

65_{xx} MICRO MAG

COMPUTING · SOFTWARE · HOBBY

DM 7,-

Nr. 9

O K T O B E R 1 9 7 9

RECHNERVERBUND

Dieses Heft zeigt Möglichkeiten auf, zwei der geläufigsten Prozessorsysteme, AIM und PET, im Datenverbund zu betreiben - ein Anfang. Der Verbund dieses Heftes geht an vielen Stellen weiter: Der Lösungsweg zahlreicher abgedruckter Dienstleistungsprogramme kann auch auf andere Systeme der Familie übertragen werden - Erfreulich ist der mehr und mehr eintretende Anschluß des BASIC an wesentlich schneller ausführende Maschinenprogramme und die sich für BASIC abzeichnende Sprach-erweiterung.

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

TRACE (PET)	3
VIEW (PET)	5
Text-Editor (PET)	6
AIM-Spezial (4)	8
PETROL - Bildschirm rollen	9
Restore Line Numer (PET)	13
Datenverbund zwischen AIM 65 und PET 2001	14
Assembler-Listing für den AIM 65	17
ASP - Advanced Subroutine Package (Teil 6)	22
Align für Alpha-Sort	28
KOORD - Plotten mit dem AIM-Printer	29
Magnetbandbetrieb	32
Umbau eines Cassettenrecorders	35
Auskunftssystem mit dem AIM 65 (2)	37
Ein Leitfaden für die Programmierung (Teil 4)	41

GWK EURO BOARD EXPANSION SYSTEM

SYSTEMERWEITERUNG FÜR **AIM 65** **KIM 1** **SYM**

Das GWK EURO BOARD EXPANSION SYSTEM bietet sowohl dem professionellen Anwender, als auch dem Hobbyisten, die Möglichkeit, seinen AIM 65, KIM 1, oder SYM optimal zu erweitern.

Dies wird erreicht durch die folgenden grundlegenden Konzepte:

Die Adressdecodierung der einzelnen Komponenten erfolgt auf jeder Karte. Das heißt, jede Komponente kann beliebig adressiert werden. Es wird kein Adressraum verschenkt.

Der GWK SYSTEM BUS und die Steckverbinder der Einzelkomponenten sind inkompatibel zum Expansionconnector von AIM, KIM und SYM.

Hierdurch wird erreicht, daß einzelne BOARD'S auch ohne Motherboard direkt am entsprechenden Microcomputer laufen.

Bisher sind folgende Komponenten lieferbar:

MOTHER BOARD

485,--DM

Einbau in Gehäuse 19 Zoll oder direkt an AIM, KIM, SYM steckbar.

Daten-, Adress- und Kontrollbus voll gepuffert und dynamisch abgeschlossen. 9 Steckplätze für Expansion, 3 für Application. Genügend Platz für ein Netzteil.

RAM BOARD

945,--DM

3 x 4 K Byte statisches RAM. Unabhängig voneinander beliebig adressierbar.

EPROM BOARD

395,--DM

12 K Byte für die 1K Eproms 2708 oder 2758

EPROM PROGRAMMIERBOARD

795,--DM

Mit residentem Betriebsprogramm. Softwareumschaltung

für 1K-, 2K-4K Byte EPROM'S mit einer Betriebsspannung, wie 2758, xx16, xx32

VIA - PIA BOARD

335,--DM

32 I/O Kanäle. Beliebige Adressierbar. Wahlweise mit VIA oder PIA bestückbar. Zwei Übergabesteckverbinder.

PET INTERFACE Pufferung, Adreßrückgewinnung

425,--DM

Adapterboard, Rückwandverdrahtung 1:1, 9 Steckplätze

295,--DM

In Vorbereitung: VIDEO INTERFACE, ANALOG I/O, CPU-BOARD, POWER SUPPLY, UNIVERSAL MEMORY BOARD.

Alle Preise zuzüglich Mehrwertsteuer und Versandkosten.

GWK TECHNISCHE ELEKTRONIK GmbH
HARDWARE SOFTWARE SYSTEMENTWICKLUNG

5120 HERZOGENRATH, ASTERNSTRASSE 2

Tel. 02406-62394

65xx MICRO MAG

TRACE (PET)

Peter W. Arps, Brockdorffstraße 5, D-2000 Hamburg 73

E: A combined Machine Language and BASIC program to trace BASIC programs.

Im BASIC-Programm TRACE wird der Maschinencode des eigentlichen TRACE-Programmes mittels POKE-Befehlen in den hinteren RAM-Bereich übertragen und der Pointer für die höchste RAM-Adresse auf 7882 gesetzt.

Das TRACE-Maschinenprogramm wird mit SYS(7888) ein- und ausgeschaltet.

TRACE nutzt die Eigenschaft des Interpreters, die BASIC-Programmzeilen Byte für Byte über die CHRGOT-Routine auf Adresse 194 zu lesen. Beim Einschalten des TRACE wird diese Routine 'aufgebohrt', d.h. die CHRGOT-Routine wird so modifiziert, daß zunächst die TRACE-Ausgaberroutine aufgerufen wird. Hier wird ermittelt, ob der Pointer (Bytes 201 und 202) auf den Anfang einer Programmzeile zeigt oder ob eine direkte Eingabe vorliegt (Byte 202 = 0). Handelt es sich um den letzteren Fall oder zeigt der Pointer nicht auf einen Zeilenanfang, so wird die TRACE-Ausgabe übergangen. Anderenfalls wird die decodierte BASIC-Zeile in reverser Darstellung in den beiden oberen Bildschirmzeilen angezeigt.

Bevor TRACE zum Interpreter zurückkehrt, wird eine Verzögerungsschleife von ca. 1,8 Sekunden durchlaufen. Deren Dauer kann durch die Veränderung des Bytes 8126 (Defaultwert=3) verändert werden.

```

10 PRINT "TAB(10)" "*****"
20 PRINTTAB(10)"* TRACE *"
30 PRINTTAB(10)"*****"
40 PRINT:PRINT
50 PRINT"DAS TRACE-PROGRAMM WIRD MIT [R]SYS(7888)"
60 PRINT"EIN- UND AUSGESCHALTET."
70 PRINT
80 PRINT"DIE DAUER DER VERZOEGERUNGSSCHLEIFE KANN MIT [R]POKE8126,X [R]VERAENDERT";
90 PRINT"WERDEN"
100 PRINT
110 PRINT"WENN 'SHIFT' GEDRUECKT WIRD, WIRDDIE VERZOEGERUNGSSCHLEIFE";
120 PRINT"UEBERGANGEN"
130 REM
140 POKE 134,202: POKE 135,30
150 FOR I=7882 TO 8187: READ A:POKE I,A:NEXT
160 PRINT:PRINT:PRINT
170 PRINT"[R]TRACE ON ":SYS(7888):PRINT
180 FOR I=1 TO 3
190 X=X+1
200 X=INT(I*X)
210 NEXT
220 PRINT"[R]TRACE OFF$SYS(7888)

1000 DATA 32,249,30,32,239,30,162,5,36,194,48,10,189,181,224,149,194,202,16
1010 DATA 248,48,8,189,202,30,149,194,202,16,248,165,210,73,29,133,210,96,230
1020 DATA 201,208,2,230,202,96,76,211,31,8,138,72,152,72,166,202,240,244,165
1030 DATA 201,56,233,8,176,1,202,133,88,134,89,160,4,177,88,208,226,133,82
1040 DATA 133,83,133,84,133,86,160,0,177,88,240,212,160,7,177,88,133,80,200
1050 DATA 177,88,133, 81,248,160,15,6,80,38,81,162,253,181,85,117,85,149,85
1060 DATA 232,48,247,136,16,238,216,162,2,181,82,134,87,72,74,74,74,32,221

```

[C] = CLEAR SCREEN

[D] = CURSOR DOWN

[R] = REVERSE ON

[] = REVERSE OFF

65_{xx} MICRO MAG

1070 DATA 31,104,41,15,32,221,31,166,87,202,16,233,32,217,31,160,9,132,87,164
 1080 DATA 87,177,88,240,75,230,87,201,34,208,8,170,165,186,73,255,133,86,138,9
 1090 DATA 0,16,46,36,86,48,44,201,255,208,4,169,94,208,34,41,127,170,160,0
 1100 DATA 185,145,192,48,3,200,208,248,200,202,16,244,185,145,192,48,12,132
 1110 DATA 85,41,63,32,233,31,164,85,200,208,239,41,63,32, 233,31,76,99,31,32
 1120 DATA 217,31,224,79,144,249,173,4,2,208,22,169,3,133,87,169,0,133,88,133
 1130 DATA 89,198,89,208,252,198,88,208,248,198,87,208,244,104,168,104,170,40
 1140 DATA 96,169,32,208,12,166,80,208,4,9,0,240,22,230,80,9,48,9,128,166,81
 1150 DATA 168,173,64,232,41,32,208,249,152,157,0,128,230,81,96

Der Herausgeber hat das mit den vorstehenden READ- und DATA-Statements erzeugte Maschinenprogramm zur Übersicht einmal im AIM-Format disassembliert:

1ECA 20 JSR 1EF9	1F1A 85 STA 56
CD 20 JSR 1EEF	1C A0 LDY \$00
D0 A2 LDX \$05	1E B1 LDA (58),Y
D2 24 BIT C2	20 F0 BEQ 1EF6
D4 30 BMI 1EE0	22 A0 LDY \$07
D6 BD LDA E0B5,X	24 B1 LDA (58),Y
D9 95 STA C2,X	26 85 STA 50
DB CA DEX	28 C8 INY
DC 10 BPL 1ED6	29 B1 LDA (58),Y
DE 30 BMI 1EE8	2B 85 STA 51
E0 BD LDA 1ECA,X	2D F8 SED
E3 95 STA C2,X	2E A0 LDY \$0F
E5 CA DEX	30 06 ASL 50
E6 10 BPL 1EE0	32 26 ROL 51
E8 A5 LDA D2	34 A2 LDX \$FD
EA 49 EOR \$1D	36 B5 LDA 55,X
EC 85 STA D2	38 75 ADC 55,X
EE 60 RTS	3A 95 STA 55,X
1EEF E6 INC C9	3C E8 INX
F1 D0 BNE 1EF5	3D 30 BMI 1F36
F3 E6 INC CA	3F 88 DEY
F5 60 RTS	40 10 BPL 1F30
1EF6 4C JMP 1FD3	42 D8 CLD
1EF9 08 PHP	43 A2 LDX \$02
FA 8A TXA	45 B5 LDA 52,X
FB 48 PHA	47 86 STX 57
FC 98 TYA	49 48 PHA
FD 48 PHA	4A 4A LSR .A
FE A6 LDX CA	4B 4A LSR .A
1FF0 F0 BEQ 1EF6	4C 4A LSR .A
02 A5 LDA C9	4D 4A LSR .A
04 38 SEC	4E 20 JSR 1FDD
05 E9 SBC \$08	51 68 PLA
07 B0 BCS 1FOA	52 29 AND \$0F
09 CA DEX	54 20 JSR 1FDD
0A 85 STA 58	57 A6 LDX 57
0C 86 STX 59	59 CA DEX
0E A0 LDY \$04	5A 10 BPL 1F45
10 B1 LDA (58),Y	5C 20 JSR 1F09
12 D0 BNE 1EF6	5F A0 LDY \$09
14 85 STA 52	61 84 STY 57
16 85 STA 53	63 A4 LDY 57
18 85 STA 54	65 B1 LDA (58),Y
	67 F0 BEQ 1FB4
	69 E6 INC 57

65xx MICRO MAG

1F6B C9 CMP \$22	1FB6 90 BCC 1FB1
6D D0 BNE 1F77	B8 AD LDA 0204
6F AA TAX	BB D0 BNE 1FD3
70 A5 LDA BA	8D A9 LDA #03
72 49 EOR #FF	BF 85 STA 57
74 85 STA 56	C1 A9 LDA #00
76 8A TXA	C3 85 STA 58
77 09 ORA #00	C5 85 STA 59
79 10 BPL 1FA9	C7 C6 DEC 59
7B 24 BIT 56	C9 D0 BNE 1FC7
7D 30 BMI 1FAB	CB C6 DEC 58
7F C9 CMP #FF	CD D0 BNE 1FC7
91 D0 BNE 1F87	CF C6 DEC 57
83 A9 LDA #5E	D1 D0 BNE 1FC7
85 D0 BNE 1FA9	D3 68 PLA
87 29 AND #7F	D4 A8 TAY
89 AA TAX	D5 68 PLA
8A A0 LDY #00	D6 AA TAX
8C B9 LDA C091,Y	D7 28 PLP
8F 30 BMI 1F94	D8 60 RTS
91 C8 INY	
92 D0 BNE 1F8C	1FD9 A9 LDA #20
94 C8 INY	DB D0 BNE 1FE9
95 CA DEX	DD A6 LDX 50
96 10 BPL 1F8C	DF D0 BNE 1FE5
98 B9 LDA C091,Y	E1 09 ORA #00
9B 30 BMI 1FA9	E3 F0 BEQ 1FFB
9D 84 STY 55	E5 E6 INC 50
9F 29 AND #3F	E7 09 ORA #30
A1 20 JSR 1FE9	E9 09 ORA #80
A4 A4 LDY 55	EB A6 LDX 51
A6 C8 INY	ED A8 TAY
A7 D0 BNE 1F98	EE AD LDA E840
A9 S9 AND #3F	F1 29 AND #20
AB 20 JSR 1FE9	F3 D0 BNE 1FEE
AE 4C JMP 1F63	F5 98 TYA
B1 20 JSR 1FD9	F6 9D STA 8000,X
B4 E0 CPX #4F	F9 E6 INC 51
	FB 60 RTS.

VIEW

Peter Trübger, Melissenweg 37, 2000 Hamburg 65

E: VIEW displays the contents of any PET memory page on the screen. This is accomplished by manipulating the interrupt vector to a small routine, which writes the sampled page to the screen memory.

Im Pet findet 60mal in der Sekunde ein Hardwareinterrupt statt: Die Tastatur wird abgefragt, die Zeitvariablen nachgestellt und schließlich das Bildschirm-RAM ausgelesen. Der Interrupt-Vektor steht in den Speicherzellen 537 und 538, man kann ihn per Programm ändern (manuellesPOKE ist zu langsam).

So kann man z.B. mit POKE 537,136 erreichen, daß die Tastatur nicht mehr abgefragt und die Uhr nicht nachgestellt wird. Vorteil: Bei Betätigung der STOP-Taste steigt der PET nicht mehr aus dem Programm aus, die Rechenzeit wird verkürzt. Eine weitere Möglichkeit wurde vor kurzem in CHIP veröffentlicht: Durch Versetzen des Interrupt-Vektors wurde die REPEAT-Funktion verwirklicht.

Ein anderes besonders simples Beispiel ist das nachfolgende VIEW als ein aus READ und DATA gebildetes Assemblerprogramm, das im Speicher des zweiten Cassettenlaufwerkes residiert. Es vermittelt interessante Einsichten in alle Veränderungen in jeder beliebigen Speicherseite. Man sieht die Uhr ticken, die Pointer in den ersten Seiten sich ändern, den Eingabepuffer sich füllen und Variable neue Werte annehmen. Die Tastatur bleibt aktiv, ebenso das BASIC oder auch ein Maschinenprogramm. VIEW ist ein Programm zum Studieren des PET. Wir drucken sowohl das Maschinenprogramm wie auch das in BASIC:

*=826

IRQL=537

IRQH=538

KEY=59013

826 SEI	DER IRQ-VEKTOR WIRD	845 LDX #0	SEITENWAHL
LDA IRQL	MIT SYS 826 JEWEILS	L1 LDA \$0100,X	(PAGE) DURCH
EOR #200	AUF 59013 BZW. WIEDER	STA \$8000,X	POKE 849,0..255,
STA IRQL	AUF 845 GESTELLT	DEX	D.H. DURCH
LDA IRQH	(FLIP-FLOP-FUNKTION),	BNE L1	PROGRAMM-MO
EOR #229	D.H. VIEW-FUNKTION	JMP KEY	DIKFIKATION
STA IRQH	EIN - AUS.	.END	
CLI			
RTS			

DIE SEQUENZ AB 845 BRINGT DEN INHALT DER AUSGEWÄHLTEN PAGE OBEN IN DEN BILDSCHIRM.

```
100 FOR J=826 TO 858: READ X:POKE J,X:NEXT:END
110 DATA 120,173,25,2,73,200,141,25,2
120 DATA 173,26,2,73,229,141,26,2,88,96
130 DATA 162,0,189,0,0,157,0,128,202,208
140 DATA 247,76,133,230
```

TEXT-EDITOR (PET)

Karlheinz Lehner, Zeppelinstraße 29, 6236 Eschborn 1

Dieser für den PET bestimmte Text-Editor dient der Textbearbeitung, Speicherung auf Cassette und der Wiedereinlesung von dort. Es sind sicher nicht alle Funktionen verwirklicht, die man an ein Textbearbeitungssystem stellen kann, der Editor ist jedoch nach Bedarf ausbaufähig. Seine Befehle werden nach der Anzeige des Menues durch Kleinbuchstaben ausgelöst. Folgende Funktionen sind verwirklicht:

- l = LIST - Zeige alle Zeilen oder ab vorgegebener Zeilennummer,
- r = RENUMBER - Texteingaben können ohne oder mit Zeilennummer erfolgen. Automatisch werden die Nummern in der Folge 5,10,15 ...

65xx MICRO MAG

vergeben. Eigene Zeilennummern bewirken eine Besetzung der Zwischenräume durch Einfügungen. RENUMBER bewirkt eine Neuordnung der Zeilennummern im 5-er Raster.

- f = FIND Es werden alle Zeilennummern gelistet, in denen der Suchstring vorkommt. Nach dem Prompt des Fragezeichens gibt man zunächst einen SPACE vor dem Suchstring ein.
- s = SAVE Der Inhalt des Textspeichers wird mit Namen auf Cassette abgespeichert.
- i = INPUT Ein Textfile wird mit Namen von Cassette in den Textspeicher geladen.

 = CLEAR SCREEN

 = REVERSE ON

 = REVERSE OFF

```

1 POKE 134,0:POKE 135,14%MU=256*14
5 REM TEXT-EDITOR BY KARLHEINZ LEHNER
10 PRINT "COMMANDS:";PRINT
15 PRINT "LIST AB ...":PRINT "Renumber":PRINT "Find ...":PRINT "Save file":
PRINT "Input file"
40 DEFFNP(X)=PEEK(X)+256*(PEEK(X+1))
41 DEFFNH(X)=INT(X/256)
42 DEFFNL(X)=X-256*FNH(X)
43 DEFFNI(X)=X+PEEK(X)+3
44 M=MU:ML=0:POKE 59468,14
45 POKE 527,34:POKE 525,1:GOTO 100
49 REM GET LINE IN MEM
50 AS="":LN=FPN(L-2):FOR K=1 TO PEEK(L):AS=AS+CHR$(PEEK(K+L)):NEXT:RETURN
59 REM GET REST OF AS
60 K=0
61 K=K+1:K$=MID$(AS,K,1):IF K$=" "OR(K$>="0"AND K$<="9") THEN 61
62 AS=MID$(AS,K):RETURN
69 MOVE MEM RIGHT
70 FOR K=M TO L-2 STEP-1:X=PEEK(K):POKE K+DI,X:NEXT:RETURN
79 REM MOVE MEMORY LEFT
80 FOR K=L-2 TO M:X=PEEK(K):POKE K+DI,X:NEXT:RETURN
99 REM *GET LINE*
100 INPUT AS:POKE 527,34:POKE 525,1:IF AS=""THEN POKE 525,0:END
110 X$=LEFT$(AS,1):IF X$=" " THEN 400
120 IF X$="r" THEN 500
130 IF X$="s" THEN 600
140 IF X$="f" THEN 700
150 IF X$="i" THEN 800
200 IF VAL(AS)=0 THEN ML=ML+5:LN=ML:GOTO 210
205 LN=VAL(AS):GOSUB 60:IF LN<=ML THEN 300
210 POKE M,FNL(LN):POKE M+1, FNH(LN):POKE M+2, LEN(AS):M=M+2
220 FOR K=1 TO LEN(AS):POKE M+K,ASC(MID$(AS,K,1)):NEXT:M=M+LEN(AS)+1:GOTO 100
299 REM *INSERT/DELETE A LINE*
300 GOSUB 60:L=MU+2
310 IF FNP(L-2)<LN THEN L=FNI(L):GOTO 310
320 IF FNP(L-2)>LN THEN 370
330 IF AS=""THEN DI=-PEEK(L)-3:L=FNI(L):GOSUB 80:M=M+DI:GOTO 100
340 DI=LEN(AS)-PEEK(L):IF DI<0 THEN P=PEEK(L):L=FNI(L):GOSUB 80:L=L-3-P:
GOTO 360
350 IF DI>0 THEN P=PEEK(L):L=FNI(L):GOSUB 70:L=L-3-P
360 POKE L,LEN(AS):FOR K=1 TO LEN(AS):POKE K+L,ASC(MID$(AS,K,1)):NEXT:
M=M+DI: GOTO 100
370 DI=3+LEN(AS):GOSUB 70:POKE L-2, FNL(LN):POKE L-1,FNH(LN):GOTO 360

```



```

399 REM *LIST*
400 LI=VAL(MID$(A$,2)):L=MU+2
410 IF FNP(L-2)<LI THEN L=FNI(L):GOTO 410
420 GOSUB 50:PRINT LN;TAB(6);A$:L=FNI(L):IF L<M THEN 420
430 GOTO 100
499 REM *RENUMBER*
500 L=MU+2:I=5
510 POKE L-2,FNL(I): POKE L-1,FNH(I):L=FNI(L):IF L>M THEN ML=I:GOTO 100
520 I=I+5:GOTO 510
599 REM *SAVE*
600 POKE 527,34:POKE 525,1:INPUT"FILE NAME";NA$:POKE 243,122:POKE 244,2:
OPEN 1,1,1,NA$

605 L=MU+2
610 GOSUB 50:PRINT#1, CHR$(34);A$:POKE 59411,53:L=FNI(L):IF L<=M THEN 610
620 CLOSE 1:GOTO 100
699 REM *FIND*
700 F$=MID$(A$,3):L=MU+2:PRINT"SEARCHING FOR ";F$
710 IF PEEK(L)<LEN(F$) THEN 740
720 GOSUB 50:FOR K=1 TO LEN(A$)-LEN(F$):IF MID$(A$,K,LEN(F$))=F$ THEN PRINT LN,
730 NEXT
740 L=FNI(L):IF L<M THEN 710
750 PRINT:GOTO 100
799 REM *INPUT FILE*
800 OPEN 1:L=MU+2:I=5
810 INPUT#1,A$:IF ST=64 THEN CLOSE 1:ML=I:M=L:GOTO 100
820 PRINT A$:POKE L-2,FNL(I):POKE L-1,FNH(I):POKE L,LEN(A$)
830 FOR K=1 TO LEN(A$):POKE L+K, ASC(MID$(A$,K,1)):NEXT L:L=FNI(L):I=I+5:
GOTO 810

```

AIM-SPEZIAL (4)

Viele Leser fragen nach dem kommentierten Programmlisting für das BASIC. Es steht uns an keiner Stelle zur Verfügung. Wegen der vielen Sprünge hilft eine Disassemblierung der ROMs kaum weiter. Interessanter wären schon Studien des Stacks, um vorangehende Aufrufe zu erkennen.

Aus dem Gedankenaustausch mit den Lesern und besonders mit Herrn Bernhard Kokula in Ludwigshafen sind folgende praktischen Hinweise zu vermitteln: Assembler und besonders BASIC graben die Zeropage knietief um. Bei gemischtem Betrieb rette man also rechtzeitig seine eigenen wertvollen Parameter bzw. wundere sich nicht über deren Zerstörung.

Der I-Command des Monitors druckt normal eine Zeile aus, wenn mit RETURN abgeschlossen wird. Die Möglichkeit des Abschlusses mit SPACE ist nicht überall geläufig. Sie erzeugt eine zweite Zeile mit dem erzeugten Befehl in hex.

Der F-Befehl (FIND) wird mit demselben String repetiert, wenn man nach Auffinden der ersten Zeile mit gleichem String wiederum F und RETURN betätigt (fortgesetztes Suchen).

K-Command: Drücken zweier Tasten mit Freigabe je einer Taste im Wechsel bewirkt Einzelschritt-Disassemblierung, mit SPACE kann dabei dauerhaft angehalten werden. Ähnlich auch bei L (LIST) aus dem Editor und beim Assembler. Die Zeilenangabe SPACE bewirkt bei K endloses Disassemblieren, bei L Listing ab offener Zeile bis zum Ende des Textbuffers.

R.L.

PETROL

Wikhart Löhr, Hansdorfer Str. 4, 2070 Ahrensburg

E: An image stored in a set aside memory or on the screen may be rolled in any direction, up and down, right and left.

In den Hefte 7 und 8 veröffentlichte Herr Kornnagel seine schönen Videoprogramme für den PET, den VIDEO-DRIVER und VIDEOEDIT. Das letztere Programm gestattet es, am Ende des RAM-Speichers eine edierte Grafik dauerhaft abzulegen und sie mit irgendeinem aufrufenden Programm zu verwerten.

Es lag nun der Gedanke nahe, diesen Bildspeicher 'rollbar' zu machen, eine Bildzeile wird z.B. oben entfernt, alle übrigen Bildzeilen rutschen um eins nach oben und die oben entfernte Zeile wird unten wieder angefügt. Ein solches Dienstprogramm muß man in Maschinencode ausführen, denn BASIC wäre für die 60 Bildwechsel in der Sekunde viel zu langsam. Lediglich das interaktive Aufrufprogramm wurde in BASIC geschrieben, man könnte seine Funktion allerdings auch in die Maschinenebene legen.

Das Rollprogramm wurde als notwendige Basis für beabsichtigte Spiele entwickelt. Es dürfte allerdings auch für ernsthafte Anwendungen gleichermaßen geeignet sein. Es ist bekannt, daß Bewegungen auf dem Bildschirm das Interesse des Betrachters auf sich ziehen. Man kann also zu Werbezwecken grafische 'action' auf den Bildschirm bringen oder auch eine Spruchrolle umlaufen lassen. Die vielen Möglichkeiten konnten noch gar nicht ausprobiert werden. Auf jeden Fall ist die Edierung einer Grafik mit dem o.e. Video-Editor viel einfacher und bequemer als mit den zahlreichen andernfalls benötigten PRINT-Statements, die entweder eine Breitenbegrenzung bewirken, wenn man das Bild in der Zeile ediert, oder die viel Geduld kosten, wenn das PRINT-Statement in die zweite Zeile läuft.

Der Anwender mag gleichwohl beides kombinieren: Er ediert seine Grafik, rollt sie dann vielleicht, und holt sie mit SYS(6957) auf den Bildschirm zurück. Mit einer Kette von PRINTs oder auch mit SYS-Aufrufen von Funktionen des VIDEO-DRIVERS kann jetzt ein feststehendes Bild darübergerlegt werden. Die Möglichkeiten sind damit nicht erschöpft. Das VIDEO-EDIT legt eine Grafik oder einen Text in 1000 Bytes ab. Man könnte seine Leistungskraft so erweitern, daß z.B. 2-4000 Plätze der Speicherung von Bildschirm-inhalten dienen und man könnte das Rollen mit PETROL auf diesen ganzen Bereich ausdehnen. Der Lösungsweg ist ja vorgezeichnet, man müßte nur größere Parameter in die definierten Übergabestellen PX bzw. PY mit POKE übertragen und die Basisadressen ändern. Man kann natürlich auch kleinere Parameter wählen, um nur einen Teil des Bildes zu rollen.

Es sind alle 4 möglichen Rollbewegungen verwirklicht, das Rollen nach rechts, nach links, nach oben und nach unten. Bei den ausführenden Routinen handelt es sich um reine Transportprogramme (MOVE) zusammen mit der notwendigen Adressenberechnung für jeden Schleifendurchlauf. Der gewählte Lösungsweg kann für alle Prozessoren angewendet werden, deren Bildspeicher direkt adressiert wird aber auch für als Terminal betriebene Monitore, für die man eine Kopie des Bildspeichers im RAM des Hauptprozessors unterhält. Man wird dann allerdings darauf achten müssen, daß man Steuerzeichen, wie CR und LF, nicht mitrollt.

Zum Aufbau des Programmes ist folgendes zu bemerken: VIDEOEDIT verblieb an den in Heft 8 dokumentierten Speicherplätzen. Bis zum Anfang des Bildspeichers verblieb dahinter nur wenig Platz. Er dient der Aufnahme

und Übergabe der Parameter an feste Stellen (POKE), die mit INIT in den Fließkomma-Akku des PET übertragen werden (Adressen 80-83 für FROM und TO). Es schließen sich noch einige Unterprogramme für die Umrechnung der Pointer an (Erhöhung oder Verminderung um dezimal 40, für eine Zeilenlänge auf dem Bildschirm).

Die eigentlich ausführenden Rollprogramme wurden vor VIDEOEDIT gelegt. Das ist vielleicht nicht elegant, war aber im Moment ganz bequem. Die Rollroutinen verändern jeweils die Pointerbasen FROM und TO, um an der richtigen Bildschirmcke anfassend zu können. Der Rest ist Zählung und Transport. PETROL bildet mit dem VIDEO-DRIVER und VIDEOEDIT eine Einheit.

Es ergeben sich lustige Spezialeffekte, wenn man die Zahl des Versatzes und auch seine Richtung innerhalb der vier Möglichkeiten von einem Zufallszahlengenerator abhängig macht. Durch Veränderung der Pointeradressen FROM und TO kann man zudem ein schräges Bildpumpen bewirken oder das Heraussprudeln eines Textes aus einer Bildschirmquelle. Viele weitere Möglichkeiten sind denkbar, auch das direkte Rollen des Bildschirms in den Adressen 8000 hex ff. Happy rolling now!

TAB=7045	UP	*=6825	BPL L10
PX=7048		LDY PY	RTS
PY=7049	L11	JSR INIT	RIGHT *=6900
INIT=7050		JSR FROMP	JSR INIT
FROMP=7064		LDX PX	LDX PX
TOP=7076		LDA (TO),Y	INC TO
FROMM=7088		PHA	BNE R1
TOM=7100	L12	LDA (FROM),Y	INC TO+1
FROM=176		STA (TO),Y	LDY PY
TO=178		JSR FROMP	LDA (FROM),Y
DOWN *=6773		JSR TOP	PHA
LDY PY		DEX	DEY
L9 JSR INIT		BNE L12	R2 LDA (FROM),Y
CLC		PLA	STA (TO),Y
LDA TO		STA (TO),Y	DEY
ADC #152		BPL L11	BPL R2
STA TO		RTS	INY
STA FROM			PLA
LDA TO+1	LEFT	*=6860	STA (FROM),Y
ADC #3		JSR INIT	JSR FROMP
STA TO+1		DEC TO	JSR TOP
STA FROM+1		BCS L1	DEX
JSR TOP		DEC TO+1	BPL R1
LDX PX	L1	LDX PX	RTS
LDA (TO),Y	L10	LDY #0	
PHA		LDA (FROM),Y	SUBROUTINES
L6 LDA (FROM),Y		PHA	*=7045
STA (TO),Y	L9A	INY	TAB .BYT 0
JSR FROMM		LDA (FROM),Y	.BYT 28
JSR TOM		STA (TO),Y	RI .BYT XX (UNUSED)
DEX		CPY PY	PX .BYT 24
BNE L6		BCC L9A	PY .BYT 39
PLA	L8	PLA	
STA (TO),Y		STA (FROM),Y	
DEY		JSR FROMP	
BPL L9		JSR TOP	
RTS		DEX	

65_{xx} MICRO MAG

INIT	*=7050	TOP	*=7076	TOM	*=7100
	CLD		CLC		SEC
	LDA #1		LDA TO		LDA TO
IN1	LDA TAB,X		ADC #40		SBC #40
	STA FROM,1		STA TO		STA TO
	STA TO,X		BCC T1		BCS T2
	DEX		INC TO+1	T2	DEC TO+1
	BPL IN1	T1	RTS		RTS
	RTS			MASTR	*=7112
					JSR 26XX (HIGH-LOW)
FROMP	*=7064	FROMM	*=7088		LDA #52
	CLC		SEC		STA 59409 DEZ.
	LDA FROM		LDA FROM		JSR 6957 DEZ.
	ADC #40		SBC #40		LDA #60
	STA FROM		STA FROM		STA 59409
	BCC F1		BCS F2		RTS
	INC FROM+1		DEC FROM+1		
F1	RTS	F2	RTS		

Vor Aufruf des Programmes MASTR als kleines Steuerprogramm POKEd man nach PX die zu rollende Zeilenzahl, nach PY die Spaltenzahl und ferner nach 7113 die Rollrichtung in die niedrige Adresse des JSR-Befehles. POKE7113,73 für DOWN. Die Maschine bildet daraus die Adresse \$1A75 usw.. Die Routine MASTR tastet den Bildschirm nach dem Rollen dunkel, bringt den Video-Speicher auf den Bildschirm und tastet wieder hell.(SYS7112).

Für vorstehende Maschinenprogramme gilt folgender dezimaler DUMP:

	0	1	2	3	4	5	6	7	8	9
677.	-	-	-	172	137	27	32	138	27	24
678.	165	178	105	152	133	178	133	176	165	179
679.	105	3	133	179	133	177	32	164	27	174
680.	136	27	177	178	72	177	176	145	178	32
681.	176	27	32	188	27	202	208	243	104	145
682.	178	136	16	208	96	172	137	27	32	138
683.	27	32	152	27	174	136	27	177	178	72
684.	177	176	145	178	32	152	27	32	164	27
685.	202	208	243	104	145	178	136	16	225	96
686.	32	138	27	198	178	176	2	198	179	174
687.	136	27	160	0	177	176	72	200	177	176
688.	145	178	204	137	27	144	246	104	145	176
689.	32	152	27	32	164	27	202	16	229	96
6900	32	138	27	174	136	27	230	178	208	2
6910	230	179	172	137	27	177	176	72	136	177
6920	176	145	178	136	16	249	200	104	145	176
693.	32	152	27	32	164	27	202	16	229	96

704.	-	-	-	-	-	0	28	0	24	39
705.	216	162	1	189	133	27	149	176	149	178
706.	202	16	246	96	24	165	176	105	40	133
707.	176	144	2	230	177	96	24	165	178	105
708.	40	133	178	144	2	230	179	96	56	165

65xx MICRO MAG

	0	1	2	3	4	5	6	7	8	9
709.	176	233	40	133	176	176	2	198	177	96
710.	56	165	178	233	40	133	178	176	2	198
711.	179	96	32	xx	26	169	52	141	17	232
7120	32	45	27	169	60	141	17	232	96	-

Ein aufrufendes BASIC-Programm könnte z.B. lauten:

```

1 PRINT "IN WELCHE RICHTUNG SOLL GEROLLT WERDEN?"
2 PRINT "NACH OBEN, UNTEN, RECHTS ODER LINKS?"
3 GET A$: IF A$ = " " THEN 3
4 IF A$ = "U" THEN POKE 7113, 117: GOTO 10
5 IF A$ = "O" THEN POKE 7113, 169: GOTO 10
6 IF A$ = "L" THEN POKE 7113, 204: GOTO 10
7 IF A$ = "R" THEN POKE 7113, 244: GOTO 10
8 GOTO 3
10 INPUT "WIE GROSS SOLL DIE ZEITSCHLEIFE SEIN?"; B
11 FOR I = 1 TO B: NEXT SYS 7112: GOTO 11

```

■ = REVERSE OFF
 □ = REVERSE ON
 ✕ = CLEAR SCREEN

Durch POKE 7046, 128 wird der Inhalt des Bildschirms direkt gerollt, also ohne Umweg über den besonderen Bildspeicher.

(Anmerkung des Herausgebers: Sein 16-jähriger Sohn hat sich mit PETROL in wenigen Tagen in die Assemblerprogrammierung eingearbeitet. Es kann also so schwer nicht sein!)

VIDEO-NEWS

VIDEO NEWS, MICRO-Shop Bodensee (Nedela). Das VIDEO-Plus ist jetzt mit komplettem Betriebssystem auch für BASIC lieferbar, 6845 CRT, 2 KROM für Programme, bis 6 KRAM auszubauen, Grafik mit 4096 Byte Display RAM für freie Zeichensätze. - Ein weiteres Programm zur AIM-Erweiterung wurde aufgenommen, das 65X1 mit und ohne Stromversorgung mit folgenden Interface-Erweiterungen: Echtzeituhr, batteriegepuffert (real time clock), Video-Ausgang mit CRT-Controller (TV moduliert/R170 Composite Video/Video-Sync), ein V24 (RS232C) Ausgang und zwei V24 (RS232C) Ein- und Ausgängen, 8 Bit parallel, Semigrafik, Groß- und Kleinschreibung, Editing.

MICRO TECHNOLOGY UNLIMITED: Visible Memory. Hal Chamberlins speziell für KIM und AIM entworfenes Grafik-Video-Interface wird in den USA für \$ 289 angeboten. 200 Bildzeilen, 320 Punkte pro Zeile, d.h. 8 k Bildspeicher (dynamisch) für 15.625 Zeilen/Sek, vertikal 60.1 Bildwechsel, 8 MHz-Bildpunktoszillator an die Ø2 des Systems angebunden (gated).

ERRATA: REMARK DELETER, Heft 8, S. 14. Herr Lehner macht auf die fehlerhafte Zeile 40 aufmerksam, die wie folgt lauten muß:

40 POKE 527, 145: POKE 528, 145: POKE 529, 145: POKE 530, 145: POKE 531, 13:
POKE 532, 13: POKE 525, 6

Einfacher kann man diese Zeile 40 wie folgt schreiben und dann je nach Zeilennummerierung wie folgt geben, z.B. "RUN 100":

40 POKE 527, 19: POKE 528, 13: POKE 529, 13: POKE 525, 3: END

65xx MICRO MAG

AUFBAU EINES BEFEHLES RESTORE LINE NUMBER FÜR PET2001

Ing. (grad.) Uwe Kornnagel, Lahnstraße 6, D6096 Raunheim/Main

Die Startadresse des nachfolgenden Programmes entspricht dezimal 7900:

```

1EDC  RES   JSR   $CCA4 ;Wertübergabe bei SYS in Floatingakku
1EDF      JSR   $D6D0 ;Umwandlung in INTEGER , INT auf 8,9
1EE2      LDA   $7A   ;Startadresse des BASIC-Programms laden
1EE4      LDX   $7B
1EE6      JSR   $C526 ;Zeile suchen
1EE9      BCC   ERR5  ;Line nicht vorhanden
1EEB      LDA   $AE   ;Eigentlicher RESTORE
1EED      SBC   # 1
1EEF      STA   $90
1EF1      LDA   $AF
1EF3      SBC   # 0
1EF5      STA   $91
1EF7      RTS
1EF8  ERR5  JMP   $C7DB ; UNDEF'D STATEMENT ERROR
  
```

Der Syntax für RESTORE Line # lautet: SYS(7900)nn.

Die Größe "nn" kann eine Variable oder Konstante sein.

Die Variablenübergabe bei SYS erfolgt mit dem Programm \$CCA4, der Wert wird im Akku # 1 (\$B0 - \$B5) abgelegt. Der SYS - Befehl hat den Vorteil, es werden am Ende des Maschinenprogramms keine Variablen ans Basic zurückgegeben werden müssen.

Als nächstes möchte ich die ERROR - Routine des PET erklären:

Sie beginnt auf der Adresse \$C359 die Codenummer des Fehlers muß, in das X-Register geschrieben werden. Es gibt 17 Interpretermeldungen, zu denen folgende Codenummern gehören.

X-Reg. Message

```

0      Next Without For Error
16     Syntax Error
22     Return Without Gosub Error
42     Out Of Data Error
53     Illegal Quantity Error
69     Overflow Error
82     Out Of Memory Error
95     Undef ' D Statement Error
112    Bad Subscript Error
125    Redim ' D Array Error
138    Division By Zero Error
154    Illegal Direct Error
168    Type Mismatch Error
  
```

181	String Too Long Error
196	Bad Data Error
204	Formular Too Complex Error
223	Can ' T Continue Error
237	Undef ' D Function Error

Als letztes Beispiel möchte ich aufzeigen, wie man im Maschinenprogramm einen Text ausgeben kann.

```
0826   TXT      .TEXT PET 2001
0884   .BYTE   00
0835   prt      LDA  # 58
```

```
0837           LDX # 3
0839           JSR  $CA27
0842           RTS
```

Das Programm \$CA27 druckt einen Text, dessen Anfangsadresse (LSB im Akku und MSB im Y-Reg.) steht. Der Text muß mit 00 abgeschlossen sein.

DATENVERBUND ZWISCHEN AIM 65 UND PET 2001

Diese ebenfalls von Herrn Kornnagel beschriebene Schaltung nebst Betriebssystem stellt in der Sicht des Autors einen ersten Versuch dar, die spezifischen Leistungsfähigkeiten der beiden Rechner zu kombinieren. Der Rechnerverbund wurde von ihm zu Studienzwecken betrieben, um insbesondere einen Drucker (den des AIM 65) am PET zu betreiben und um andererseits auch den AIM auf einen vorhandenen Bildschirm arbeiten zu lassen. (Anm. d. Hrsg.)

Der Datenverbund umfaßt drei Routinen, die einen ASCII-Austausch zwischen den Systemen AIM und PET erlauben. Mit der ersten Routine kann der AIM als Drucker über User Port an den PET angeschlossen werden. Schaltungstechnisch müssen die Datenleitungen PA0-PA7 der beiden Rechner miteinander verbunden werden. Ferner müssen die Kontroll-Leitungen PET CA1 mit AIM CA2 und PET CB2 mit AIM CA1 zusammengeschaltet werden. Eine Masseverbindung zwischen den Rechnern darf ebenfalls nicht vergessen werden.

Die AIM-Routine "AIM-Listener" wird durch die Funktionstaste "F1" aktiviert. Beim PET muß im BASIC-Programm die erste definierte Variable eine Stringvariable zur Aufnahme der Druckzeile sein. Es ist wichtig, daß diese Variable wirklich die erste ist, da das Maschinenprogramm nach dieser sucht. Deshalb der Vorschlag des Autors ein Programm zum Datenverbund immer mit folgender Zeile beginnen zu lassen:

```
Ø PR$=""
```

Das Ausdrucken aus dem PET erfolgt dann dort mit SYS(880). Das Maschinenprogramm setzt kein CR (Carriage Return) an das Ende des Textes. Soll ein CR mit ausgegeben werden, so lautet die Programmzeile im PET:

```
XXX PR$=.....+CHR$(13):SYS(880)
```

Die zweite Routine dient zum Parallelschalten des AIM-Keyboards mit dem des PET. Hierbei können alle Zeichen zwischen \$00 und \$7F an den PET übergeben werden. Diese Routine wird am AIM mit der Taste "F2" aufgerufen und kann mit "ESC" wieder verlassen werden.

65xx MICRO MAG

Am PET aktiviert man den Keyboard-Verbund mit SYS(850). Wenn dabei am AIM die Taste F2 noch nicht betätigt war oder wenn er mit RESET gestoppt wurde, bleibt beim PET nach dem Aufruf SYS(850) der Cursor weg. Durch Drücken der Taste F2 wird der Fehler aber wieder behoben. Es erscheinen dann lediglich einige "?" auf dem Bildschirm. Wenn die Routine mit ESC-Taste verlassen wird, tritt dieser Effekt nicht auf.

Eine weitere Möglichkeit bietet die dritte Routine. Mit ihrer Hilfe kann man ein Programm im Textbuffer des AIM entwickeln und es dann an den PET weitergeben. Der Text-Editor ist voll zu verwenden, er darf nur nicht den Programmbereich \$200-\$2FF berühren. Nun hat man eine elegante Methode, die fehlende UNLIST-Funktion des PET zu realisieren. Der PET muß nur mit SYS(850) auf Parallelbetrieb umgeschaltet sein. Die Übergabe des Textbuffers erfolgt dann mit der Taste "F3" am AIM. Am Ende der Übergabe springt der AIM die Funktion "F2" an. Die Parallelschaltung läßt sich durch SYS(870) wieder aufheben. Diese Betriebsart wird ebenfalls aufgehoben, wenn der PET irgendeine Kassettenroutine anspringt (dabei erfolgt Umschaltung des VIA).

PROGRAMMBESCHREIBUNG:

F1= PET TO AIM PRINTROUTINE

F2= AIM TO PET INPUTROUTINE. AIM + PET KEYBOARD

F3= AIM TO PET LISTROUTINE. AIM TEXTBUFFER IS SENT TO PET. AIM IS KEYBOARD

(K)*=010C

/03

010C 4C JMP 0266

USER FUNCTIONS

010F 4C JMP 0243

ON AIM 65

0112 4C JMP 0231

(K)*=0200

/59

0200 A9 LDA #0D

0238 A5 LDA E3

0202 20 JSR EEFC

023A 85 STA 00

0205 A9 LDA #7F

023C A5 LDA E4

0207 8D STA A003

023E 85 STA 01

020A 20 JSR E93C

0240 4C JMP 0200

020D F0 BEQ 020A

0243 A9 LDA #3C

F2 FUNCTION

020F C9 CMP #FF

0245 A2 LDX #E9

0211 F0 BEQ 020A

0247 20 JSR 024D

0213 48 PHA

024A 4C JMP 0200

0214 20 JSR EEFC

024D 8D STA 020B

0217 68 PLA

0250 BE STX 020C

0218 8D STA A001

0253 60 RTS

021B AD LDA A001

0254 A0 LDY #00

TEXTBUFFER OUT

021E 29 AND #80

0256 B1 LDA (00),Y

0220 F0 BEQ 021B

0258 48 PHA

0222 A9 LDA #00

0259 E6 INC 00

0224 8D STA A001

025B D0 BNE 025F

0227 AD LDA A001

025D E6 INC 01

022A 29 AND #80

025F 68 PLA

022C D0 BNE 0227

0260 F0 BEQ 0263

022E 4C JMP 020A

0262 60 RTS

0231 A9 LDA #54 F3 FUNCTION

0263 4C JMP 0243

0233 A2 LDX #02

0235 20 JSR 024D

0266 A9 LDA #00	F1 FUNCTION	23 0366 LDA #85	ADDRESS 870
0268 8D STA A003		24 0368 LDX #E6	
026B AD LDA A001		25 036A JMP 035B	
026E F0 BEQ 026B		26 036D BRK	
0270 20 JSR EEFC		27 036E BRK	
0273 A9 LDA #EE		28 036F BRK	
0275 8D STA A00C		29 0370 LDA #FF	ADDRESS 880
0278 A9 LDA #CC		30 0372 STA E843	
027A 8D STA A00C		31 0375 LDA #00	
027D AD LDA A001		32 0377 STA E841	
0280 D0 BNE 027D		33 037A LDY #02	
0282 F0 BEQ 026B		34 037C LDA (7C),Y	
0284 00 BRK		35 037E STA B2	
a PET PROGRAM DISASSEMBLED			
1 033A LDA #00	ADDRESS 826	36 0380 INY	
2 033C STA E841		37 0381 LDA (7C),Y	
3 033F LDA E841		38 0383 STA B0	
4 0342 BNE 0347		39 0385 INY	
5 0344 JMP E685		40 0386 LDA (7C),Y	
6 0347 PHA		41 0388 STA B1	
7 0348 LDA #FF	8 034A STA E841	42 038A LDY #00	
9 034D PLA		43 038C LDA (B0),Y	
10 034E JMP E72C		44 038E STA E841	
11 0351 NOP		45 0391 LDA E84D	
		46 0394 AND #02	
12 0352 LDA #80	ADDRESS 850	47 0396 BEQ 0391	
13 0354 STA E843	DDRA	48 0398 LDA #00	
14 0357 LDA #3A		49 039A STA E841	
15 0359 LDX #03		50 039D INY	
16 035B SEI		51 039E CPY B2	
17 035C STA 0219		52 03A0 BCC 038C	
18 035F STX 021A		53 03A2 RTS	
19 0362 CLI		54 03A3 BRK	
20 0363 RTS			
21 0364 NOP			
22 0365 NOP			

FLOPPY DISK FÜR AIM. Die amerikanische Compas Microsystems in Ames, Iowa, bietet für den AIM das sog. DAIM-System an (Vertrieb in Europa durch Erhardt + Jost Electronic, Rössligasse 2, CH-4450 Sissach, Tel. 061-98 30 30. Nach dem Handbuch besteht es aus Interfacekarte, Stromversorgung und bis zu zwei Minifloppy-Laufwerken. Das Betriebssystem wird über die Funktionstasten des AIM angesprochen. Es hat folgende Commands: Compress Disk, List File, Rename File, Recover File, Init Disk Zero Directory, List Directory, Delete File, List Room Left, Close File. Je Laufwerk können bis zu 60 Anwenderfiles mit zusammen 78 k Umfang unterhalten werden. Das Beschreiben der Disketten erfolgt mit der Anwenderfunktion USER. Die Zusammenarbeit mit Assembler und BASIC ist lt. Datenblatt gewährleistet, zur Vermeidung von Speicherkonflikten müssen allerdings die BASIC-Pointer ab Zelle 0073 (hex) auf 80400 eingerichtet werden, weil das DAIM seinen Puffer in 0300-03FF hat. Der hiesige Preis für das System ist noch nicht bekannt.

ASSEMBLER-LISTING FÜR DEN AIM 65

DIPL.-ING. HORST STEDER, TALSTR. 10 B, 6000 FRANKFURT 56

E: The AIM Assembler output listing is reformatted according to the program listing below. Better readability is the result.

Das Assembler-ROM im AIM 65 bietet zusammen mit dem Editor eine hervorragende Möglichkeit, problemlos auch längere Programme zu erstellen und zu editieren. Leider stellt der 20-stellige Drucker eine gravierende Limitierung dar. Die Dokumentation eines Programmes mittlerer Länge wird sehr unübersichtlich, besonders dann, wenn auch Kommentare mit aufgelistet werden sollen.

Da keine Dokumentation des Assemblers zur Verfügung steht, wurde vom Verfasser versucht, anhand von im Standard-Format vorhandener Kriterien ein reformatiertes Listing für ein TV-Display und einen Zeilendrucker zu erhalten. Das in zahlreichen Versuchen entwickelte Programm hat sich auf den folgenden Seiten selbst aufgelistet.

Das Kernprogramm ohne Initialisierungs- und Ausgaberroutine (Locations 8000-8032 bzw. 80E3-813D) benötigt 233 Bytes. Es müßte also möglich sein, unter Verwendung der 20mA-Schnittstelle und der im Monitor enthaltenen OUTTTY-Routine das Programm komplett in einer Page unterzubringen.

Das Programm wird beim Aufruf der Assembler-Sequenz mit LIST-OUT=U über die User Output Function aufgerufen. Das ausgedruckte Format entspricht nicht ganz den üblichen Listings: Label- und Opcodefeld sind miteinander vertauscht. Auf eine fortlaufende Zeilennummerierung wurde, obwohl leicht implementierbar, zugunsten eines längeren Kommentarfeldes verzichtet. Hier liegt noch ein Betätigungsfeld für Perfektionisten. Der Verfasser ist aus beruflichen Gründen nicht in der Lage, zuviel Zeit in dieses schöne "Noch"-Hobby zu investieren.

Die Variablen sind in der Zero-Page in einem Bereich untergebracht, der vom Assembler unter keinen Umständen benutzt wird. Der Adreßzähler konnte nach längerem Suchen in den Plätzen 32 und 33 (hex) der Zero-Page gefunden werden. Bei der Anwendung dieses Programmes gelten folgende zusätzlichen Regeln für die Erstellung des Quellenprogrammes:

- 1) Wenn ein reiner Kommentar, beginnend in Spalte 13, gewünscht wird, so muß dem Semikolon unmittelbar ein Punkt folgen. Dieser Punkt wird beim Listing wieder unterdrückt.
- 2) Bei Verwendung von ASCII-Zeichen als Operand muß das Zeichen oder Feld unbedingt dann mit einem "Quote" abgeschlossen werden, wenn in der gleichen Zeile ein Kommentar stehen soll (Beispiele in Loc. 80B1 und 80B8).
- 3) Bei der Adreßzuordnung von Symbolen mittels der "Equate-Directive" führt die Einfügung eines Zwischenraumes zwischen letztem Symbolcharacter und dem Gleichheitszeichen zu einem übersichtlicheren Ausdruck der Adressen in einer Spalte untereinander. Andernfalls erfolgt das Listing ohne Zwischenraum.
- 4) Bei Verwendung eines Matrixdruckers mit Zeilenspeicher wird die letzte Zeile des Assembler-Listings (ERRORS=xxxx) nicht mehr automatisch ausgedruckt, da das "CR" nicht mehr zur Ausführung gelangt, sondern nur noch die entsprechende Flag setzt. Der Verfasser verwendet zum "Nachschieben" ein Miniprogramm in der Zeropage, und zwar in den ersten 4 Plätzen: 20 CC 80 00, d.h. JSR CRLF, BRK.

Zu der vom Verfasser eingesetzten Output-Routine ist noch folgendes zu sagen: Der User Output Port A des VIA wird für das VIC-1 TV-Interface von ELTEC mit paralleler Ansteuerung (PA0-PA6 für Daten, PA7 für Strobe) verwendet. Hierfür wird auch die Delay-Routine in 811F-813D benötigt. Der Output Port B dient als Schnittstelle für den HP Interface-Bus, der den Normen IEEE 488 bzw. IEC-Bus entspricht. Dabei dienen PB0-PB6 als Datenleitungen, PB7, CB; und CB2 als Output bzw. Inputs für die Handshake-Signale DAV, NRFD und NDAC des Busprotokolls.

Beim Aufruf der User Function mit "U" erfolgt ein Prompt "U=", der mit "P" beantwortet wird, wenn TV-Display und Drucker angesprochen werden sollen. Beim Drücken einer anderen Taste wird nur das Display aktiviert. Mit dem TV-Display kann man nun bequem auch Errors beim Assemblieren listen, zumal der Assembler jederzeit mit der "Space"-Taste angehalten werden kann. Die meterlangen Papierschlangen des Thermodruckers müssen nicht mehr in Kauf genommen werden.

```

8177          PASS 1
8177          PASS 2
0000
0000          ; *****
0000          ; *
0000          ; *      *** T P A S M ***      *
0000          ; *      ASSEMBLER REFORMATTER  *
0000          ; *      FOR VIC-1 TV-DISPLAY AND *
0000          ; *      HP 9876A THERMOPRINTER  *
0000          ; *      CONTROLLER: AIM 65      *
0000          ; *
0000          ; * VERS. 2.3   H.STEDER 8/79 *
0000          ; *****
0000
0000          CNTR          =#00D0
0000          EQFLG        =#00D1
0000          CRFLG        =#00D2
0000          SAVB         =#00D3
0000          SAVX         =#00D4
0000          SMFLG        =#00D5
0000          ICNT         =#00D6
0000          PRIFLG       =#00D7
0000          EQUAL        =#E7D8
0000          INPUT        =#E95F
0000
0000          **=$010A
010A          0000        .WOR TPASM          ;VECTOR TO USER OUTPUT FUNCTION
010C
010C          **=$8000
8000          TPASM      B031        BCS TDSP          ;INITIALISATION:
8002                  A2FF        LDX #FF          ;DEF ALL BITS OF:
8004                  8E02A0      STX #A002          ;PBD AS OUTP TO 9876A
8007                  8E03A0      STX #A003          ;PAD AS OUTP TO TV-DSPL
800A                  E8          INX
800B                  86D7        STX PRIFLG          ;CLEAR PRINTER FLAG
800D                  A27F        LDX #7F
800F                  8E0EA0      STX #A00E          ;CLEAR INTERRUPT ENABLE REG.
8012                  20D8E7      JSR EQUAL
8015                  205FE9      JSR INPUT
8018                  C950        CMP #'P'          ;PRINTER?
801A                  D00E        BNE TV            ;DISPLAY ONLY
801C                  A950        LDA #50          ;SET PCR, CB1 & CB2 POS. EDGE INTRPT
801E                  8D0CA0      STA #A00C          ;CB1 FOR 'NRFD', CB2 FOR 'NDAC'

```

65xx MICRO MAG

```

8021      A201   LDX #1
8023      86D7   STX PRIFLG      ;SET PRINTER FLAG
8025      A90D   LDA #0D
8027      20F800 JSR DPR         ;RESET PRINTER
802A TV    A90C   LDA #0C
802C      201481 JSR DSPL       ;AND DISPLAY
802F      201F81 JSR DELAY1
8032      60     RTS
8033
8033      ; ** START OF FORMATTING ROUTINE **
8033
8033 TDSP    D8     CLD
8034      86D4   STX SAVX        ;SAVE X-REG
8036      A223   LDX #35        ;DEF START COL# FOR COMMENT FIELD
8038      86D6   STX ICNT
803A      68     PLA
803B      85D3   STA SAVA        ;SAVE ACCU
803D      297F   AND #7F        ;STRIP OFF PARITY BIT
803F      C90D   CMP #0D        ;'CR' ?
8041      D005   BNE N1         ;NO 'CR', CHECK FOR '='
8043      A201   LDX #1

8045      86D2   STX CRFLG      ;'CR', SET FLAG, DON'T EXECUTE
8047      60     RTS
8048
8048 N1      C93D   CMP #'='      ;FIRST '=' IN LINE?
804A      D01A   BNE N6
804C      A6D2   LDX CRFLG      ;IF CR-FLAG SET AT FIRST '='
804E      F00A   BEQ N2
8050      20CC80 JSR CRLF        ;THEN EXECUTE CR
8053      A201   LDX #1
8055      86D1   STX EQFLG      ;SET EQUAL FLAG
8057      4CC980 JMP R1
805A N2      A6D1   LDX EQFLG      ;SECOND '=' ?
805C      D003   BNE N3
805E      4CC380 JMP NEXT       ;NO, OUTPUT CHAR
8061 N3      46D1   LSR EQFLG      ;CLEAR EQUAL FLAG
8063      4CC980 JMP R1
8066 N6      A6D2   LDX CRFLG
8068      F03F   BEQ N10
806A      C93B   CMP #';'      ;FIRST CHAR AFTER 'CR' A SEMICOLON?
806C      D028   BNE N8
806E      46D2   LSR CRFLG      ;YES, CLEAR CR-FLAG
8070      A201   LDX #1
8072      86D5   STX SMFLG      ;AND SET SEMI-FLAG
8074      4CC980 JMP R1
8077 NC      A6D0   LDX CNTR
8079      E022   CPX #34        ;IF OPERAND FIELD EXCEEDS COL# 34
807B      9005   BCC SP1
807D      8A     TXA
807E      6902   ADC #2
8080      85D6   STA ICNT        ;BACK ACCORDINGLY
8082 SP1     A6D0   LDX CNTR
8084      E4D6   CPX ICNT        ;CURRENT POSITION=COL #35?
8086      F006   BEQ N7
8088      20C180 JSR SPACE      ;NO, INSERT SPACES UNTIL #=35
808B      4C8280 JMP SP1
808E N7      A93B   LDA #';'
8090      20C380 JSR NEXT       ;OUTPUT A SEMICOLON
8093      4CA180 JMP NN         ;AND FIRST CHAR OF COMMENT
8096 N8      A6D0   LDX CNTR
8098      E00C   CPX #12        ;CURRENT POS <=#12 ?
809A      B00A   BCS N9

```


65xx MICRO MAG

```

809C      46D2   LSR CRFLG      ;YES, CLEAR CR-FLAG AND
809E      206381 JSR COUNT      ;INSERT SPACES UNTIL POS #12
80A1 NN    A5D3   LDA SAVA
80A3      4CC380 JMP NEXT
80A6 N9    206D81 JSR RESET      ;START NEW LINE
80A9 N10   A5D3   LDA SAVA
80AB      A6D5   LDX SMFLG
80AD      F00E   BEQ N11
80AF      46D5   LSR SMFLG
80B1      C92E   CMP #'.'      ;IF A "." FOLLOWS A "; "
80B3      D0C2   BNE NC
80B5      206D81 JSR RESET      ;THEN START COMMENT FIELD AT COL# 13
80B8      A93B   LDA #' '
80BA      4CC380 JMP NEXT
80BD N11   C97D   CMP #$7D      ;ANY 'J' (FILLER CHAR)?
80BF      D002   BNE NEXT
80C1 SPACE A920   LDA #$20      ;YES, REPLACE WITH SPACE
80C3 NEXT  20E380 JSR PRTR
80C6      203381 JSR DELAY3
80C9 R1    A6D4   LDX SAVX
80CB      60     RTS
80CC
80CC CRLF  A90D   LDA #$0D
80CE      20E980 JSR PRT1
80D1      202981 JSR DELAY2

80D4      A90A   LDA #$0A
80D6      20E980 JSR PRT1
80D9      202981 JSR DELAY2
80DC      A900   LDA #0
80DE      85D0   STA CNTR
80E0      85D2   STA CRFLG
80E2      60     RTS
80E3
80E3      ; ** UTILITY SUBROUTINES **
80E3
80E3 PRTR  A6D0   LDX CNTR      ;OUTPUT TO DSPL/PRINTER
80E5      E04F   CPX #79      ;LINE FULL?
80E7      B033   BCS IN1
80E9 PRT1  A6D7   LDX PRIFLG    ;PRINTER FLAG SET?
80EB      F027   BEQ DSPL      ;NO, ONLY TV-DSPL
80ED      AA     TAX
80EE INT1  AD0DA0 LDA $A00D
80F1      2910   AND #$10
80F3      C910   CMP #$10      ;NRFD SET?
80F5      D0F7   BNE INT1
80F7      8A     TXA
80F8 DPR   48     PHA
80F9      49FF   EOR #$FF      ;INVERT CHAR TO HP-IB FORMAT
80FB      8D00A0 STA $A000      ;SEND TO PRINTER
80FE      297F   AND #$7F      ;PULL 'DAV' LOW
8100      8D00A0 STA $A000
8103      AA     TAX
8104 INT2  AD0DA0 LDA $A00D
8107      2908   AND #$08
8109      C908   CMP #$08      ;NDAC SET?
810B      D0F7   BNE INT2
810D      8A     TXA
810E      4980   EOR #$80      ;PULL 'DAV' HIGH
8110      8D00A0 STA $A000
8113      68     PLA
8114 DSPL  8D01A0 STA $A001      ;CHAR TO TV-DISPLAY
8117      0980   ORA #$80
8119      8D01A0 STA $A001      ;STROBE

```

65xx MICRO MAG

```

811C IN1      E6D0  INC CNTR
811E          60    RTS
811F
811F DELAY1   A97D  LDA #125      ;133 MSEC - DELAYS FOR TV-DSPLE
8121          8D97A4 STA $A497
8124 D1       2C85A4 BIT $A485
8127          10FB  BPL D1
8129 DELAY2   A97D  LDA #125      ;8.3 MSEC
812B          8D96A4 STA $A496
812E D2       2C85A4 BIT $A485
8131          10FB  BPL D2
8133 DELAY3   A97D  LDA #125      ;125 USEC
8135          8D94A4 STA $A494
8138 D3       2C85A4 BIT $A485
813B          10FB  BPL D3
813D          60    RTS
813E
813E ADDR5    A533  LDA #33        ;ADDR5 HI-BYTE
8140          204C81 JSR NUMA
8143          A532  LDA #32        ;ADDR5 LO-BYTE
8145          204C81 JSR NUMA
8148          20C180 JSR SPACE
814B          60    RTS
814C
814C NUMA     48     PHA            ;HEX TO 2 ASCII-CHARS
814D          4A    LSR A
814E          4A    LSR A
814F          4A    LSR A

8150          4A    LSR A
8151          205781 JSR NOUT
8154          68     PLA
8155          290F  AND #$F
8157 NOUT     18     CLC
8158          6930  ADC #$30
815A          C93A  CMP #$3A
815C          9002  BCC LT10
815E          6906  ADC #6
8160 LT10     40C380 JMP NEXT
8163
8163 COUNT     20C180 JSR SPACE
8166          A6D0  LDX CNTR
8168          E00D  CPX #13
816A          D0F7  BNE COUNT
816C          60    RTS
816D
816D RESET     20CC80 JSR CRLF      ;START NEW LINE
8170          203E81 JSR ADDR5
8173          206381 JSR COUNT
8176          60    RTS
8177
8177          .END
8177          ERRORS= 0000

```

KLEINANZEIGEN DER LESER

AIM 65 mit 20 KRAM, BASIC und VIDEO-Plus-Platine, alles im Gehäuse, zu verkaufen: Hans Jürgen Roeschen, Bremer Weg 7a, 2875 Ganderkesee 3.

AIM-Besitzer sucht Software-Tauschpartner. Hase, Tellstr. 6, 435 Recklinghausen, Tel. 02361-15691.

A S P - ADVANCED SUBROUTINE PACKAGE (TEIL 6)

Michael Zimmermann, Eberstädter Str. 170, 6102 Pfungstadt

E: *Concluding his series, Mr. Zimmermann covers I/O with tape, display and printer on the AIM 65.*

Inhaltsverzeichnis für den 6. Teil (Schluß)

- 6. *Unterprogramme der Ein- und Ausgabe*
 - 6.1 *Unterprogramme zur Bandbearbeitung*
 - 6.2 *Unterprogramm zur Ausgabe auf der Sichtanzeige (DISP)*
 - 6.3 *Unterprogramm zur Ausgabe auf dem Thermodrucker (PRT)*
 - 6.4 *Unterprogramm zum Lesen von alphanumerischen Strings über die Tastatur (READ)*
 - 6.5 *Unterprogramm zum Einlesen von gepackten numerischen Werten (NKBD)*
 - 7. *Abschluß und Ausklang*
- * * *
- 6. *Unterprogramme der Ein- und Ausgabe*

Die nachfolgenden Unterprogramme sollen das ASP abschließen. Sie dienen dem Verkehr mit den peripheren Einheiten des AIM, soweit diese in der Grundausstattung des Gerätes enthalten sind.

Da diese Programme eine Reihe von Monitorroutinen benutzen, beschränkt sich ihre Anwendung auf den AIM und den PC100. Ebenso sind Änderungen bei der Belegung der Page Zero berücksichtigt, die bereits in Heft 6 angedeutet wurden. Der durch das Vorbereitungsprogramm beanspruchte Bereich liegt an den Adressen 70 ff. (hex).

6.1 Unterprogramme zur Bandbearbeitung

Die nachfolgenden Programme zur Bandbearbeitung werden unter diesen Voraussetzungen betrieben:

1. Jede Lese- oder Schreiboperation beginnt mit einem neuen Block.
2. Für Eingabe und für Ausgabe ist jeweils nur eine Bandeinheit zulässig. Beide Bandeinheiten können also nicht gleichzeitig in der gleichen Betriebsart gefahren werden.
3. Jede Datei beginnt mit einem Block als Kennsatz, der in den ersten Stellen den Namen der Datei und sonst nichts außerdem enthält. Der erste Datensatz beginnt also mit dem folgenden Block.
4. Jede Datei endet mit 255 Zeichen hexa 14. Sobald dieses Zeichen an einem Blockanfang erkannt wird, ist das Ende der Datei erreicht.

Die Unterprogramme der Datenausgabe gliedern sich in:

- OPO Open for Output
- WRITE Write one Block
- CLO Close Output

65xx MICRO MAG

OPO benutzt zuerst das Monitorprogramm WHEREO, um zur Laufzeit eines Anwenderprogrammes Namen und Bandeinheit der Ausgabedatei zu spezifizieren. Anschließend wird zu dem Teil des Unterprogrammes verzweigt, in dem der Block geschrieben wird.

WRITE benötigt 3 Bytes als Parameter. Das erste enthält die Satzlänge, die folgenden 2 Bytes die Adresse des Ausgabepuffers. Im Unterprogramm wird Zeichen um Zeichen geladen und dem Monitorprogramm TOBYTE übergeben, bis der Satzlängenzähler erreicht ist. Darauf wird der Block geschrieben.

CLO wird ebenso wie OPO ohne Parameter aufgerufen. Es schreibt 255-mal das Hexazeichen 14 und komplettiert dann den letzten Block.

Für die Dateneingabe sind folgende Unterprogramme enthalten:

OPI Open for Input

READ Read a Block

OPI wird ohne Parameter benutzt und übergibt mit Hilfe des Monitorprogrammes WHEREI zur Laufzeit Namen und Nummer der Bandeinheit.

READ erfordert dieselben Parameter wie WRITE, die Länge des Blockes und die Pufferadresse, und liest einen Satz. - Wurde der EOF-Indikator erkannt (End Of File), so wird das Carry-Flag gesetzt. Andernfalls kehrt READ mit Carry Clear an das aufrufende Programm zurück (normal).

Einzelheiten der Programme sind den dokumentierten Listings zu entnehmen. Vor einer Benutzung sollte der Anwender sich auch mit den aufgerufenen Monitorroutinen vertraut machen.

0200 20 JSR E871	OPO	JSR WHEREO	GET WHERE TO WRITE
0203 18 CLC		CLC	
0204 90 BCC 0227		BCC BLKFIL	
0206 A0 LDY #03	WRITE	LDY #S03	GET PARAMETERS
0208 20 JSR OFE2		JSR VORB+2	
020B A0 LDY #00		LDY #S00	SET CHARACTER COUNT
020D B1 LDA (73),Y	L10	LDA (BUF),Y	GET CHARACTER
020F 20 JSR F18B		JSR TOBYTE	AND WRITE IT
0212 C8 INY		INY	INCREASE COUNT
0213 C4 CPY 72		CPY CNT	AND CHECK
0215 D0 BNE 020D		BNE L10	REPEAT
0217 F0 BEQ 0227		BEQ BLKFIL	OR FILL BLOCK
0219 A2 LDX #FF	CLO	LDX #SFF	SET CHARACTER COUNT
021B A9 LDA #14	L20	LDA #S14	LOAD EOF-INDICATOR
021D 20 JSR F18B		JSR TOBYTE	AND WRITE IT
0220 CA DEX		DEX	UNTIL DONE
0221 D0 BNE 021B		BNE L20	
0223 EA NOP		NOP	
0224 EA NOP		NOP	
0225 EA NOP		NOP	
0226 EA NOP		NOP	
0227 A9 LDA #4F	BLKFIL	LDA #S4F	LOAD BLOCK CHARACTER COUNT
0229 8D STA A437		STA TAPTR2	AND STORE TO COUNTER
022C 20 JSR F18B		JSR TOBYTE	AND WRITE WITH BLOCK FULL
022F 60 RTS		RTS	RETURN
0230 20 JSR E848	OPI	JSR WHEREI	GET WHERE TO READ
0233 18 CLC		CLC	
0234 90 BCC 025F		BCC REAEND	

0236 A0 LDY #03	READ	LDY #03	GET PARAMETERS
0238 20 JSR 0FE2		JSR VORB+2	
023B A9 LDA #50		LDA #50	LOAD BLOCK CHARACTER COUNT
023D 8D STA A436		STA TAPTR	& STORE IN COUNTER
0240 A9 LDA #00		LDA #00	SET BUFFER POINTER
0242 8D STA 0115		STA BLK	MAKE VOID BLOCK COUNT
0245 20 JSR ED3B		JSR TIBYTE	INPUT FIRST CHARACTER
0248 C9 CMP #14		CMP #14	EOF-INDICATOR?
024A D0 BNE 024F		BNE L30	NO, BRANCH
024C 38 SEC		SEC	SET CARRY
024D B0 BCS 025F		BCS REAEND	AND BRANCH TO END
024F A0 LDY #00	L30	LDY #00	LOAD BUFFER POINTER
0251 F0 BEQ 0256		BEQ L40	AND BRANCH TO STORE
0253 20 JSR ED3B	L35	JSR TIBYTE	READ SUBSEQUENT CHARACTER
0256 91 STA (73),Y	L40	STA (BUF),Y	AND STORE IN BUFFER
0258 C8 INY		INY	INCREASE POINTER
0259 C4 CPY 72		CPY CNT	& CHECK IT
025B D0 BNE 0253		BNE L35	AND REPEAT
025D 18 CLC		CLC OR DONE	
025F EA NOP		NOP	
0260 60 RTS		RTS	RETURN

6.2 Unterprogramm zur Ausgabe auf der Sichtanzeige (DISP)

Dieses Unterprogramm dient der Ausgabe von alphanumerischen Werten auf der LED-Sichtanzeige des AIM. Der Aufruf erfolgt so:

```

JSR    DISP
        .BYTE  Pufferlänge
        .WORD  Pufferadresse

```

Der Puffer ist durch den 2. Parameter angesprochen. Sein Inhalt wird auf dem Display ausgegeben. Eine Beschränkung auf die Kapazität der Anzeige erfolgt nicht automatisch, sondern unterliegt der Kontrolle durch den Anwender.

Ebenso hat der Benutzer die Kontrollzeichen 0D und 8D (hex) selbst zu verwalten und in den Puffer mit einzugeben und bei der Anzahl der auszugebenden Zeichen zu berücksichtigen.

0200 A2 LDY #03	DISP	LDY #03	GET PARAMETERS
0202 20 JSR 0FE2		JSR VORB+2	ASP
0205 20 JSR EB44		JSR CLR	MONITOR
0208 A0 LDY #00		LDY #00	SET COUNTER
020A A6 LDX 72		LDX CNT	
020C B1 LDA (73),Y	L10	LDA (BUF),Y	GET CHARACTER
020E 20 JSR EF05		JSR OUTDIS	AND DISPLAY IT
0211 C8 INY		INY	INCREASE POINTER
0212 CA DEX		DEX	DECREASE COUNTER
0213 D0 BNE 020C		BNE L10	REPEAT
0215 60 RTS		RTS	RETURN WHEN DONE

6.3 Unterprogramm zur Ausgabe auf dem Thermodrucker (PRT)

Dieses Programm zur Ausgabe von Werten auf dem Thermodrucker gleicht dem vorstehenden DISP, mit Ausnahme der angesprochenen Monitorroutine.

65xx MICRO MAG

Auf eine gesonderte Beschreibung wird daher verzichtet. Man beachte allerdings, daß ein Ausdruck nur dann stattfindet, wenn das Printer-Flag auf 'EIN' geschaltet ist.

0200 A0 LDY #03	PRT	LDY #\$03	GET PARAMETERS
0202 20 JSR OFE2		JSR VORB+2	
0205 20 JSR EB44		JSR CLR	
0208 A0 LDY #00		LDY #\$00	
020A A6 LDX 72		LDX CNT	
020C B1 LDA (73),Y	L10	LDA (BUF),Y	
020E 20 JSR F000		JSR OUTPRI	
0211 C8 INY		INY	
0212 CA DEX		DEX	
0213 D0 BNE 020C		BNE L10	
0215 60 RTS		RTS	

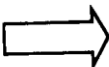
6.4 Unterprogramm zum Lesen von alphanumerischen Strings (über die Tastatur READ)

Die Einlesung der von der Tastatur her eingegebenen Strings erfolgt in einen Eingabepuffer. Ein String wird entweder durch Erreichen der vorgesehenen Länge oder durch Eingabe eines CR (Carrage Return) abgeschlossen. Der Aufruf erfolgt so:

```
JSR    READ
.BYTE  maximale Länge des Strings
.WORD  Adresse des Eingabepuffers
```

Das Programm übernimmt zunächst die Parameter, füllt dann den Eingabepuffer mit Blanks. Danach erfolgt die Einlesung bis entweder ein CR-Zeichen erkannt wird oder der Pointer auf maximale Länge zeigt.

0200 A0 LDY #03	READ	LDY #\$03	GET PARAMETERS
0202 20 JSR OFE2		JSR VORB+2	
0205 A9 LDA #20	L10	LDA #\$20	BLANK CHARACTER
0207 91 STA (73),Y		STA (BUF),Y	CLEAR BUFFER
0209 88 DEY		DEY	
020A D0 BNE 0205		BNE L10	REPEAT
020C A0 LDY #00		LDY #\$00	LOAD POINTER
020E 20 JSR E973	L20	JSR REDOUT	GET CHARACTER
0211 C9 CMP #0D		CMP #\$0D	IS IT A "CR"?
0213 F0 BEQ 021C		BEQ REAEND	YES, WE ARE READY
0215 91 STA (73),Y		STA (BUF),Y	NO, STORE TO BUFFER
0217 C8 INY		INY	INCREASE POINTER
0218 C4 CPY 72		CPY CNT	AND COMPARE
021A D0 BNE 020E		BNE L20	NOT YET FILLED
021C 60 RTS	REAEND	RTS	FULL, READY



KLEINANZEIGEN DER LESER

VERKAUFE AIM 65, BASIC, ASSEMBLER, 4 KRAM, ZEISSLER-GEHÄUSE,
NETZTEIL UND SOFTWARE DM 1.500. I. DRUST, LANDWEHRSTRASSE 5,
6100 DARMSTADT.

6.5 Unterprogramm zum Einlesen von gepackten numerischen Werten (NKBD)

Neben alphanumerischen Daten werden vielfach auch rein numerische Werte eingegeben, die für eine spätere Verarbeitung gleich in gepackter Form abgespeichert werden sollen, ohne daß hierfür ein besonderer Arbeitsgang erforderlich ist. Das diese Aufgaben erfüllende Programm wird wie folgt aufgerufen:

```
JSR    NKBD
.BYTE  Länge in Bytes
.WORD  Adresse des Puffers für die gepackten Daten
```

Das Unterprogramm übernimmt zunächst die Parameter und löscht dann den Zwischenspeicher in Page Zero. Danach werden Zeichen eingelesen und auf diverse Merkmale geprüft.

Ist das Zeichen ein CR, so wird die Eingabe abgeschlossen und der Zwischenspeicher in den Eingabepuffer übertragen. Wurde ein Plus- oder ein Minuszeicheneingegeben, so wird ein entsprechender Schalter in Page Zero gesetzt. Bei Eingabe eines nichtnumerischen Zeichens wird dieses übergangen, es werden also lediglich numerische Zeichen von rechts nach links in den Puffer in der Zeropage eingeschoben. Nach einem CR-Zeichen wird der Pufferinhalt unter Berücksichtigung von Länge und Vorzeichenschalter in den Eingabebereich gepackt und vorzeichengerecht übernommen.

0200 A0 LDY #03	NKBD	LDY #03	GET PARAMETERS
0202 20 JSR OFE2		JSR VORB+2	
0205 20 JSR EB44		JSR CLR	CLEAR PAGE ZERO
0208 A4 LDY 72		LDY CNT	LOAD COUNT
020A A9 LDA #00		LDA #00	CLEAR SIGN
020C 85 STA 71		STA SIGN	
020E 20 JSR E973	L10	JSR REDOUT	GET CHARACTER
0211 C9 CMP #0D		CMP #0D	CR ?
0213 F0 BEQ 0244		BEQ LCR	
0215 C9 CMP #2B		CMP #2B	+ ?
0217 D0 BNE 021F		BNE LMI	
0219 A9 LDA #00		LDA #00	CLEAR SIGN IF PLUS
021B 85 STA 71		STA SIGN	
021D F0 BEQ 020E		BEQ L10	BRANCH ALWAYS
021F C9 CMP #2D	LMI	CMP #2D	'-' ?
0221 D0 BNE 0229		BNE LSO	SET SIGN IF MINUS
0223 A9 LDA #01		LDA #01	
0225 85 STA 71		STA SIGN	
0227 D0 BNE 020E		BNE L10	BRANCH ALWAYS
0229 C9 CMP #30	LSO	CMP #30	LESS ZERO ?
022B 90 BCC 020E		BCC L10	SKIP CHARACTER IF LESS
022D C9 CMP #3A		CMP #3A	GREATER 9 ?
022F B0 BCS 020E		BCS L10	SKIP CHAR IF GREATER
0231 2A ROL .A		ROL	ROLLNUMERIC VALUE INTO
0232 2A ROL .A		ROL	LEFT HALFBYTE
0233 2A ROL .A		ROL	
0234 2A ROL .A		ROL	
0235 A0 LDY #04		LDY #04	SET FOR ROLL IN BUFFER
0237 A2 LDX #20	L15	LDX #20	
0239 2A ROL .A	L20	ROL	AND ROLL ONE BIT
023A 36 ROL 40,X		ROL BUF,X	A TIME, ALL 4 BITS

65xx MICRO MAG

023C CA DEX		DEX	
023D D0 BNE 023A		BNE L20	
023F 88 DEY		DEY	
0240 D0 BNE 0237		BNE L15	
0242 F0 BEQ 020E		BEQ L10	ROLLING DONE, GET NEXT CHAR.
0244 A2 LDX #60	LCR	LDX #360	MOVE BUFFER OUT
0246 A4 LDY 72		LDY CNT	TAKING SIGN AND LENGTH
0248 20 JSR XXXX		JSR MNACH+2	
024B 60 RTS		RTS	

7. Abschluß und Ausklang

Der Verfasser darf sich bei seinen Lesern für das Interesse bedanken und der Hoffnung Ausdruck geben, daß die Unterprogramme einer Vielzahl von Anwendern nützlich sein werden. Etwaige Fehler mögen entschuldigt werden. Für viele der aufgezeigten Ansätze gibt es alternative Lösungen, die besser sein mögen. Der Herausgeber und auch der Verfasser werden sich über jede Anregung freuen und diese an einen größeren Leserkreis weitergeben.

* * *

Literaturhinweis: In Heft 19 der FUNKSCHAU vom 14. Sept. 1979 sind auf Seite 1125 ff. von Michael Konz erarbeitete 6502-Rechenroutinen mit hexadezimalen Dump und mit Gebrauchsanleitung, abgedruckt. Es können Gleitkommazahlen von 1×10^9 bis $9,999999 \times 10^9$ in den Grundrechenarten verarbeitet werden, die interne Genauigkeit beträgt 12 Stellen in der Mantisse. - Ein für wissenschaftliche Berechnungen sicher sehr nützlich Programmpaket und eine Ergänzung zu Herrn Zimmermanns ASP, das vorwiegend für kaufmännische Anwendungen konzipiert wurde. Inwieweit eine Vereinigung von Dienstleistungen aus beiden Quellen möglich ist und inwieweit die Zeropage gemeinsam zu belegen wäre, bleibt dem weiteren Studium vorbehalten. Auch hierzu wären Anregungen sehr willkommen.

Die FUNKSCHAU bringt seit vielen Monaten regelmäßig Beiträge zur Mikroprozessortechnik, und zwar sowohl für die Hardware wie auch vollständige Programme. Der Schwerpunkt liegt auch hier beim 6502.

Besonders ist auch auf das Sonderheft der FUNKSCHAU hinzuweisen 'Hobby Computer 2', das ebenso sehr auf 6502 abgestellt ist und viele nützliche Informationen und Anregungen enthält.

SYNERTEC präsentierte auf der SYSTEMS in München (Sept. 79) das Micro Development Tool MDT100 mit folgenden Merkmalen: 6502 CPU, kompatibel zum Bus des EXORCisor, Tastatur mit Gehäuse und 54 Tasten, Interface für 2 Cassetten, EPROM-Programmer für 2716 und 2732, Motherboard mit vier Aufnahmen, seriell und paralleles Druckerinterface, RS232-Interface für 110-19200 Baud, 4 k Monitorprogramm, residenter Assembler/Editor/Lader in 8 k ROM, 8 k BASIC als Option, Video-Interface mit 6545 CRTIC. Hierfür gibt es auch eine europäische Version.

ERIC, C. REHNKE, Begründer der Zeitschrift 'KIM USER NOTES', jetzt '6502 USER NOTES', wird bei Rockwell International in Anaheim demnächst den newsletter INTERACTIVE herausbringen.

ALIGN FÜR ALPHA-SORT

Klaus Schuenemann, An der Schanz 2, 5000 Köln 60

Der Alpha-Sort aus Heft 4 (65xx MICRO MAG) sortiert gewissermaßen linksbündig. Um ihn auch für Zahlentabellen einsetzen zu können, habe ich nachfolgende Align-Routine geschrieben, welche Tastatureingaben auf gleiche Kommastelle bringt, bevor die Daten rechtsbündig in den Speicher übertragen werden.

Im Beispiel befindet sich der Eingabe-Stack in 02F0-02FF. Dabei enthält 02F0-02F2 eine Kennzahl, 02F3 einen Feld-Separator. Die eingegebene Zahl muß einen ASCII-Dezimalpunkt enthalten.

Nach dem Align befindet sich der Dezimalpunkt in 02FD, die Zahl hat dann 2 Kommastellen, abgeschnitten, nicht gerundet. Die Befehle CLV und BVC dienen der Vermeidung absoluter Jumps, um die Verschieblichkeit des Programmes zu verbessern. Abänderungen bezüglich Speicherplatz, Komma-stelle oder Feldbreite sollten leicht möglich sein.

089A	A9 F4	LDA #\$F4	INITIALISIERE
089C	8D A0 08	STA 08A0	
089F	AD FD 02	LDA 02FD	
08A2	C9 2E	CMP #\$2F	DEZIMALPUNKT ?
08A4	F0 10	BEQ 08B6	ALIGN
08A6	AD A0 08	LDA 08A0	
08A9	C9 FF	CMP #\$FF	ÜBERHAUPT DEZIMALPUNKT
08AB	D0 03	BNE 08B0	VORHANDEN ?
08AD	4C 34 08	JMP 0834	DIESE ROUTINE DRUCKT 'FALSCH'
08B0	EE A0 08	INC 08A0	
08B3	B8	CLV	
08B4	50 E9	BVC 089F	
08B6	AD A0 08	LDA 08A0	
08B9	C9 FD	CMP #\$FD	SOLLSTELLE FÜR DEZIMALPUNKT
08BB	F0 32	BEQ 08EF	KEIN ALIGN NOTIG
08BD	B0 16	BCS 08D5	LEFT SHIFT (STAMMT AUS KIMATH)
08BF	A2 0A	LDX #\$0A	RIGHT SHIFT
08C1	B0 F4 02	LDA 02F4,X	
08C4	9D F5 02	STA 02F5,X	
08C7	CA	DEX	
08C8	10 F7	BPL 08C1	
08CA	A9 20	LDA #\$20	ASCII 'SPACE'
08CC	8D F4 02	STA 02F4	
08CF	EE A0 08	INC 08A0	
08D2	B8	CLV	
08D3	50 E1	BVC 08B6	
08D5	A2 00	LDX #\$00	LEFT SHIFT
08D7	BD F5 02	LDA 02F5,X	
08DA	9D F4 02	STA 02F4,X	
08DD	E8	INX	
08DE	E0 0A	CPX #\$0A	
08E0	90 F5	BCC 08D7	
08E2	F0 F3	BEQ 08D7	
08E4	A9 30	LDA #\$30	ASCII '0'
08E6	8D FF 02	STA 02FF	
08E9	CE A0 08	DEC 08A0	

08EC	B8	CLV
08ED	50 C7	BVC 08B6
08EF	60	RTS

Fortsetzung im Kästchen

K O O R D

Peter Zechner, Halskestraße 11, 8000 München 70

Das hier vorgelegte Programm bringt einen oder mehrere Werte im Bereich von 1 bis 100 über eine Druckerlinie (-zeile) am AIM-Printer zum Ausdruck. Es wurde durch die AIMPLLOT-Routine des Herausgebers (Heft 6 dieser Zeitschrift) angeregt. Die Unterprogramme eignen sich für die Darstellung von mathematischen Funktionswerten ebenso wie für den Ausdruck von Bildern.

Es werden keine Speicherplätze außerhalb 0E10-0FFF benutzt, dadurch entsteht keine Kollision mit einem nicht allzu langen BASIC-Programm. Wir geben für das Maschinenprogramm zwei Darstellungen, einmal als Disassembler-Ausdruck und zweitens als Programmzeugung durch READ- und DATA Statements in einem BASIC-Programm. Es wird ein 10 Linien tiefer Bildabschnitt verwendet, der nach Wunsch vergrößert werden könnte.

Hinsichtlich der Erklärung und Dokumentation wird im wesentlichen auf das o.e. Programm AIMPLLOT verwiesen und natürlich auch auf das Listing im Monitorprogramm. Für die Routinen besteht folgende Dreiteilung:

1. Druckzeilenspeicher löschen. Der für den Ausdruck benutzte Speicherbereich wird auf Null gesetzt.
2. Druckpunkt setzen. Setzt den im Akkumulator übergebenen Wert in den Druckspeicher, ohne vorher dort eingeschriebene Druckpunkte zu zerstören. Kann pro Zeile beliebig oft aufgerufen werden.
3. Ausdrucken. Bringt die in der Druckzeile gespeicherten Werte zum Ausdruck. Für die im Druckspeicher abgelegten Daten wurde eine sequentielle Form gewählt. Anders also als in der AIM-Originalroutine werden die Druckpunkte für eine Linie auf 20 Speicherplätze und nicht auf 100 abgelegt. Dadurch ergibt sich die Notwendigkeit, den Speicher nach jeder Linie wieder zu löschen. Dieses Verfahren ist für eine Bildspeicherung recht einfach, und so kann ein 100x10-Bildstreifen noch bequem gespeichert und wieder ausgelesen werden.

Sofern man die nachfolgenden Maschinenprogramme unter I-Command oder Assembler erzeugt, können im beispielhaften BASIC-Programm die Zeile 9 und die DATA-Statements entfallen.

EINE ZEILE

0E10 2C BIT A411 ↑	0E2D 20 JSR 0E40	0E42 8D STA A801 ↑	DIE PFEILE
0E13 10 BPL 0E3F	0E30 20 JSR 0E40	0E45 AD LDA A80D ↑	ZIELEN AUF
0E15 20 JSR F0CB ↑	0E33 AD LDA A477 ↑	0E48 29 AND #02	HARDWARE-
0E18 20 JSR 0E76	0E36 C9 CMP #0A	0E4A F0 BEQ 0E45	UND MO-
0E1B A9 LDA #C1	0E38 90 BCC 0E2D	0E4C AD LDA A80C ↑	NITOR-
0E1D 8D STA A80C ↑	0E3A A9 LDA #E1	0E4F 49 EOR #01	ADRESSEN
0E20 20 JSR FFA0 ↑	0E3C 8D STA A80C ↑	0E51 8D STA A80C ↑	
0E23 D0 BNE 0E2D	0E3F 60 RTS	0E54 EE INC A477 ↑	
0E25 20 JSR FFA0 ↑	SUBRO	0E57 AD LDA A479 ↑	
0E28 D0 BNE 0E2D		0E5A OD ORA A800 ↑	
0E2A 4C JMP F079 ↑	0E40 A9 LDA #00	0E5D 8D STA A800 ↑	

0E60 AD LDA A478 ↑	0E8F 8D STA A478 ↑	PUNKT SETZEN
0E63 8D STA A801 ↑	0E92 D0 BNE 0E9D	0EB5 A2 LDX #00
0E66 A9 LDA #A4	0E94 AD LDA A47B ↑	0EB7 A9 LDA #8C
0E68 8D STA A808 ↑	0E97 0D ORA A479 ↑	0EB9 C9 CMP #05
0E6B A9 LDA #06	0E9A 8D STA A479 ↑	0EBB 90 BCC 0EC2
0E6D 8D STA A809 ↑	0E9D 0E ASL A47A ↑	0EBD E8 INX
0E70 20 JSR 0E76	0EA0 2E ROL A47B ↑	0EBE E9 SBC #05
0E73 4C JMP FOBA ↑	0EA3 CA DEX	0EC0 D0 BNE 0EB9
DRUCKSPEICHER VORBER.	0EA4 CA DEX	0EC2 A8 TAY
0E76 A2 LDX #00	0EA5 10 BPL 0E7B	0EC3 A9 LDA #40
0E78 20 JSR F121 ↑	0EA7 4C JMP F118 ↑	0EC5 18 CLC
0E7B 18 CLC	LÖSCHEN	0EC6 6A ROR .A
0E7C 7E ROR OFEA,X	0EAA A2 LDX #14	0EC7 88 DEY
0E7F BD LDA OFEA,X	0EAC A9 LDA #00	0EC8 10 BPL 0EC6
0E82 2C BIT A47C ↑	0EAE 9D STA OFEA,X	0ECA 1D ORA OFEA,X
0E85 F0 BEQ 0E9D	0EB1 CA DEX	0ECD 9D STA OFEA,X
0E87 AD LDA A47A ↑	0EB2 D0 BNE 0EAE	0EDO 60 RTS
0E8A F0 BEQ 0E94	0EB4 60 RTS	
0E8C 0D ORA A478 ↑		

In BASIC hat ein Beispielprogramm einschl. Initialisierung des Maschinenprogramms folgendes Aussehen:

```

9 FOR SA=3600 TO 3793:READ A:POKE SA,A:NEXT
10 PRINT"VOM TONBAND ?"
11 GET A$:IF A$="" GOTO 11
12 IF A$="J" THEN GOSUB 902 London
13 FOR B=0 TO 100:GOSUB 900:A=50:GOSUB 902 Punkt setzen
14 IF B=50 GOTO 70
15 REM MATHROUTINEN
20 A=INT(SIN(B/10)*50)+50:GOSUB 902 Math.
28 A=B+40
30 GOSUB 902 Math.
39 GOSUB 910 Speicher
40 GOSUB 904 Print
50 NEXT
60 END

70 FOR A=1 TO 100:GOSUB 902:NEXT:GOTO 15
700 REM FOLGENDE DATA-STATEMENTS SIND AUF WENIGE ZEILEN KOMPRIMIERT
701 REM BEI EINGABE AIM-ZEILENLAENGE 60 BERUECKSICHTIGEN
800 DATA 44,17,164,16,42,32,203,240,32,118,14,169,193,141,12,168,32,160
802 DATA 255,208,8,32,160,255,208,3,76,121,240,32,64,14,32,64,14,173
806 DATA 119,164,201,10,144,243,169,225,141,12,168,96,169,0,141,1,168
808 DATA 173,13,168,41,2,240,249,173,12,168,73,1,141,12,168,238,119,164,173
811 DATA 121,164,13,0,168,141,0,168,173,120,164,141,1,168,169,164,141,8
814 DATA 168,169,6,141,9,168,32,118,14,76,186,240,162,0,32,33,241,24,126
818 DATA 234,15,189,234,15,44,124,164,240,22,173,122,164,240,8,13,120,164
820 DATA 141,120,164,208,9,173,123,164,13,121,164,141,121,164,14,122,164,46
824 DATA 123,164,202,202,16,212,76,24,241,162,20,169,0,157,234,15,220,208
826 DATA 250,96,162,0,169,0,201,5,144,5,232,233,5,208,247,168,169,64
830 DATA 24,106,136,16,252,29,234,15,157,234,15,96,0,0

899 REM CLEAR BD E8D löschen
900 POKE 4,176:POKE 5,14:Z=USR(0):RETURN
901 REM PUNKT BERECHNEN E88
902 POKE 4,181:POKE 5,14:POKE 3768 ,A:Z=USR(0):RETURN

```

65_{xx} MICRO MAG

```

10
903 REM PRINT ONE LINE E10 Eine Zeile
904 POKE 4,16:POKE 5,14:Z=USR(0):RETURN
910 REM SPEICHERPLATZ ANWEISEN
912 FOR P1=0 TO 20:A=PEEK(4074+P1)
914 SA=20*P1+3840:POKE SA,A:NEXT P1:P=P+1
916 IF P=11 THEN P=0:GOSUB 980
918 RETURN
920 REM VOM TONBAND DRUCKEN
921 GOSUB 960
922 FOR P1=0 TO 10:GOSUB 900:FOR P=0 TO 20:A=PEEK(3840+20*P1+P)
924 SA=4074+P:POKE SA,A:NEXT P
925 GOSUB 904:NEXT P1
926 PRINT"WEITER ?"
927 GET AS:IF AS="" GOTO 927
928 IF AS="J" GOTO 920
929 RETURN

960 POKE 42001,0:PRINT"WIEDERGABE DRUECKEN":GOSUB 992
962 POKE 42002,75:POKE 42036,2
964 POKE 4,164:P:POKE 5,227:X=USR(0):GOSUB 990 E3A4
966 GOSUB 900:GOSUB 904:POKE 42001,128:RETURN

980 POKE 42001,0:GOSUB 990:PRINT"AUFNAHME DRUECKEN!"
981 GOSUB 990
982 POKE 42010,0:POKE 42011,15:POKE 42012,234:POKE 42013,15
984 POKE 42003,75:POKE 42037,2:GOSUB 992
986 POKE 4,135:POKE 5,229:X=USR(0):GOSUB 990 E5D7
988 PRINT":POKE 42001,128:RETURN
990 POKE 4,203:POKE 5,230:X=USR(0):RETURN E6CB
992 POKE 41992,91:POKE 41993,16:POKE 42033,49:POKE 42034,49:RETURN

```

DAS ERSTE OFFENE 65_{xx} COMPUTERMEETING in Riedelbach bei Frankfurt am 13./14. Okt. 1979 wurde von etwa 25 Teilnehmern besucht, die z.T. weit angereist waren. Das überwiegende Interesse galt dem AIM 65, der sehr gelobt wurde. Aber auch für PET und KIM fanden Gruppen zusammen. Wie in Gorleben, so waren auch hier tüchtige BASIC-Tiefbohrer in Aktion. Schön war die Demonstration einer Spracherweiterung für den PET mit verbalen Commands und Parameterübergabe noch im SYS-Befehl. Hier tun sich neue Horizonte auf, über die wir auch Eingang der erwarteten Beiträge noch berichten werden. Auch an den Salzstöcken des AIM wird gebohrt, z.T. mit gleichem Bemühen. Nach dem Hörensagen wird man das E/A des AIM-BASIC verbessern und sicher auch Spracherweiterungen implementieren können. Der Gedankenaustausch war intensiv und für alle Teilnehmer nützlich, es haben sich weiterreichende Korrespondenzen gebildet. - Vielen Dank für die Teilnahme!

ASSEMBLER-WORKSHOPS AIM 65/PC100. Der erste vom Herausgeber zusammen mit Herrn Dohmann und Herrn Zimmermann in Friedrichsdorf bei Frankfurt abgehaltene Workshop hatte mit 15 Teilnehmern seine Sollstärke und wurde ein voller fachlicher Erfolg. Der nächste Wochenendworkshop findet vom 23.-25. November 1979 in Augsburg statt. Workshops an anderen Orten sollen 1980 folgen. Die technische Ausrüstung des Dozententeams wird weiter verbessert, die Didaktik ist überarbeitet und führt vom Editor über die Assemblersprache zur intensiven Interfaceprogrammierung. Für Teilnehmer mit Vorkenntnissen.

MAGNETBANDBETRIEB**VORBEMERKUNGEN**

Die große Zahl von Anrufen zu diesem Thema gibt Anlaß, noch einmal auf die Besonderheiten speziell beim AIM 65 und PC100 einzugehen. Die Aussagen lassen sich aber auch auf andere Systeme übertragen.

An den AIM können bis zu zwei handelsübliche Cassettenrecorder zum Zwecke der Programm- und Datenspeicherung angeschlossen werden. Für den zufriedenstellenden Gebrauch dieser preiswerten Möglichkeiten sind zahlreiche Gesichtspunkte zu beachten, die sich aus der Unterschiedlichkeit der vielen am Markt befindlichen Cassettenrecorder, aus dem Bandmaterial oder auch aus den Besonderheiten des Betriebsprogrammes ergeben.

DAS AUFZEICHNUNGSVERFAHREN

Die in Form von Bytes vorliegende digitale Information wird durch das Betriebsprogramm über das Cassetteninterface bitweise-seriell ausgegeben. Für eine logische 0 oder 1 werden dabei unterschiedliche Folgen von Tonfrequenzen als Rechteckimpulse gesendet. Die Dokumentation dazu findet sich in Anhang F des Anwenderhandbuches. Diese Signale gelangen auf den Schreibverstärker des Magnetbandgerätes. Die handelsüblichen Rekorder haben ihren Arbeitspunkt im linearen Bereich der Magnetisierungskurve (Hysteresis). Der Trend zu HiFi hat zu vielen Kennkurven der Aufnahme- und Wiedergabesysteme geführt. Es entsteht mit ihnen in jedem Falle aus dem angebotenen digitalen Signal ein analoges auf dem Magnetband, zusätzlich mit Fourier-Schwingungen behaftet. Auf dem Schirm eines Oszilloskopes sehen wir ein Alpenpanorama. Aus diesem Signal muß beim Wiedereinlesen ein digitales Signal zurückgewonnen werden. Das besorgt das AIM-Interface mit dem LM311 als Operationsverstärker. Der AIM will Zeitabstände zwischen den Phasenwechseln messen können. Es ist klar, daß der Eintritt dieser Phasenwechsel vom Arbeitspunkt des LM311 (Poti VR1) und ggfs. auch vom Gleichstromanteil oder auch von der Kennlinie des Audio-Verstärkers im Rekorder abhängig ist, sicher auch von der Signalstärke, die sich aus Bandmaterial und Aussteuerung bei der Aufnahme ergibt (der Autor ist kein Analogtechniker). - Unser Aufzeichnungsverfahren unterscheidet sich also von den digitalen, die in der Sättigung der Magnetisierung arbeiten und für die sich wahrscheinlich Zeiten zwischen den Phasenwechseln ergeben, die außerhalb des Toleranzbereiches liegen, die die Zeitkonstanten im Monitorprogramm zulassen (Routinen RDBIT, CKFREQ).

ANFORDERUNGEN AN DIE AUFNAHME- UND WIEDERGABETECHNIK

Wegen der zeitlichen Erfordernisse ist ein Laufwerk mit sehr guten Gleitlaufeigenschaften zu wählen. Diese Aussage gilt ebenso für Cassetten. Billige Cassetten laufen wegen schlechter 'innerer Führung' nur ungleichmäßig durch das Gerät, es entstehen Zeitdifferenzen für o.a. Phasenwechsel. Ein Wiederlesen ist oft nicht möglich. Zum Studium schraube man einmal eine billige Warenhauscassette, eine MAXELL und eine BASF-SM auf (Spezial Mechanik). Man wird dabei etwa feststellen: Ganz billige Cassetten haben keine Führungsrollen in den Ecken, das Bandmaterial wird über Kunststoff-Pins hinweggezogen, bzw. geschauert. Bei billigen Cassetten besteht die Seitenauskleidung gegen den Bandwickel aus einer Art rauhem Karteikartenkarton, der sicher keinen guten Durchzug gewährleistet. Hochwertige Cassetten haben hier immer eine gleitfreundige Kunststoff-Folie. Und Anlegezungen, die den Wickel gleichmäßig halten, findet man nur (nach den Beobachtungen) bei BASF-SM.

65xx MICRO MAG

Wegen der Mechanik läuft eine 60er Cassette gleichmäßiger und leichter als eine 90er. Es gibt einige Betreiber, die diese Beobachtung wie folgt auswerten: Man nehme ein hochwertiges Band, wickle davon etwa 10 m ab und verstaue den Wickel in einem Cassettengehäuse, das man zu gut einer Mark gekauft hat, dessen Band man entfernte und das meinetwegen eine Seitenauskleidung aus Karton hat. Bei der Kürze des Bandes soll die Gleichlaufseigenschaft nicht beeinträchtigt sein. Der Autor hat hierzu keine eigenen Erfahrungen, darf aber anmerken, daß der Hamming-Way, das schnelle Bandverfahren aus Heft 6 - mit Korrektur in Heft 8 - die Nagelprobe darstellt, die BASF-SM auf jeden Fall besteht, ohne Umwicklung.

Durch Verunreinigungen und auch Ungleichmäßigkeiten im Bandmaterial können einzelne Informationen verloren gehen. Das Leseprogramm wird mit Error-Meldung abgebrochen, die Information ab entsprechender Stelle ist verloren. Zur eigenen Sicherheit wird man daher eine stereophonische Datenaufzeichnung in zwei Tonbandspuren bevorzugen, zumindest aber eine Aufzeichnung zweimal hintereinander vollziehen.

Sehr aufschlußreich waren auch die Berichte des Herrn Dipl.-Ing. Horst Steder in Frankfurt beim offenen Computermeeting am 13./14. Okt. 1979 in Riedelbach. Er ist beruflich mit den hervorragenden Meßmöglichkeiten der Fa. Hewlett-Packard ausgerüstet und schaltet beim Lesen beide Stereophonische Spuren auf den AIM. Bei Bandfehlern kann der AIM meistens noch das richtige Signal differenzieren. Nach den Beobachtungen dort betreffen Band- und Lesefehler selten nur einzelne Bits, sondern meistens drop-outs von ca. 20 Bit. Dem ist auch nicht mit dem veröffentlichten Hamming-Way zu begegnen, der ja nur ein single-bit-repair vollzieht. Von der Fa. Dohmann wurde dazu berichtet, es gäbe auch dokumentierte Hamming-Codes, die weniger Redundanz als die des Autors beinhalten, zugleich aber auch solche längeren drop-outs reparieren könnten. Man wird sich damit beschäftigen müssen, soweit es sich um die Aufzeichnung unwiederbringlicher Daten handelt. - Eine andere Methode wäre wie folgt: Man bildet Blocks von 25 oder 30 Bytes, wendet den Hamming-Way an, aber sendet von allen Bytes zunächst das erste Bit, dann das zweite usw.. Dieses letztere Verfahren hätte allerdings wohl den Nachteil, daß man nicht punktgenau bis zu einer Adresse BIS aufsetzen kann - wegen der Blockung. Allerdings könnte man die mitgesendete Number of Bytes beim Lesen verwenden.

KOMPATIBILITÄT ZWISCHEN CASSETTENREKORDERN

Bei der Vielzahl der am Markt angebotenen Geräte besteht oft keine Austauschbarkeit von Aufzeichnungen zwischen den Geräten. Ursache hierfür dürften unterschiedliche Spurlage und Spaltbreite sowie Verkantungen der Köpfe sein. Aber auch unterschiedliche Eigenschaften der Schreib- und Leseverstärker. In einigen Fällen wird man mit geeigneten Meßmitteln nachjustieren können.

AUSSTEUERUNG BEI DER AUFNAHME

Je nach Gerät und Bandmaterial wird man bei der Aufnahme eine unterschiedliche Aussteuerung wählen müssen, um Wiederlesbarkeit zu gewährleisten. Die Verwendung einer automatischen Aussteuerung ist nicht zu empfehlen, man beraubt sich eines vielleicht wichtigen Freiheitsgrades. Die Aussteuerung zu 7/10 des Maximums (die Nadeln sind am Anschlag) hat sich in vielen Fällen bewährt. Beim Zurücklesen ergibt sich damit ein Pegel von etwa 100% am VU-Meter (Grenze zwischen schwarzem und rotem Feld). Der Einsatz von CrO2-Band (es ist rauher, Kopfverschleiß, Entmagnetisierung) und von Dolby ist nicht erforderlich.

EINSTELLUNG BEI DER WIEDERGABE

Die Vorspannung am Poti VR1 ist auf die Bedürfnisse einzustellen. Es mögen sich an einzelnen Geräten Unterschiede gegen den im Anwenderhandbuch genannten Spannungen ergeben. Ggfs. probiere man manuell den geeigneten Arbeitspunkt aus (für LM311). Nach den Beobachtungen des Autors bei der Entwicklung des Hamming-Way ist zu berichten: Die Arbeitspunkte des LM311 klaffen bei seinem eigenen Gerät für das AIM-Format und für den Hamming-Way auseinander, daher wurden hier statt des VR1 zwei feinfühligke Spindelpotis installiert, die einzeln aufgeschaltet werden, VR1 wurde ausgelötet. Andere Betreiber in der Leserschaft fanden einen übereinstimmenden Arbeitspunkt. Beim Justieren des LM311 verwendet man zunächst Magnetbänder, die nur mit dem regelmäßigen Muster (hex) AA beschrieben sind, später geht man zur Erkennung von SYNCs über (hex 16). Stellt man nun mit dem Oszilloskop auf die logisch richtigen Breiten der Rechteckimpulse hinter dem LM311 ein, so wird man oft feststellen, daß der AIM gleichwohl nicht lesen kann. Der Arbeitspunkt, ab dem gelesen werden kann, liegt etwas neben dem oszilloskopisch richtigen. Hier mögen kapazitive Effekte hineinspielen.

Aus der guten alten KIM-Zeit stammen folgende Empfehlungen: Ein Rekorder braucht ein gewisses 'warm up', "rast' ich, so rost' ich", die Lager müssen wieder laufen und ihre Betriebstemperatur haben. Vor Aufnahme des Magnetbandbetriebes lege man daher eine Warmlaufzeit. Bei sehr billigen Geräten kann darin auch heute noch ein Heilmittel liegen - für das Zeitverhalten innerhalb der zulässigen Toleranzen.

BESONDERHEITEN DES BETRIEBSPROGRAMMES BEI AUFNAHME UND WIEDERGABE

Der AIM erlaubt die Verwendung von 3 verschiedenen Formaten bei der Magnetbandaufzeichnung und -wiedergabe. Sofern man nicht den Datenaustausch mit anderen Prozessoren der Systemfamilie beabsichtigt, sollte man das sehr sichere AIM-Format verwenden, das durch die Initialisierung des Prozessors automatisch zum Einsatz kommt. In Zelle A408, TSPEED, steht dann der Wert C7. Für die beiden KIM-Formate muß man bei Aufnahme und Wiedergabe *manuell* die Werte 5A bzw. 5B in diese Speicherzelle einsetzen und ggfs. hinterher wieder durch C7 ersetzen.

Im AIM-Format wird die Information in geblockter Form auf das Magnetband gebracht (Abschnitt F des Anwenderhandbuches). Für normale Programmspeicherung und Aufzeichnungen mit LIST aus dem Editor sind keine Besonderheiten des Betriebssystems zu beachten.

Die Verwendung der ROM-Programme Assembler und BASIC verlangt besondere Aufmerksamkeit hinsichtlich des Bandbetriebes. Während des Einlesens findet bereits eine Datenverarbeitung statt, die Zeit beansprucht. Ehe diese Verarbeitung vollendet ist, mag das Magnetband schon über die Synchronisationszeichen und Satzetiketten des unmittelbar nachfolgenden Blockes hinweggelaufen sein. Die Einrastung auf die Synchronisation erfolgt dann erst beim übernächsten Satz, damit stimmt aber der Block Count (in wörtlich falscher Übersetzung: der Wohnblock-Graf) nicht mehr überein, es wird Error gemeldet, das Bandladen wird abgebrochen. Es muß also zwischen den Bandblöcken eine ausreichende Verarbeitungszeit eingeräumt werden. Das besorgt der Parameter GAP in Zelle A409. Bei der Initialisierung, beim cold start schreibt der AIM in diese Zelle 08 hinein. Mit dieser Bandlückensteuerung kann fast immer problemlos arbeiten, wenn man Maschinenprogramme abspeichert oder Texte aus dem Textbuffer des Editors. Aber schon der Anschluß eines mit nur 300 Baud betriebenen (Video-) Terminals

kann zu Störungen führen, die mit einer Vergrößerung des GAP zu beseitigen sind. Beim Einlesen von Quellentexten mit dem Assemblerprogramm oder beim Laden von BASIC-Programmen erfolgt ein Mehr von Datenverarbeitung. Zwischen den Magnetbandblöcken müssen daher größere Lücken sein, um dem Programm ausreichend Zeit für die Verarbeitung zu lassen. Für den Assembler liegt das notwendige GAP irgendwo bei hex 20, bei BASIC bei bis zu hex 80. Das Laden wird damit eine langweilige Angelegenheit. Man sollte daher den empfohlenen Weg beschreiten und die Rekorder fernsteuern (Motor an/aus). Ob man hierzu nun die empfohlenen Schaltungen verwendet oder ganz einfach ein Reed-Relais in Dip-Form nimmt, ist wohl Geschmackssache.

Unter Fernsteuerung ist das GAP auf die Charakteristik des Laufwerkes beim Abstoppen und beim Wiederanlaufen einzustellen. Das Band muß seine volle Betriebsgeschwindigkeit wieder erreicht haben, ehe die Kennung des nachfolgenden Blockes am Lesekopf erscheint. Für weitergehende Manipulationen am Rekorder (Kurzschluß der Motorwicklung beim Abstoppen und Entfernung entbehrlicher Kondensatoren) sei auf die Ausführungen des Herrn Helm hingewiesen (in diesem Heft).

In der Zusammenfassung: Schwierigkeiten im Magnetbandbetrieb resultieren meistens aus der Vernachlässigung der Bandlücke GAP. Weitere Ursachen können in der Einstellung des Poti VR1, in den Gleichlaufeigenschaften des Rekorders oder der Cassetten oder im Magnetisierungsgrad liegen. Wertvolle Aufzeichnungen sollte man entweder doppelt oder auf 2 Spuren vornehmen.

UMBAU EINES CASSETTENRECORDERS

Mathias Helm, Adolf-Ey-Straße 2a, 3392 Clausthal-Zellerfeld

Der Cassettenrekorder CT2183 (Quelle-Bestell-Nr. 004.905 6, Preis unter DM 100) wurde wie folgt modifiziert und an den AIM angeschlossen: Der AIM steuert über ein Reed-Relais (Völkner 'A-5') den Remote-Control-Eingang des Cassettenrecorders an (Buchse J-1, Stifte 6 u. 7 des Recorders). Dem Schaltplan des Recorders ist zu entnehmen, daß die Spannungsversorgung im Lieferzustand für das gesamte Gerät über die Fernbedienung geschaltet wird. Für ein zuverlässiges Arbeiten mit dem AIM mußte hierbei das GAP auf hex 24 eingestellt werden. Die Hauptplatine wurde daher gemäß nachfolgender Zeichnung anders verschaltet. Auf der Hauptplatine und derjenigen mit den beiden Schieberegler wurden je zwei Leiterbahnen aufgetrennt und ein Relais eingeklebt (1x Um, 6V Betriebsspannung, 2A Schaltleistung, z.B. Kuhnke Kartenrelais Typ RK-62 - bei Völkner). Die weitere Verdrahtung ist der Skizze zu entnehmen.

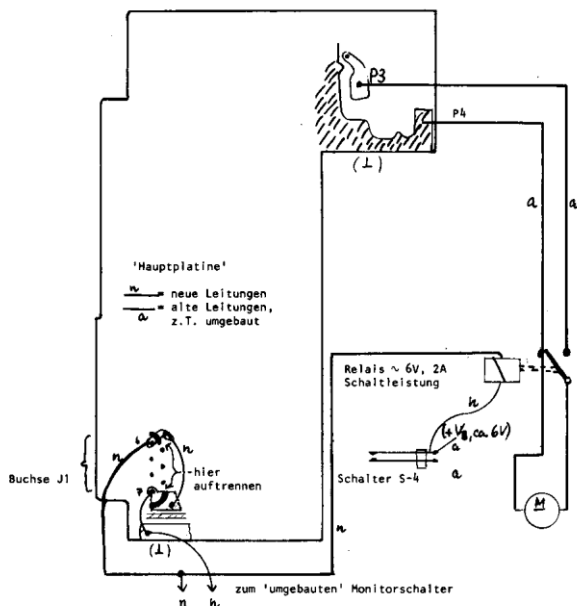
Nach Drücken einer Laufwerksfunktion des Recorders kann jetzt über die Fernbedienung der Motor eingeschaltet bzw. kurzgeschlossen werden (Kurzschlußbremse). Im Start-Stop-Betrieb kommt man nun mit einem GAP von 06 aus! Damit der Recorder auch ohne Einfluß durch die Fernsteuerung betrieben werden kann (u.a. mit GAP=03 als schnellster Betriebsart zum Laden in einen frisch initialisierten Editor), hat der Autor die drei ursprünglich zum Monitor-Schalter S-3 führenden Drähte abgelötet und danach zwei Leitungen zu verlegt, daß mit S-3 zwischen Fernsteuerung und Handbetrieb umgeschaltet werden kann. Der Regler TON erschien überflüssig und behindernd für hohe Frequenzen im NF-Signal. Auf seiner Platine kann man nach

Durchtrennen der Leiterbahnen, die ihn mit dem NF-Teil der Schaltung verbinden, z.B. das ehemals zu S-3 führende blaue Kabel über einen die maximale Lautstärke begrenzenden 100Ω-Widerstand am Schleifer des Tonreglers und das rote und das braune Kabel am oberen Abgriff anschließen und hat so eine Mithörmöglichkeit während der Aufnahme.

Weitere Verbindungen zwischen AIM und Recorder: Audio Out LO mit Punkt 1 bzw. 4 der Buchse J-1 (Rec), Audio In mit Punkt 3 bzw. 5 von J-1. Alle 3 Leitungen Audio In, Audio Out und Remote Control in einzeln abgeschirmten Kabeln.

Der Autor verwendet, weil preiswert vorhanden, C-90 Cassetten mit einem Eisenoxydband (LH von BASF). Änderungen an der Aussteuerautomatik mußten nicht vorgenommen werden.

Die Tonkopf-Justierungsschraube des Recorders war durch einen Lacktropfen ungenügend gesichert. Zur Justierung des Tonkopfes wurde daher eine in der üblichen Spurlage aufgenommene Musikkassette verwendet. Der Tonkopf konnte nach Gehör auf optimale Höhenlage eingestellt werden. Zum Erreichen der Justierungsschraube in der Wiedergabestellung muß man ggfs. ein kleines Loch in das Gehäuse treiben. Nach dieser Einstellung kamen die Empfehlungen in Anwenderhandbuch zum Einsatz: Für einige Minuten wurden SYNC-Zeichen auf das Band geschrieben. Mit diesem Band kann jederzeit die Einstellung des Poti VR1 justiert werden, ein Hilfsmittel auch wie folgt: Das Abhören von SYNCs oder eines Programmes bei Betriebsbeginn dient der Erkennung eines verstellten oder auch verschmutzten Tonkopfes. Es stellt sich dann eine 'dumpfe' Wiedergabe ein. Auch in falscher Spurlage aufgenommene Bänder konnten nach Einstellung des Tonkopfes wieder geladen werden. Der AIM des Autors hat bisher jedes Byte von Band lesen können.



AUSKUNFTSSYSTEM MIT AIM 65 (2)

Das in Heft 8 veröffentlichte Auskunftssystem konnte verbessert werden: Es ist jetzt das Absuchen einer Banddatei mit 3 Suchstrings logisch ODER möglich (Unterprogramm STRING). Diese dürfen zusammen 96 Zeichen haben, können aber in einzelnen unterschiedlich lang sein. Alternativen könnten natürlich vorgesehen werden. Weiterhin: Der Suchlauf erfaßt jetzt alle Strings in der Datei, die den Vorgaben entsprechen. Diese werden ausgedruckt. Der Suchlauf endet erst an der Endemarke @. Aus einer Ortsdatei kann man also z.B. alle Orte mit 'burg' ausdrucken lassen. Benutzung des Programmes mit den Funktionstasten F1-F3 wie in Heft 8. Das sehr schnelle Bandformat Hamming-Way konnte aus Zeitgründen noch nicht eingearbeitet werden.

Zahlreiche Leser baten um das verbesserte Programm. Häufig wurde auch der Wunsch geäußert, einmal ein Quellenprogramm in Assemblersprache zu präsentieren, weil da noch einige Unsicherheiten bestehen. Diesem Wunsch sei gern entsprochen. Der Autor muß allerdings darauf hinweisen, daß bei 4 k RAM wenig Platz für den ausgiebigen Gebrauch von Symbolen und Kommentaren frei bleibt. Man möge gleichwohl das Prinzip erkennen: Mit .OPT GEN werden die Tabellenbytes vollständig angeschrieben. Die Verschieblichkeit des Programmes ergibt sich aus der einen Deklaration START=\$C00.

Als sehr nützlich erweisen sich ferner die Erklärungen EDITOR=\$200 und FINIS=\$BFF. Diese Vektoren werden in der Tabelle TAD symbolisch angesprochen und mit dem Unterprogramm TRANS als EDITOR-Begrenzungen in die Zellen DF ff. gebracht. Mit einer einfachen Zuweisung erhält man somit z.B. eine neue Obergrenze FINIS. Das Gleiche gilt für die aus Tabelle TVEC zu transportierenden Anwendervektoren für page 1, Zellen 0108 ff.

Das Programm beginnt mit der Erklärung von Symbolen, es hätten wesentlich mehr sein sollen. In diesem Falle aber hätte man aus dem Quelltext 2 Magnetbandfiles machen müssen und sie mit der .FILE-Anweisung verbinden müssen. Dergleichen stellt kein Problem dar, ist aber etwas umständlicher in der Handlung.

;AUSKUNFTSSYSTEM	==0000	
.OPT GEN		
;SYMBOLE		
START=\$C00	==0000 START=\$C00	
EDITOR=\$200	==0000 EDITOR=\$200	
FINIS=\$BFF	==0000 FINIS=\$BFF	
PNTL=\$DF	==0000 PNTL=\$DF	
PNTH=\$E0	==0000 PNTH=\$E0	
FNAM=\$E8A2	==0000 FNAM=\$E8A2	
*=START	==0000 *=START	START
CLD	==0C00	D8
JSR FNAM		20A2E8
LDX #12		A20C
TR	==0C06 TR	
LDA TVEC,X		BD220C VEKTORENTTRANSPORT IN PAG 1
STA 010B,X		9D0801
DEX		CA
BPL TR		10F7
RENTER JSR TRANS	==0C0F RENTER	20690C INITIALISIERUNG D. EDITORS
LDA #0D		A90D
STA \$A412 INFLG		8D12A4
JMP \$F7D5		4CD5F7 SPRUNG INS EDITORPROGRAMM

Die Aufstellung enthält in der linken Spalte den aus dem Textbuffer stammenden Quelltext, in der Mitte die vom Assembler verwendeten Symbole und rechts davon das erzeugte Maschinenprogramm. ENDE D. SYMBOLDEKLARATION

TAD .BYT <EDITOR,>EDI ==0C1A
TOR.<EDITOR,>EDITOR

.BYT <EDITOR,>EDITOR,
<FINIS,>FINIS
TVEC .BYT \$C7,\$OF,<D
UMP,>DUMP,\$4C,< START
>START
.BYT \$4C,<DUMP,>DUMP
.BYT \$4C,<FIND,>FIND

DUMP LDA #\$2C ==0C2F DUMP
STA \$A419 COUNT
JSR \$EA13
JSR \$F8BC
LDA #\$54
STA \$A413 OUTFLG
JSR \$E56F ==0C3F
JMP LST02
LST01 JSR \$E907 ==0C45 LST01
JSR \$E790
BEQ LST3
LST02 JSR \$F727 ==0C4D LST02
JSR \$F709 < JSR \$F8E9
BCC LST01
LST03 JSR \$F73F ==0C58 LST03
JSR \$FF0D
PLA
PLA
PLA
LDA #4 ;STOP TAPE
STA \$A
JMP RENTER

TRANS LDX #7 ==0C69 TRANS
LDA TAD,X
STA PNTL,X
DEX
BPL TRANS+2
LDY #0
TYA
STA (PNTL),Y
RTS

STRING LDY #0 ==0C79 STRING
STR1 STA FC41+1. ==0C7B STR1
JSR \$E7BD
FC1 JSR \$E95F ==0C81 FC1
CMP \$0D
BNE FC3
CPY \$0
BNE FC3
FC2 JSR \$F709 ==0C8C FC2
JMP FC5
FC3 STA 0,Y ==0C92 FC3
CMP #\$0D
BEQ FC4

00 ADRESSVEKTOREN
02 FÜR INITIALI-
00 SIERUNG DES
02 EDITORS

0002
FF0B
C70F VEKTOREN FÜR USER
2F0C UND F1-F3
4C000C
4C2F0C
4CA80C

A92C
8D19A4
2013EA CRLOW
20BCF8 TOPNO
A954 TAPE
8D13A4
206FE5 DUMPTA, OPEN FILE
4C4D0C
2007E9 RCHEK
2090E7 DONE
F00B
2027F7 PLNE, DISPLAY CURR.LINE
2009F7 UPNO < 20E9F8
90ED
203FF7 PO3
200DFF PATC14
68
68
68
A904
8D00A8 PORT A
4C0F0C

A207 INIT EDITOR
BD1A0C
95DF
CA
10F8 SCHLEIFE
A000
98
91DF 00 ALS ERSTES EDITOR-BYTE
60

A000 SUCHSTRINGS ANLEGEN
8DA50C ADRESSMODIFIKATION
20BDE7 PROMPT
205FE9 RDRUB, LIES MIT RUBOUT
C90D CR?
D00A
C000
D006
2009F7 UPNO
4CC40C
990000 ZEICHENABLAG
C90D CR?
F00B

65_{xx} MICRO MAG

INX		C8	96 ZEICHEN?
CPY #560		C060	
BNE FC1		DOE3	NÄCHSTES ZEICHEN
JMP \$FA72		4C72FA	ERROR
FC4 JSR \$EA24	==0CA1 FC4	2024EA	
FC41 STY \$EB	==0CA4 FC41	84EB	
INX		C8	
RTS		60	
;SUCHEN MIT F3			
FIND LDA #<FORT	==0CA8 FIND	A935	BELEGUNG TASTE F1
STA \$010D		8D0D01	WIRD HIER ALLERDINGS
LDA #>FORT		A90D	NICHT VERWENDET, DA
STA \$010E		8D0E01	AUTOMAT. FORTSUCHEN
JSR FNAME		20A2E8	FILENAME
LDA #SEC		A9EC	
JSR STRING		20790C	3 STRINGS ANLEGEN
LDA #SED		A9 ED	
JSR STR1		207B0C	
LDA #SEE		A9EE	
JSR STR1		207B0C	
FC5 JSR TRANS	==0CC4 FC5	20690C	LEERE TEXT BUFFER
LDY #'T'		A054	TAPE FOR INPUT
STY \$A412 INFLAG		8C12A4	
JSR \$E32F		202FE3	LOADTA, LESE BLOCKS
LAD JSR INL	==0CCF LAD	206D0D	WIE IN INPU1
JSR UPNO		20D80C	NÄCHSTE ZEILE
JMP LAD		4CCF0C	BIS ENDE
UPNO LDY \$0	==0CD8 UPNO	A000	FOLGEZEILE EINSTELLEN
JSR \$F8E9		20E9F8	ATBOT, LETZTE ZEILE?
BCC UP1		900B	NEIN
LDX #\$FF		A2FF	STACKPOINTER DEFINIEREN
TXS		9A	
LDA #4		A904	STOP TAPE
STA \$A800		8D00A8	PORT
JMP PRUEF		4CED0C	TEXTBUFFER CHECKEN
UP1 JMP \$F713	==0CEA UP1	4C13F7	UP1, REST DES UPROGR.
PRUEF JSR \$F8BC	==0CED PRUEF	20BCF8	TOPNO, 1. ZEILE
LDY #0		A200	1. ZEICHEN DAS END OF
LDA (DF),Y		B1DF	FILE, EIN @?
CMP #\$40		C940	
BNE NEOF		D003	NEIN
JMP \$FA5C		4C5CFA	ENDERR, FERTIG, RÜCKGABE
NEOF LDY #0	==0CFB NEOF	A000	ABSUCHEN LOGISCH ODER
STY \$EB		84EB	
TOP JSR \$F8BC	==0CFF TOP	20 BCF8	TOPNO, AUF 1. ZEILE
TOP1 LDY \$EB	==0D02 TOP1	A4EB	
LDA SEC,Y		B9EC00	
LDX \$EB,Y		B6EB	
BEQ TOP2		F001	
INX		E8	
TOP2 JSR BLOCK	==0D0C TOP2	201B0D	
INX		C8	
INC \$EB		E6EB	
LDA \$EB		A5EB	
CMP #3		C903	3 SUCHBEGRIFFE?
BMI TOP		30E7	
JMP FC5 NACHLADEN		4CC40C	

BLOCK STX \$EF	==0D1B BLOCK	86EF	DRUCKEN
STA \$A429		8D29A4	SAVS
JSR PR		203B0D	
BCC PRINT		9001	
RTS		60	
PRINT LDA #\$80.	==0D26 PRINT	A980	PRINTER 'ON'
STA \$A411		8D11A4	PRIFLG
JSR \$F80F		200FF8	FCHA1
LDA #0		A900	& 'OFF'
STA \$A411		8D11A4	
PLA		68	STACK
PLA		68	
FORT JSR \$F709	==0D35 FORT	2009F7	UPNO, N. ZEILE
JMP TOP1		4C020D	WEITER SUCHEN
PR LDY #0	==0D38 PR	A000	WIE FC5, EDITOR
STY STY \$A415		8C15A4	
FC6 LDY \$A415	==0D40 FC6	AC15A4	CURPO2
LDX \$EF		A6EF	
CPX \$A429		EC29A4	
BEQ RET		F01C	
FC7 LDA (DF),Y	==0D4A FC7	B1DF	ENDE TEXTBUFFER?
BNE FC8		D002	
SEC		38	
RTS		60	
FC8 CMP #\$0D	==0D50 FC8	C90D	CR? NEUE ZEILE?
BEQ FC22		F013	
CMP 0,X		D500	VGL. M. STRING
BEQ FC9		F006	
INC \$A415		EE15A4	CURPO2
JMP FC6		4C400D	
FC9 INY	==0D5E FC9	C8	
INX		E8	
CPX \$A429		EC29A4	DONE?
BNE FC7		DOE5	
CLC		18	
RET RTS	==0D66 RET	60	
FC22 JSR \$F7 09	==0D67 FC22	2009F7	UPNO, N. ZEILE
JMP PR		4C3B0D	
INL JSR \$F8B6	==0D6D INL	2086F8	ENTSPRICHT INL IM EDITOR
LDY #0		A000	
STY \$E9		84E9	
JSR \$E7BD		20BDE7	
JSR \$EB44		2044EB	
INO2 JSR \$E993	==0D7A INO2	2093E9	
JSR \$FEF8		20F8FE	
INO2A CMP #\$0A	==0D80 INO2A	C90A	
BEQ INO2		F0F6	
CMP #\$0D		C90D	
BEQ INO3		F01B	
CPY #\$3C		C03C	
BCS INO3B		B008	
STA \$A438,Y		9938A4	
INX		C8	
CPY #60	==0D90	C03C	
BNE INO2		DOE6	
INO3B LDY #60	==0D94 INO3B	A03C	
LDA #1		A901	

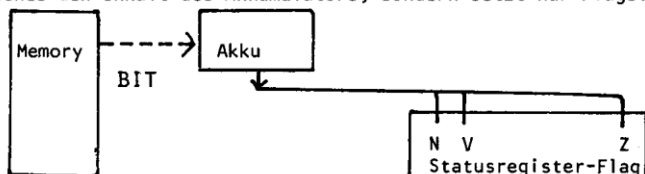
 FORTSETZUNG
 AUF SEITE 43


EIN LEITFADEN FÜR DIE PROGRAMMIERUNG (TEIL 4)

7.5 Der BIT-Befehl

Der letzte Abschnitt in den Heften 5 und 6 dieser Zeitschrift behandelte die logischen Befehle AND, OR und EOR. Viele Leser stehen dem BIT-Befehl etwas fremd gegenüber. Daher folgende Erklärungen:

Der BIT-Befehl ähnelt dem logischen UND, er zerstört allerdings nicht wie jenes den Inhalt des Akkumulators, sondern setzt nur Flags.



Der BIT-Befehl bewirkt also nur eine Statusanzeige, wie sie sich nach einem AND ergeben würde, insoweit gleicht er den Compare-Befehlen, die ja auch nur das Ergebnis einer 'gedachten' Subtraktion Akkumulator minus Operand in den Status eintragen. Wir machen auf dem AIM 65 einen Versuch:

```

0400 A9 LDA #FF
0402 2C BIT EOC2 IN DER ANGESPROCHENEN ZELLE STEHT HEX 00
0405 00 BRK
  
```

<P>=00 STATUS DEFINIERT SETZEN

Wenn wir diese Befehlsfolge ausführen, finden wir hinterher mit dem R-Command zur Registeranzeige '32' im Status und 'FF' im Akku. Das Ergebnis zeigt: Der mit FF vorgeladene Akku wurde durch BIT nicht zerstört. Status 32 heißt: Ergebnis Zero (Equal=00). Dieses AND-mäßige Ergebnis entspricht den Erwartungen:

AKKU	1 1 1 1 1 1 1 1
AND-OPERAND	0 0 0 0 0 0 0 0
LOG. ERGEBNIS	0 0 0 0 0 0 0 0

Wir wiederholen den Versuch mit einer Zelle, in der 'FF' steht (AIM!).

```
0402 2C BIT EOC2 ZELLE ENTHÄLT FF
```

Wir finden hinterher FO im Status und FF im Akku. FO bedeutet: Negativ- und Overflow-Flag gesetzt, Ergebnis des AND ungleich Zero ($\neq 0$).

AKKU	1 1 1 1 1 1 1 1
AND-OPERAND	1 1 1 1 1 1 1 1
LOG. ERGEBNIS	1 1 1 1 1 1 1 1

Soweit haben sich keine Besonderheiten ergeben. Wir ändern das Programm nun wie folgt ab:

```

LDA #10
BIT EOD8 IN DIESER SPEICHERZELLE STEHT HEX 80
BRK
  
```

Nach der Ausführung finden wir im Status B2 und die unzerstörte 10 im Akku. Der Status sagt: Negativ-Flag gesetzt und AND-Ergebnis = 0.

AKKU	0 0 0 1 0 0 0 0
AND-OPERAND	1 0 0 0 0 0 0 0
LOG. ERGEBNIS	0 0 0 0 0 0 0 0
GESETZTES FLAG	N

Der BIT-Befehl setzt also in Abhängigkeit vom Operator - gleichgültig welches Bitmuster im Akku besteht - das N-Flag, wenn die vorderste Bitstelle des Operanden eine '1' enthält. Es handelt sich also um eine zusätzliche Leistung oder Prüfung mit dem BIT-Befehl.

Wir führen diese Befehlsfolge nun mit einer Zelle aus, die hex 42 als BIT-Operanden enthält, nämlich E027. Hinterher finden wir 72 im Status und wieder die unzerstörte 10 im Akku. 72 heißt: Overflow-Flag gesetzt und AND-Ergebnis ist Zero.

AKKU	0 0 0 1 0 0 0 0
AND-OPERAND	0 1 1 0 0 0 0 0
LOG. ERGEBNIS	0 0 0 0 0 0 0 0
GESETZTES FLAG	V

Der BIT-Befehl setzt also auch das V-Flag, wenn im zweitvordersten Bit des Operanden eine 1 angetroffen wird, bzw. löscht es, wenn dort 0 ist. Also: Mit dem BIT-Befehl kann man unabhängig vom Akku-Inhalt und ohne diesen zu zerstören, Bitstrukturen im Operanden auf schnellem Wege abfragen, insbesondere für die beiden vordersten Bitpositionen. Abfrage und Verzweigung haben daher oft folgenden Ablauf:

BIT OPERAND
 BMI ODER BPL IN ABHÄNGIGKEIT VOM VORDERSTEN BIT
 BVS ODER BVC IN ABHÄNGIGKEIT VOM ZWEITVORDERSTEN BIT

...
 Die ö.a. Befehlsfolge führen wir schließlich noch mit dem Operanden 57 aus (hex), der in E046 steht. Wir finden danach 70 im Status (V-Flag gesetzt, Not Zero) und wieder 10 im Akku:

AKKU	0 0 0 1 0 0 0 0
AND-OPERAND	0 1 0 1 0 1 1 1
LOG. ERGEBNIS	0 0 0 1 0 0 0 0

Für BIT sind nur die nicht indizierten Adressierungsarten vorgesehen, nämlich Zeropage und absolut.

Es ist nun sicher müßig - wie in unseren Beispielen - nur im Monitorprogramm enthaltene Bytes zu prüfen. Man wendet den BIT-Befehl vor allem zur Prüfung von Registern in Interfacebausteinen an. Dort ist zumeist in vorderster Stelle das Interrupt-Flag für den gesamten Chip enthalten. Bei der Durchprüfung verschiedener zum Interrupt zugelassener Chips kann man mit BIT sehr schnell den aktiven ausmachen. Man ist also nicht auf so umständliche Abfragen angewiesen wie LDA Register, AND #80 und BNE ... In gleicher Weise prüft man schnell auf Pins 7 und 6 gelegte Signale an den Ports der Interfaces.

Innerhalb der Software befragt man mit BIT Tabellen und der Steuerung dienende Verarbeitungsergebnisse hinsichtlich der Bistruktur. Sollen mehr als die beiden vordersten Bits überprüft werden, so gibt es verschiedene Möglichkeiten, z.B.:

65xx MICRO MAG

```

LDA #20
BIT OPERAND
BMI ...      FÜR BIT 7=1
BVS ...      FÜR BIT 6=1
BNE ...      FÜR BIT 5=1
LEAST  LSR .A  RECHTSVERSCHIEBUNG
        BIT OPERAND
        BNE ...      FÜR BIT 4=1 USW.

```

Ab Programmstelle LEAST könnte man auch programmieren:

```

LEAST  LDA OPERAND
        LSR .A      RECHTSVERSCHIEBUNG IM AKKU, BIT 0 INS CARRY
        BCS ...     FÜR BIT 0=1
        ...         USW. MIT RECHTSVERSCHIEBUNGEN

```

Die vorstehende Sequenz kann man natürlich auch in eine Programmschleife bringen. - Die Abfrage der hinteren Bits wird also etwas langwierig.

Für den BIT-Befehl gibt es leider keine Direktoperanden, er wirkt immer Akku gegen Memory. Gleichwohl gibt es einen Ersatz, mit dem man den Status beliebiger Bitpositionen (auch gemischt) überprüfen kann: Man nimmt einen geeigneten Operanden aus dem Festwertspeicher, so wie wir es für den AIM in obenstehenden Beispielen getan haben, und hält den in den Akku geladenen oder dort berechneten Prüfling gegen diesen. Wo finden wir nun geeigneten Operanden? Der AIM hat seinen Monitor selbst analysiert - siehe Tabelle auf der Folgeseite.

(WIRD FORTGESETZT) R.L.

FORTSETZUNG VON SEITE 40:

ORA \$A411		0D11A4
STA \$A411		8D11A4
STY \$A415		8C15A4
BNE IN02		D0D7
IN03 STY \$EA	==0DA3 IN03	84EA
CPY #0		C000
BNE IN05		D015
LDA \$A419		AD19A4
BNE IN05		D010
JSR \$EA24		2024EA
JSR \$FF03		2003FF
IN03A JSR \$F727	==0DB4 IN03A	2027F7
JSR UPNO		20D80C
JSR \$F6D8		20D8F6
BRK		00
IN05 JSR \$F93F	==0DBE IN05	203FF9
JMP \$EA24		4C24EA
.END		
ERRORS= 0000		

R.L.

IN EIGENER SACHE: Drei Papierpreiserhöhungen allein im Jahre 1979, die nachhaltige Steigerung der Postgebühren am Jahreswechsel 1978/79 und die für alle übrigen Leistungen ebenfalls leider festzustellenden Verteuerungen machen nach eineinhalb Jahren auch für diese Zeitschrift eine Preisanpassung zum 1. Dezember 1979 notwendig, um die Leistungskraft zu erhalten. Der neue Abonnementspreis von DM 45,- für 6 Hefte wird sich dabei für die meisten Leser erst bei der Abonnementserneuerung im Sommer 1980 bemerkbar machen. Der Herausgeber bittet um Verständnis.

65_{xx} MICRO MAG

TABELLE: DIREKTOPERANDEN FÜR DEN BIT-BEFEHL - ADRESSEN AUS DEM AIM-MONITOR

MSD	LSD															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	E0CD	E101	E076	E0B0	E079	E245	E15C	E0BA	E0AB	E127	E1AD	E527	E0E4	E203	E0C8	E223
1	E08A	E19A	E225	E103	E096	E0FB	E33C	E134	E2CE	E448	E48B	E205	E23D	E242	E31B	E213
2	E008	E07C	E084	E087	E093	E08B	E08F	E1E9	E6C8	E159	E009	E721	E11D	E368	E09C	E0A2
3	E152	E1DC	E1DD	E1DE	E1DF	E1E0	E1E1	E262	E16F	E1E5	EA55	E05E	EF18	E064	E18F	E1D5
4	E29F	E011	E027	E048	E044	E01F	E000	E1C8	E04C	E02A	E58D	E063	E039	E003	E022	E002
5	E00E	E84C	E001	E00F	E005	E02E	E0DE	E046	E014	E017	E1D9	E1E2	ECE2	E1E3	E1E4	E8D7
6	E119	E180	F313	F2F9	F3C6	E208	E31E	ED90	E07E	EA52	EC2A	E0AD	E075	E0BD	E316	E883
7	E122	E460	E142	E624	FOCE	EE41	EF68	EC3B	E0C0	F060	E188	E148	E13C	E1B3	E1B9	E082
8	E0D8	-	E1AA	E8C9	E3D8	E265	E268	E476	E0A5	-	E146	E51B	E086	E07B	E083	EC47
9	E0AF	E83A	-	E2EA	E219	E3EF	E18B	FFA3	E2CD	E09B	E0C3	-	EF15	E0CC	E211	F498
A	E023	E38C	E0C1	E448	E077	F54A	E3DE	E453	E0CE	E0FE	E14E	-	E095	E098	E1A5	E22D
B	E21E	E7AD	F525	F576	F60A	EE73	F674	F489	F835	E2DD	E091	-	E186	E004	E2E0	E020
C	E286	E069	-	E368	E06A	E053	E026	E75C	E25C	E2AE	E0CF	E029	E067	E0E0	E047	E056
D	E0E3	E3E6	E054	E01B	E19F	F85E	-	E715	E082	E20D	FB4B	E2A4	E05D	E196	F12B	F6A1
E	E0FA	E0B3	E08E	E308	E1F4	E1B0	E0A6	E0AE	E0F9	E0BB	E140	E258	E5B7	E336	E1F9	E14C
F	E0AA	E25F	E1FB	E370	E151	E0FD	E1E6	E0D1	E0B8	E123	E201	E12D	E991	E307	E13D	E0C2

Stellenanzeige

Hersteller von Spezialelektronik im Großraum München
sucht zum nächstmöglichen Termin:

ENTWICKLUNGS UND FERTIGUNGSINGENIEUR

erfahren in single board Computern, z.Zt. 6502 (A)
und R 6502, Erweiterungsplatten einschl. Entwicklung
und Bau von Peripherie.

Voraussetzung: Hochschulstudium oder gleichwertige
Erfahrung aus der Praxis.

Ferner:

LABOR- UND SERVICETECHNIKER

für Arbeiten mit Mikroprozessoren und dazugehörigen
mechanischen Aufbauten.

Ausführliche schriftliche Bewerbungen an den Heraus-
geber. Sperrvermerke werden berücksichtigt.

AUFTRAG ZU VERGEBEN

AN NACHWEISBAR ERFAHRENE SPEZIALISTEN FÜR DIE

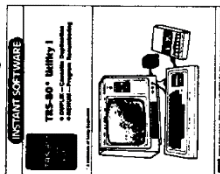
AUFLÖSUNG EINES MIKROCOMPUTERS (CPU 6502) -
1-10 K PROM, 1-16 K RAM,
STROMVERSORGUNG, DISPLAY, TASTATURANSCHLUSS,
DRUCKERANSCHLUSS UND MEHRERE SCHNITTSTELLEN
DER AUFTRAG UMFASST GETESTETE LEITERPLATTEN.
INDUSTRIEMÄSSIGE AUSFÜHRUNG ERFORDERLICH.

Angebote mit Preis- und Terminvorstellungen treuhän-
derisch an den Herausgeber.

INSTANT SOFTWARE

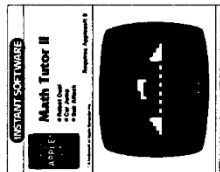
Belvoirborough New Hampshire 03458

Programme auf C-Cassette für Ihren TRS-80, CBM (Pet) und APPLE II. 80 Programme sind bereits lieferbar. Händleranfragen willkommen.



TTS-80 UTILITY I Ever wonder how some programmers give their programs that professional look? Instant Software has the answer with the TTS-80 Utility. Package included are:

- **RENUM** - Now you can easily renumber any Level II program to make room for modification, or to clean up the listing.
- **DUPLX** - This program will let you duplicate any BASIC, assembler, or machine-language program, verify the data, merge two or more programs into one large block, and even copy Level I programs into a Level II machine. For TTS-80 Level II (EK, Order No. 00911)



MATH TUTOR I Parents, teachers, students, now you can turn your Apple computer into a math tutor! Your children or students can learn math concepts and lessons with these programs. Their math skills will improve and they will love it!

- **Hopping -** Perfect your skills with decimal numbers while you try to cheat the hognomen.
- **Spellfinder -** Cast spells against a competing magician as you practice working with fractions.
- **Whole Space -** While you exercise your whole space, you exercise your whole self! While using whole numbers, you ship spacecrafts the enemy planet and destroys alien spacecrafts. All programs have varying levels of difficulty. All you need is AppleLink II with your Apple II 24K, Order No. 00732A.

24,90 DM incl. 13% MwSt. Bitte fordern Sie
INSTANT-SOFTWARE-Liste an oder lesen Sie
regelmäßig Klobaud MICROCOMPUTING.

INSTANT SOFTWARE

J E T Z T Programme in DEUTSCH

ISI 005P Biorythmus und Gewichts-
Kontrolle

ISI 006P Hypotheken Berechnung
mit Vorräszahlung und
FINANZIERUNGSBERECHNUNG

ISI 014P Casino I(Blackjack/Roulette)

ISI 022P DAME und BACCARA

ISI O29P Spiele für zwei Spieler:
Tangle und Supertrap

ISI 038P Qubic und Go-Muko

Ausserdem 30 Programme in Englisch,
Preis DM 24,90 incl. 13% Mwst.
INFO anfordern oder MICROCOMPUTING
das aktuelle amerikanische Magazin
lesen.

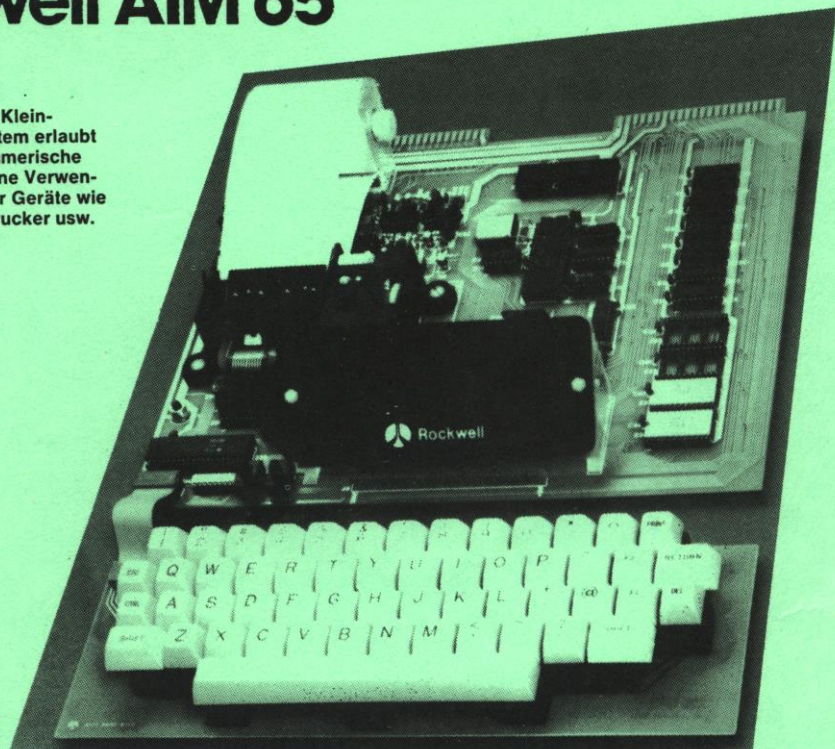
Telefon 07544-3575

MSB Micro-Shop-Bodensee
M & R Nedela
Postfach 1122
Marktstraße 3
7778 Markdorf

For learning, designing, work or just plain fun...
The Head-Start in Computers

Rockwell AIM 65

Das vollständige Klein-Entwicklungssystem erlaubt erstmals alphanumerische Ein-/Ausgabe ohne Verwendung zusätzlicher Geräte wie TTY, CRT, TV, Drucker usw.



Rockwell

- Tastatur im «Terminal style»
- 20 charakter alphanumerische Anzeige
- 20 charakter alphanumerischer Thermodrucker
- R 6502 CPU
- bis 4K RAM «on board», extern erweiterbar
- Interfaces für zwei Kassettengeräte und TTY
- Assembler und Basic in ROM
- Lieferung

PREISE:

AIM 65 mit 1K Byte RAM	875,--DM
AIM 65 mit 4K Byte RAM	1050,--DM
Resident Assembler ROM 4K	230,--DM
BASIC Interpreter ROM 8K	280,--DM
On Board RAM Erweiter. 3K	175,--DM

Netzteil 5V/3A, 24V/1,5A 315,--DM

Alle Preise zuzüglich MWST und Versandspesen.

D-5120 HERZOGENRATH
ASTERNSTRASSE 2
TEL.: 02406-62394

GWK

FÜR TECHNISCHE

GESELLSCHAFT
ELEKTRONIK mbH.

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 1979 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: Gerhard M. Meier, Hamburg 70.

BEZUGSBEDINGUNGEN: Abonnement ab laufender Nr. für 6 Hefte DM 40,- (Inlandsendpreis). Ausland/Foreign DM 46,- (surface mail). USA: \$ 28.00 air. Alle früher erschienenen Hefte dieser Zeitschrift können zum Einzelpreis von DM 7,- nachbezogen werden. Ab 1. Dezember 1979 gelten neue Abonnementspreise: DM 45,- im Inland, DM 50,- Ausland, \$ 30 US air. Richten Sie Ihre Überweisungen/Schecks an Roland Löhr, Konto 21/01122 bei der Vereins- und Westbank Ahrensburg, BLZ 200 300 00. Bezieher im Ausland überweisen am preiswertesten auf dem Postwege.

AIM 65-SERVICE

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/PC100 in kontrastreicher Spitzenqualität, Qualitätsfreigabe durch Rockwell, Californien. Packung mit 8 Großrollen, zus. 520 m, 57 mm breit, preiswert. Packung

DM 50,85

Ersatzplatte für Thermodrucker mit Montageanleitung (Wechsel in 10 Minuten) frischt den Druck wieder auf

St.DM 21,--

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

ASSEMBLER-WORKSHOP AIM 65 - PC100

Wochenendworkshop von Freitag, den 23. Nov. bis Sonntag, den 25. Nov. 1979 in Augsburg.

Intensives Programmierseminar am Gerät, einschl. Interfaceprogrammierung. Unter Leitung des Herausgebers von erfahrenen Dozenten didaktisch aufgebaut und mit Ausgabe der Programmierung auf Bildschirme für die Teilnehmer (mit Vorkenntnissen), einschl. Lehrgangsbuch, ca. 160 Seiten. Sonderprospekt/Anmeldung beim Herausgeber.