltafel und Cross Reference List für den AIM Assembler. - Disassemblierung von Maschinenprogrammen in den Texteditor in assemblierfähigem Format. - AIM Spezial.- u.a.m.

REDAKTIONS- UND ANZEIGENSCHLUSS FÜR HEFT 12 IST DER 3. APRIL 1980 - GRÜNDONNERSTAG.

MICROPROCESSOR INTERFACE TECHNIKEN



ISBN: 3-922187-00-5 DM 39,-

M & R Medien
Postfach 1122
Marktstraße 3
7778 Markdorf

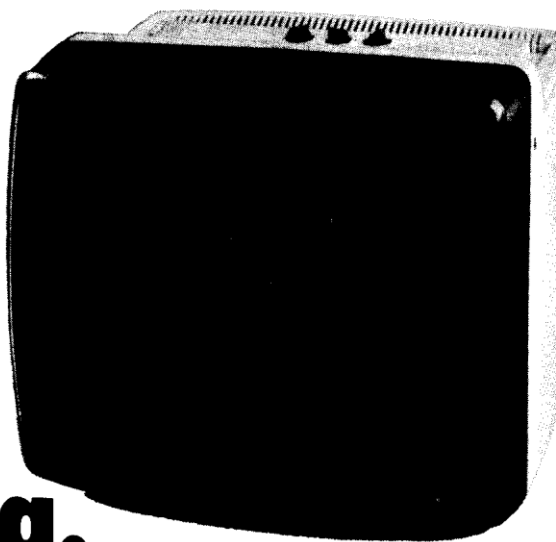
Um es gleich vorwegzunehmen: Das Buch gehört zum Besten, was es auf diesem Sektor in deutscher Sprache gibt. Wer einmal die Grundbegriffe der Mikroprozessortechnik begriffen hat, bekommt hier konkrete Auskunft, wie es jenseits der CPU weitergeht.

Wie schließt man eine Tastatur an? Welche Signale benötigt eine Floppy-Disk-Einheit? Wie sehen die Pegel beim RS232C-Standard aus? Wie funktioniert der IEEE-488-Bus? Die Liste der Themen, die das Buch in erstklassiger Form darstellt, ließe sich noch lange fortsetzen. Obwohl in erster Linie der Entwicklungsingenieur in der Industrie angesprochen wird, kann auch - oder gerade - der Hobbyist sehr viel Nutzen aus diesem Buch ziehen. Mit Sicherheit wird er manches mal überrascht sein, wie einfach die Dinge doch sein können, wenn sie nur richtig erklärt werden.

Auszug ELO 8/79



**Wer beruflich
»in die Röhre schaut«
sieht bei uns grün.
Und dies
nicht nur
scharf,
sondern
vielmehr
auch ruhig.**



Für alle, die mit Bildschirmdaten leben, entwickelte Sanyo den augenfreundlichen Typ DM5912CX. Denn EDV-Informationen lesen, kann weitaus deutlicher und erholsamer sein. So schrieb uns ein anwendungserfahrener Experte: „Der von Ihnen gelieferte Monitor ist ein wirklich angenehmes Display für den Mikroprozessor. Das grüne Bild steht scharf und ruhig und beansprucht mein leider krankes Augenlicht in keiner Weise...“ Das alles wurde von uns in ein formschönes, elfenbeinfarbiges Kunststoffgehäuse mit Rauchglas-Frontblende eingebaut.

Technische Hauptmerkmale:

Bildschirmdiagonale:	12" (31 cm)
Videobandbreite:	18 MHz
Geometriefehler:	kleiner als 1%
Displayformat:	80 Charakter, 24 Zeilen
Eingangssignal:	Standard Video 1 V _{SS}



Ausführliche Informationen mit Fachhändler-Nachweis:
SANYO Video Vertrieb GmbH. & Co., Lange Reihe 29, 2000 Hamburg 1, Telefon (040) 24 62 6

FARBGRAPHIKTERMINAL

Das von mir entwickelte preisgünstige Farbgraphikterminal besteht aus einem Basisgerät und Einschubkassetten in verschiedenen Leistungsstufen und ist an jedes Heimfernsehgerät anschließbar. Als Basisgerät dient die von verschiedenen Firmen für Telespiele angebotene F8-Platine.

Bei allen Ausführungen sind 8192 Bildpunkte adressierbar. Hintergrund- und Vordergrundfarben sind für Zeile und Bildpunkt getrennt wählbar. Eine 15-polige Buchsenleiste verbindet das Farbterminal mit dem anzuschließenden Prozessorsystem über eine parallele Schnittstelle. Die Ansteuerung für alle bekannten Standardterminalfunktionen im ASCII-Code. Farb- und Sonderfunktionen werden durch Kontrollzeichen generiert.

Die Serie beginnt mit dem Gerät STEWE I:

5x7 Bit-Matrix zur Darstellung von Zahlen, Buchstaben und Sonderzeichen (64). Volle Cursor-Kontrolle (zeichenweise, bzw. bitweise) Punkt malen/nicht malen oder löschen. Verschiedene Strichstärken. Frei wählbare Farben für Vorder- und Hintergrund, Positionierung wie beim Cursor, auch für Sonderzeichen und Gruppen. Display löschen, Zeile löschen, pulsierender Cursor ja/nein.

Dieses preisgünstige Gerät der Serie wird nur komplett geliefert, es ist voll ausbau- und erweiterungsfähig. Preis ab DM 748,- inkl. MWSt.

Die Terminalcassette STEWE II bietet darüber hinaus:

Wahlweise Umschaltung auf 7x9 und 5x5 Bit-Matrizen (96 Zeichen bei 7x9). Groß- und Kleinschreibung, Definition von Zeilen- und Zeichenabstand. Vollständige Anwenderprogramme können dauerhaft gespeichert werden (EEPROM).

Ausführung STEWE III:

Zusätzlich mit Tabulator, unabhängigem Laufschriftgenerator. Möglichkeit der Ein- und Ausgabe von selbstdefinierten Zeichen.

Sonderausführungen sind in vielen Variationen erhältlich. Die Cassetten enthalten Mikrocomputersysteme der 6502-Familie. Spannungsversorgung über das Basisgerät.

Das Basisgerät ist ferner ausbaubar als

DIGITAL-MULTIMETER mit farbiger Anzeige der Meßwerte auf dem Bildschirm und als

LOGIKANALYSATOR für 8-48 Kanäle mit binärer, hexadezimaler oder auch gemischter Darstellung. Es können bis zu 9 Logikzustände gespeichert werden. Er ist zudem extern triggerbar.

BITTE VERLANGEN SIE MEINEN SONDERDRUCK!

32 kBYTE-RAM-KARTEN FÜR 8 BIT-MICROCOMPUTER

ANSCHLUSSFERTIG FÜR AIM 65, OSI, ALPHA, KIM, SYM,

PC 100, SC/MP, PET 2001, 6800

Beliebige 2k-Adressenzuordnung (PROM).

Direkt systemkompatibel ohne Motherboard.

Busbelastung 100 μ A, 10 pF.

Nur 5V und 1,2 A erforderlich.

Im Vergleich zu Speichererweiterungen mit dem

IC 2114L ist die Leistungsaufnahme um 80% geringer

und die Kapazität auf einer Europakarte um 200% höher.

DM 894,96 (= 792,- + 13% MWSt) für 32 kB

DM 594,38 (= 526,- + 13% MWSt) für 16 kB.

OEM-Mengenrabatte. 1 Jahr Garantie.

Außerdem bieten wir im Europaformat:

EPROM-KARTE MIT PROGRAMMER,

SCHNELLE A/D- UND D/A-WANDLER,

VIDEO-GRAFIK-INTERFACES.

PROSPEKT ANFORDERN!

DIPL.-ING. HORST NEUDECKER

MEHRINGPLATZ 13, 1000 BERLIN 61, TEL. 614 8900, 251 2000

8 BIT A/D- UND D/A-WANDLER AUF EUROPAKARTE

ANSCHLUSSFERTIG FÜR DEN SYSTEMBUS (EXPANSION-CONNECTOR)
VON AIM 65, OSI, ALPHA, PC 100, KIM, SYM, PET 2001, CBM,
SC/MP, 68000.

Aufbau: Vorverstärker (Eingangswiderstand $> 1 \text{ M}\Omega$),

20 kHz-Tiefpass (aktives Filter 4. Ordnung), Abtast-
und Halteschaltung (Aperturzeit $1 \mu\text{s}$),

A/D-Wandler (Abtastfrequenz bis zu 80 kHz, Wandlungszeit
minimal $12 \mu\text{s}$) - (Computer) -

D/A-Wandler (Einschwingzeit $1 \mu\text{s}$), Ausgangsfilter,
Obersteuerungsanzeige.

Anwendungen: Analogdatenerfassung jeder Art,
industrielle Prozeßüberwachung, med. Anwendungen (EKG, EEG),
NF-Transientenrekorder, NF-Speicheroszilloskop,
elektronischer Hall, Klangeffekte,
Scrambler, Korrelationsanalyse,
Sprachanalyse und -synthese,
Funktionsgenerator,
programmierbare digitale Filter.

Einige ausgesuchte interessante 6502-Programme werden
mitgeliefert.

PREIS: DM 350,- + MWST.

DIPL.-ING. HORST NEUDECKER

MEHRINGPLATZ 13, 1000 BERLIN 61, TEL. 614 8900, 251 2000

GWK

FÜR TECHNISCHE

GESSELLSCHAFT
ELEKTRONIK mbH.

D 5120 Herzogenrath
Asterstr. 2
Tel.: 02406/62394

NEU! NEU! NEU! NEU! NEU!

P C 1 0 0 , MICROCOMPUTER VON SIEMENS

Komplettgerät im Gehäuse. Ausgestattet mit 4K Byte RAM,
BASIC Interpreter und Netzteil. Garantiezeit beträgt 6 Monate.

1.990,--DM +1

V I D E O M O N I T O R , S A N Y O D M 5 9 1 2 C X

Datenmonitor mit grünleuchtendem 12" Bildschirm. Stellt 24 Zeilen
mit je 80 Zeichen dar. Bandbreite 18 MHz. Professionelle Qualität.

798,--DM +1

M A T R I X D R U C K E R , T X 8 0

Groß und Kleinschreibung, 95 ASCII Charakter plus 64 Grafik
Charakter. Normalschrift und Breitschrift. 2 Zeilenabstände
wählbar. Verstellbare Tractorführung für Papierbreiten bis 254mm.
80 Zeichen pro Zeile, 150 Zeichen pro Sekunde.
Schnittstelle: 8 BIT Parallel plus Handshake Control.
Lieferung inklusive SOFTWARE zur Ansteuerung über den
USER Output Handler bei Anschluss an User VIA von AIM 65 und
PC 100.

2.448,--DM +1

Alle Preise zuzüglich MWST und Versandkosten.

GWK EURO BOARD EXPANSION SYSTEM

Microcomputer AIM 65,1 K Byte RAM	875,--	988,75 DM
Microcomputer AIM 65,4 K Byte RAM	1050,--	1186,50 DM
Resident ASSEMBLER,4K,in ROM	230,--	259,90 DM
BASIC,Interpreter,8K,in ROM's	280,--	316,40 DM
BASIC ERWEITERUNG,2K,auf Cassette	145,--	163,85 DM
BASIC ERWEITERUNG,2K,in EPROM,steckbar in Z 24	280,--	316,40 DM
BASIC LISTING,ca 70 Seiten DIN A4,gut kommentiert	45,--	50,85 DM
AIM 65 ANWENDERHANDBUCH,deutsch	35,--	39,55 DM
AIM 65 MANUALS,englisch	je 15,--	16,95 DM
AIM 65 SCHALTPLANPOSTER	5,--	5,65 DM
THERMOPAPIER,approved,schwarzdruckend,lagerfähig,50m	6,50	7,35 DM
STECKERLEISTEN,44 Pole,nach MIL-C-21 097	12,50	14,13 DM

MOTHER BOARD,12 Steckplätze f.Expansion,3 f.Appl. 485,-- 548,05 DM
direkt an AIM steckbar,oder Einbau in 19" Gehäuse.Voll gepuffert.

ADAPTER BOARD,Busplatine ohne Pufferung,9 Steckplätze 295,-- 333,35 DM

RAM BOARD,3 x 4 K Byte statisches RAM,unabhängig voneinander beliebig adressierbar,write protectable. 945,-- 1067,85 DM

EPROM BOARD,12 K Byte,für die 1K EPROM's 2708 oder 2758. 395,-- 446,35 DM

EPROM PROGRAMMER BOARD,mit res.Betriebsprogramm,Softwareumschaltung für 1K,2K und 4K Eprom's mit einer Versorgungsspannung. 795,-- 898,35 DM

VIA/PIA BOARD,32 I/O Kanäle,beliebig adressierbar. 335,-- 378,55 DM

PROTOTYP BOARD,Experimentierplatine zum Wrappen oder Fädeln.Durchdachte Aufteilung,über 1800 Rasterbohrungen. 75,-- 84,75 DM

PET INTERFACE,ermöglicht den Anschluss der GWK BOARD's 425,-- 480,25 DM
an den PET.Pufferung,Rückgewinnung der oberen Adressbits.

POWER SUPPLY BOARD,hochwertiges Schaltnetzteil incl.Trafo.

5V/3A;26V/o,7A 315,-- 355,95 DM

5V/3A,Notstromversorgung;26V/o,7A;+/- 12V/o,7A 450,-- 508,50 DM

5V/6A,Notstromversorgung;26V/o,7A;+/- 12V/o,7A 595,-- 672,35 DM

GEHÄUSE,für AIM,Kunststoff 147,-- 166,11 DM

GEHÄUSE,für AIM,Metall mit Platz für Netzteil u.kl.Erweiter. 345,-- 389,85 DM

PULTGEHÄUSE,für AIM und System,Metall,mit 19" Baugruppenträger. 595,-- 672,35 DM

Diverse LSI Chips:6500 Familie,EPROM's,RAM's,usw bitte anfragen.

Alle Preise zuzüglich Versandkosten.

zzgl MWST incl.MWST

65_{xx} MICRO MAG

COMPUTING · SOFTWARE · HOBBY

HERAUSGEBER:

DIPL.-VOLKSWIRT ROLAND LÖHR

HANSDORFER STRASSE 4

2070 AHRENSBURG

☎ (04102) 55 816

65xx MICRO MAG erscheint zweimonatlich. Beiträge, die nicht besonders gekennzeichnet sind, stammen vom Herausgeber. COPYRIGHT 1980 by Roland Löhr. Alle Rechte vorbehalten, auch die des auszugsweisen Nachdrucks, der Übersetzung, der fotomechanischen Wiedergabe und die der Verbreitung auf magnetischen und sonstigen Trägern. Offsetdruck: Druckartist Gerhard M. Meier, Hamburg 70.

BEZUGSBEDINGUNGEN: Abonnement ab laufender Ausgabe für 6 Hefte DM 45,- (Inlandsendpreis). Ausland/Foreign via surface mail DM 50,-, USA air \$30. Die früher erschienenen Ausgaben dieser Zeitschrift sind nachlieferbar. Einzelpreis für Nos. 1-6 DM 7,-/St., Nos. 7-10 DM 7.80/St. Ein Sammelband für die früheren Ausgaben ist vorgesehen. Richten Sie bitte Ihre Überweisungen/Schecks an Roland Löhr, Konto 20/01121 bei der Vereins- und Westbank in Ahrensburg, BLZ 200 300 00. Zuschreibung auch über das Postscheckkonto der Vereinsbank: PSchA Hamburg 2244-207.

SERVICE AIM 65 - PC 100

Anwenderhandbuch für AIM 65 in deutscher Sprache.
Original Rockwell-Handbuch mit zahlreichen Verbesserungen gegenüber der englischen Vorlage.

Ab Lager lieferbar. Endpreis

DM 32,10

Thermopapier für AIM 65/PC 100 in kontrastreicher Spitzenqualität. Qualitätsfreigabe durch Rockwell, Kalifornien. Packung mit 8 Großrollen, zus. 520 m, 57 mm breit, preiswert. Packung

DM 50,85

Vorstehende Preise gelten für Vorkasse. Nachnahme + DM 1,50.
Bezug beim Herausgeber.

BUCHANGEBOT

Camp, Smay, Triska: Microprocessor Systems Engineering. Lehrbuch für 65xx und speziell für AIM 65, besprochen auf Seite 11 in diesem Heft. DM 64,- + Versandkostenanteil DM 2,-.

PROGRAMMIERWORKSHOPS 65xx

2-tägige Wochenendworkshops des erfahrenen Dozententeams Anfang Mai im Raum Frankfurt. Einführungsstufe: Darstellung und Übung des Befehlssatzes in Maschinensprache. Fortgeschrittenenstufe: Formulierung von Quellenprogrammen in Assemblersprache am AIM 65/PC 100 oder am PET/CBM, Interfaceprogrammierung. Besondere Lehrgangsunterlagen. Übertragung des Vortrages auf Bildschirmmonitore. Anmeldung und Sonderprospekt beim Herausgeber.

PRINTERPLATTEN FÜR AIM 65 WIEDER LIEFERBAR! DM 21,-/St.