

# 65<sub>xx</sub> MICRO MAG

## COMPUTING · SOFTWARE · HOBBY

DM 7,-

Nr. 8

AUGUST 1979

### SCHWERPUNKT VIDEO

*Für die Kommunikation mit dem Bediener ist der Bildschirm wohl das wichtigste Interface eines Computers. Es arbeitet lautlos, gibt großformatige Übersichten und bedruckt für vorübergehende Information nicht unnötig Papier.*

*Wichtige Beiträge dieses Heftes sind auf das aktuelle Thema Video abgestellt. Für den PET ist ein sehr nützlicher Video-Editor enthalten. Speziell für den AIM 65 legt ein großer Grundsatzartikel zwei komplette Schaltungen für ein Video-Interface sowie die zum Betrieb benötigte Software dar. Ein weiterer Artikel eröffnet mit einem einfachen Interface und dem Betriebsprogramm die Textausgabe auf den Bildschirm eines Oszilloskopes. - Das Thema Video wird durch zahlreiche aktuelle Hinweise und auch durch die für den AIM nun mögliche Kleinschreibung abgerundet.*

*Diese Zeitschrift wird die maschinennahe Programmierung weiter pflegen, speziell auch für den PET, für den in diesem Sinne ein Primzahlenprogramm aus einer ganzen Reihe bereits angemeldeter Programme vorgelegt wird. Die Reihe der so erfolgreichen Artikel wie über das VIA 6522 und jetzt über die CRT-Controller wird für weitere Interfacebausteine fortgesetzt werden, so wie auch die bereits begonnenen Serien im Herbst ihren Schluß finden.*

*Der Leser sei besonders auf den ersten AIM 65-PC100-Workshop und auf das offene 65xx-Computermeeting im Raume Frankfurt hingewiesen.*

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

Primfaktoren-Zerlegung (PET)	3
Video-Edit (PET)	11
PET-Petits	14
Gedanken zum Video-AIM	16
Video-News	32
Oszillograph als Bildschirm für den AIM 65	33
Zufallszahlengenerator	35
Auskunftssystem mit dem AIM 65	36
AIM-Tastatur mit Kleinschreibung	39
AIM-Spezial (3)	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 verschwendet.

Der GWK SYSTEM BUS und die Steckverbinder der Einzelkomponenten sind pinkompatibel 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 PROGRAMMER BOARD**

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

ELEKTRO-HOBBY 79 vom 4.-7. Okt., Killesberg, Stuttgart: Das GWK-System ist auf dem Stand des Franzis-Verlages ausgestellt.

# 65xx MICRO MAG

## PRIMFAKTOREN-ZERLEGUNG

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

E A PET BASIC program is combined with a machine language program in order to divide any integer number into its prime numbers. The integer to be divided may have up to 39 digits. This program executes very fast, nevertheless much time may be consumed to work on numbers in this magnitude. - The count of keystrokes may not be altered when typing in since the address space of the machine language program is fixed as listed below.

### Vorbemerkungen

Das Commodore-BASIC hat zwei Arten der internen Zahlendarstellung:

- a) Festkommazahlen (Integerzahlen) von -32768 bis 32767, dargestellt in zwei Bytes, und
- b) Fließkommazahlen von rd.  $10^{-38}$  bis  $10^{38}$ , dargestellt in fünf Bytes (1 Byte Exponent und 4 Bytes Mantisse).

Für ein Programm, das ganze positive Zahlen größer 32767 in Primfaktoren zerlegen soll, kommt daher nur die Fließkommadarstellung in Frage. Aber auch hier ist die Freude nicht ungetrübt. Da die Mantisse vier Bytes groß ist, können nur Zahlen mit maximal neun signifikanten Stellen zerlegt werden. Für Zahlen mit mehr als neun signifikanten Stellen sind folglich eigene Routinen zu schreiben.

Bei dem folgenden Programm erfolgt ein Teil der Ein- und Ausgabe sowie die gesamte Zerlegung in Maschinensprache-Unterroutinen. Die zu zerlegende Zahl kann bis zu 39 Stellen groß sein. Eine Erhöhung der Stellenzahl ist nicht sinnvoll, da bereits die Zerlegung einer 39-stelligen Zahl unter Umständen Tage dauern kann.

### Arbeitsweise des Programms

Die Eingabe erfolgt in einer in Maschinensprache geschriebenen Unterroutine. Dort findet auch die Umwandlung von der ASCII-Darstellung in eine Binärzahl von 16 Byte statt. Bei der Eingabe von mehr als 39 Stellen oder eines Buchstabens oder Sonderzeichens wird ein Returncode (=RC) gesetzt, der im BASIC-Programm ausgewertet wird.

Bei der Zerlegung wird die eingegebene Zahl (=ZAHL) auf ihre Teilbarkeit durch Primzahlen geprüft. Ist die Zahl teilbar, so wird der Divisor im Feld PRFAKT zwischengespeichert und die Zahl auf erneute Teilbarkeit geprüft. Ist die Zahl nicht ohne Rest durch den Divisor teilbar, so wird der Divisor um einen Wert aus einer Tabelle (=INCR) erhöht. Die Zerlegung ist beendet, wenn ZAHL gleich 1 ist (Returncode =1) oder wenn ZAHL kleiner als das Quadrat des Divisors ist (Returncode =2). In diesem Fall ist der Inhalt von Zahl auch eine Primzahl.

### Programmiertechnik

Die Unterroutinen in Maschinensprache sind Bestandteil des Programms; d.h. die Befehle werden nicht in DATA-Statements definiert und nicht mit einer Initialisierungsroutine über POKE in den Speicher gebracht. Daher dürfen im BASIC-Teil auch nur Veränderungen vorgenommen werden, die die Gesamtlänge des BASIC-Programmes nicht verändern, da sich sonst die Adressen des Maschinensprache-Teiles verschieben. Bei einer Änderung ist eine Längenanpassung in den Zeilen 340 und 350 vorzunehmen.

Das BASIC-Programm ist normal einzugeben. Danach ist die Länge des Programms mit POKE 124,254 und POKE 125,9 zu modifizieren. Jetzt kann der Teil in Maschinensprache mit einem Monitor-Programm oder durch POKE an das BASIC-Programm angehängt werden.

```

10 REM AUTOR PETER W. ARPS
20 REM
30 PRINT CHR$(147);CHR$(18);TAB(8);"PRIMFAKTOREN-ZERLEGUNG"
40 PRINT
50 PRINT"ZU ZERLEGENDE GANZE POSITIVE"
60 PRINT"ZAHL EINGEBEN!"
70 PRINT:PRINT
80 PRINT"WIE LAUTET DIE ZAHL?"
90 SYS(1872):REM EINGAB + PRUEFUNG
100 ON PEEK(714)+1GOTO130,120,110
110 PRINT:PRINT"EINGABE ZU LANG!":GOTO70
120 PRINT:PRINT"UNGUELTIGES ZEICHEN!":GOTO70
130 PRINTCHR$(147);"DIE ZAHL":SYS(1976)
140 PRINT"LAESST SICH IN FOLGENDE PRIMFAKTOREN"
150 PRINT"ZERLEGEN:"
160 SYS (2007):REM ZERLEGUNG
170 RC=PEEK(714):IFRC=3THENSTOP
180 IFRC=2THENIFPEEK(712)=0THEN205
190 SY=2166:GOSUB300;IFRC=0THEN160
200 IFRC=1THEN210
205 SY=2198:POKE 712,1:GOSUB300
210 PRINT:PRINT:PRINT"NEUER DURCHGANG?"
220 PRINT""J" ODER 'N' DRUECKEN!"
230 GETA$:IFAS=""THEN230
240 IFAS=""J"THEN70
250 IFAS><"N"THEN230
260 PRINTCHR$(18);" T S C H U E S S "
270 T=TI
280 IF (TI-T)<300THEN280
290 PRINTCHR$(147):END
300 P=PEEK(712)
310 PRINTSPC(5-LEN(STR$(P)));P;"MAL DIE";
320 SYS(SY):REM AUSGABE DES PRIMFAKTORS
330 RETURN
340 REM FILLER FILLER FILLER
350 REM FILLER FILLER

```

REDAKTIONS- UND  
 ANZEIGENSCHLUSS  
 FÜR HEFT 8  
 5. OKTOBER 1979

SYMBOLE UND IHRE WERTE:

DVDENT=\$24	CHAR	=\$280
REST =\$34	ZAHL	=\$2A8
LOW =\$44	TEILER	=\$2B8
HIGH =\$45	PRFAKT	=\$2C0
LNG =\$46	MAL	=\$2C8
VON =\$47	LAENGE	=\$2C9
NACH =\$49	RC	=\$2CA
ZF1 =\$4B	SAVE	=\$2CB
ZF2 =\$4C	STK	=\$103
	READ	=\$FFCF
	WRITE	=\$FFD2
	STOP	=\$F32A

65<sub>xx</sub> MICRO MAG

0750	D8	START	CLD	MODUS BINÄR
51	20 34 09		JSR CLEAR	ZAHL BIS SAVE LÖSCHEN
54	23		.BYTE 35	
55	A8 02		.WORD ZAHL	
57	85 4B		STA ZF1	ZF1 LÖSCHEN
59	20 E3 08		JSR CR	CR SCHREIBEN
5C	A0 00		LDY #\$00	
5E	20 CF FF	INPUT	JSR READ	ZAHL LESEN UND SPEICHERN
61	99 80 02		STA CHAR,X	
64	C9 0D		CMP #13	
66	F0 08		BEQ EODAT	ENDE DER EINGABE MIT CR
68	C8		INY	HOLE WEITERE ZEICHEN
69	C0 28		CPY #40	MAXIMAL ABER 39
6B	D0 F1		BNE INPUT	
6D	4C D0 07		JMP RC2	FEHLERCODE SETZEN
70	8C C9 02	EODAT	STY LAENGE	LÄNGE DER EINGABE SICHERN
73	84 4C		STY ZF2	
75	4C 94 07		JMP FIRST	UMWANDLUNG ASCII NACH HEX
78	A2 0F	LOOP1	LDX #15	
7A	18		CLC	
7B	3E A8 02	ROLZ	ROL ZAHL,X	
7E	CA		DEX	
7F	10 FA		BPL ROLZ	
81	20 41 09		JSR MOVE	
84	0F		.BYTE 15	LÄNGE -1
85	A8 02		.WORD ZAHL	
87	24 00		.WORD DVDENT	
89	20 F7 08		JSR RLDVD2	ZAHL x2x4 ENTSPRICHT x8
8C	20 6F 09		JSR PLUS	ZAHL x2 PLUS ZAHL x8
8F	0F		.BYTE 15	= ZAHL x10
90	A8 02		.WORD ZAHL	
92	24 00		.WORD DVDENT	
94	A6 4B	FIRST	LDX ZF1	
96	BD 80 02		LDA CHAR,X	
99	38		SEC	
9A	E9 30		SBC #48	
9C	30 35		BMI RC1	ZEICHEN IST KEINE ZAHL
9E	C9 0A		CMP #10	
A0	B0 31		BCS RC1	ZEICHEN IST KEINE ZAHL
A2	18		CLC	
A3	6D B7 02		ADC ZAHL+15	
A6	8D B7 02		STA ZAHL+15	
A9	90 06		BCC ++6	
AB	20 7F 09		JSR UEBTR	ZAHL UM 1 ERHÖHEN
AE	0E		.BYTE 14	
AF	A8 02		.WORD ZAHL	
B1	E6 4B		INC ZF1	
B3	C6 4C		DEC ZF2	
B5	D0 C1		BNE LOOP1	EINGABE NOCH NICHT ABGEARBEITET
B7	60		RTS	
B8	A9 12	AUSGABE	LDA #18	DIE EINGABE REVERSE DARSTELLEN
BA	20 D2 FF		JSR WRITE	
BD	AC C9 02		LDY LAENGE	
C0	A2 00		LDX #0	
C2	BD 80 02	OUTL	LDA CHAR,X	
C5	20 D2 FF		JSR WRITE	ZEICHEN SCHREIBEN

07C8	E8		INX	
C9	88		DEY	
CA	10 F6		BPL OUTL	
CC	60		RTS	
CD	EE CA 02	RC3	INC RC	RETURNCODES SETZEN
D0	EE CA 02	RC2	INC RC	
D3	EE CA 02	RC1	INC RC	
D6	60		RTS	
D7	D8	ENTRY1	CLD	MODUS BINAR
D8	AD CB 02		LDA SAVE	TEST AUF INDEX =0.
DB	F0 06		BEQ ERHOEH	ERSTMALIGER PROGRAMMAUFRUF
DD	20 9B 09		JSR ENDR2	TEST AUF ENDBEDINGUNG 2
E0	4C 11 08		JMP DIVISN	NICHT ENDE
E3	20 91 09	ERHOEH	JSR ENDR1	TEST AUF ZAHL=1, WENN JA, ENDE
E6	AC CB 02		LDY SAVE	INDEX FÜR INCREMENT LADEN
E9	B9 E8 08		LDA INCR,Y	INCREMENT LADEN
EC	6D BF 02		ADC TEILER+7	TEILER UM INCREMENT ERHÖHEN
EF	8D BF 02		STA TEILER+7	
F2	90 06		BCC *+6	KEIN ÜBERTRAG
F4	20 7F 09		JSR UEBTR	
F7	06		.BYTE 6	
F8	B8 02		.WORD TEILER	
FA	EE CB 02		INC SAVE	INCREMENT-INDEX ERHÖHEN
FD	AC CB 02		LDY SAVE	
0800	C0 0C		CPY #12	
02	90 0D		BNE DIVISN	NOCH NICHT 12
04	20 9B 09		JSR ENDR2	TEST AUF ENDEBEDINGUNG 2
07	A9 04		LDA #4	
09	8D CB 02		STA SAVE	INCREMENT-INDEX AUF 4
0C	20 2A F3		JSR STOP	
0F	F0 BC		BEQ RC3	
11	20 41 09	DIVISN	JSR MOVE	DIVISIONS-ROUTINE
14	0F		.BYTE 15	ZAHL NACH DV DENT
15	A8 02		.WORD ZAHL	
17	25 00		.WORD DV DENT	
19	20 34 09	WEITER	JSR CLEAR	REST LÖSCHEN
1C	07		BYTE 7	
1D	34 00		.WORD REST	
1F	A0 80		LDY 128	ANZAHL BITS DES DIVISORS LADEN
21	84 4B		STY ZF1	... UND SICHERN
23	18		CLC	
24	4C 3E 08		JMP DIV 2	
27	20 5D 09	DLOOP	JSR CMPS	REST KLEINER TEILER?
2A	07		.BYTE 7	
2B	34 00		.WORD REST	
2D	B8 02		.WORD TEILER	
2F	90 0D		BCC DIV2	JA
31	A2 07		LDX #7	BEI NEIN
33	38		SEC	
34	B5 34	MINUS	LDA REST,X	REST - TEILER = REST
36	FD B8 02		SBC TEILER,X	
39	95 34		STA REST,X	
3B	CA		DEX	
3C	10 F6		BPL MINUS	
3E	20 FB 08	DIV2	JSR ROLDVD+1	DIVIDEND UM 1 BIT N. LINKS SHIFTEN
41	C6 4B		DEC ZF1	

65<sub>xx</sub> MICRO MAG

0843	30 06		BMI DIVEND	DIVISION BEEENDET
45	20 04 09		JSR RLREST+1	REST UM 1 BIT NACH LINKS SHIFTEN
48	4C 27 08		JMP DLOOP	
48	20 4E 09	DIVEND	JSR CMPN	REST =0?
4E	07		.BYTE 7	
4F	34 00		.WORD REST	
51	90 90		BCC ERHOEH	NEIN, TEILER IST NICHT PRIMFAKTOR
53	20 41 09		JSR MOVE	
56	0F		.BYTE 15	
57	24 00		.WORD DVDENT	
59	A8 02		.WORD ZAHL	
5B	20 5D 09		.JSR CMPS	WURDE DER GLEICHE PRIMFAKTOR
5E	07		.BYTE 7	BEREITS GEFUNDEN ?
5F	C0 02		.WORD PRFAKT	
C1	B8 02		.WORD TEILER	
63	90 06		BCC **6	NEIN
65	EE C8 02		INC MAL	JA, NUR ZAHLFELD ERHÖHEN
68	4C 19 08		JMP WEITER	
6B	AD C8 02		LDA MAL	1. GEFUNDENER PRIMFAKTOR?
6E	D0 25		BNE RET	NEIN
70	20 88 08		JSR ENTR2A	JA, NUR FELDER ÜBERTRAGEN
73	4C D7 07		JMP ENTRY1	UND NÄCHSTE PRÜFUNG
76	D8	ENTRY2	CLD	AUSGABE DES PRIMFAKTORS
77	20 34 09		JSR CLEAR	
7A	07		.BYTE 7	
7B	24 00		.WORD DVDENT	
7D	20 41 09		JSR MOVE	
80	07		.BYTE 7	
81	C0 02		.WORD PRFAKT	
83	2C 00		.WORD DVDENT+8	
85	20 9F 08		JSR ASCII	UMWANDLUNG IN ASCII-CODE UND
88	A9 01	ENTR2A	LDA #1	AUSGABE
8A	8D C8 02		STA MAL	
8D	20 41 09		JSR MOVE	
90	07		.BYTE 7	
91	B8 02		.WORD TEILER	
93	C0 02		.WORD PRFAKT	
95	60	RET	RTS	
96	D8	ENTRY3	CLD	
97	20 41 09		JSR MOVE	ZAHL NACH DVDENT
9A	0F		.BYTE 15	
9B	A8 02		.WORD ZAHL	
9D	24 00		.WORD DVDENT	
9F	A9 27	ASCII	LDA #39	INDEX FÜR CHAR LADEN
A1	85 4B		STA ZF1	UND SICHERN
A3	A9 00	ALOOP1	LDA #0	
A5	85 34		STA REST	REST LÖSCHEN
A7	A0 80		LDY #128	ANZAHL BITS VON DVDENT LADEN
A9	38	ALOOP2	SEC	
AA	A5 34		LDA REST	IST REST => 10 ?
AC	E9 0A		SBC #10	
AE	10 04		BPL **4	JA
B0	18		CLC	
B1	4C B7 08		JMP ARL	
B4	85 34		STA REST	
B6	38		SEC	

65<sub>xx</sub> MICRO MAG

08B7	20 FB 08	ARL	JSR RLDVD*1	DVDENT UM 1 BIT NACH LINKS SHIFTEN
BA	88		DEY	FERTIG ?
BB	30 05		BMI ARAUS	JA
BD	26 34		ROL REST	REST UM 1 BIT NACH LINKS SHIFTEN
BF	4C A9 08		JMP ALOOP2	WEITER
C2	A5 34	ARAUS	LDA REST	
C4	09 30		ORA #48	
C6	A6 4B		LDX ZF1	INDEX LADEN
C8	9D 80 02		STA CHAR,X	ZEICHEN SPEICHERN
CB	C6 4B		DEC ZF1	
CD	20 4E 09		JSR CMPN	DVDENT =0 ?
D0	0F		.BYTE 15	
D1	24 00		.WORD DVDENT	
D3	90 CE		BCC ALOOP1	NEIN, NÄCHSTES ZEICHEN HOLEN
D5	A6 4B		LDX ZF1	
D7	E8		INX	
D8	BD 80 02	ALOOP3	LDA CHAR,X	
DB	20 D2 FF		JSR WRITE	ZEICHEN AUSGEBEN
DE	E8		INX	
DF	E0 28		CPX #40	
E1	D0 F5		BNE ALOOP3	
E3	A9 0D	CR	LDA #13	
E5	4C D2 FF		JMP WRITE	CR SCHREIBEN
E8	02 01 02	INCR	.BYTE	2,1,2,2,4,2,4,2,4,6,2,6,
EB	02 04 02			
EE	04 02 04			
F1	06 02 06			
F7	20 FA 08	RLDVD2	JSR ROLDVD	DVDENT UM 2 BIT NACH LINKS SHIFTEN
FA	18	RLDVD	CLC	DITO, UM 1 BIT NACH LINKS SHIFTEN
FB	A2 0F		LDX #15	
FD	36 24	RLDVDL	ROL DVDENT,X	
FF	CA		DEX	
0900	10 FB		BPL RLDVDL	
02	60		RTS	
03	18	RLREST	CLC	REST UM 1 BIT NACH LINKS SHIFTEN
04	A2 07		LDX #7	
06	36 24	RLRSTL	ROL REST,X	
08	CA		DEX	
09	10 FB		BPL RLRSTL	
0B	60		RTS	
0C	8E 1F 09	HOLPRM	STX OPMOD1+1	PARAMETERLÄNGE IN DEN BEFEHL
0F	8E 29 09		STX OPMOD2+1	
12	BA		TSX	
13	BD 04 01		LDA STK+1,X	AUFRUFADRESSE SICHERN
16	85 45		STA HIGH	
18	BD 03 01		LDA STK,X	
1B	85 44		STA LOW	
1D	18		CLC	
1E	69 FF	OPMOD1	ADC #255	... UND UM PARAMETERLÄNGE ERHÖHEN
20	9D 03 01		STA STK,X	(STACKMANIPULATION)
23	90 03		BCC **3	
25	FE 04 01		INC STK+1,X	
28	A0 FF	OPMOD2	LDY #255	
2A	B1 44	HOLADR	LDA (LOW),Y	PARAMETER NACH LNG, VON UND
2C	99 45 00		STA LNG-1,Y	GGFS. NACH ÜBERTRAGEN

65<sub>xx</sub> MICRO MAG

092F	88		DEY	
30	D0 F8		BNE HOLADR	
32	A8		TAY	
33	60		RTS	
34	A2 03	CLEAR	LDX #3	LÖSCHROUTINE
36	20 0C 09		JSR HOLPRM	
39	A9 00		LDA #0	
3B	91 47	CLRL	STA (VON),Y	
3D	88		DEY	
3E	10 FB		BPL CLRL	
40	60		RTS	
41	A2 05	MOVE	LDX #5	ÜBERTRAGUNGSRoutine
43	20 0C 09		JSR HOLPRM	
46	B1 47	MVL	LDA (VON),Y	
48	91 49		STA (NACH),Y	
4A	88		DEY	
4B	10 F9		BPL MVL	
4D	60		RTS	
4E	A2 03	CMPN	LDX #3	VERGLEICH AUF 0
50	20 0C 09		JSR HOLPRM	
53	18		CLC	
54	B1 47	CMPNL	LDA (VON),Y	
56	D0 04		BNE ++4	
58	88		DEY	
59	10 F9		BPL CMPNL	
5B	38		SEC	CY=0, FELD NICHT 0
5C	60		RTS	CY=1, FELD OST 0
5D	A2 05	CMPS	LDX #5	VERGLEICH ZWEIER FELDER
5F	20 0C 09		JSR HOLPRM	
62	38		SEC	
63	B1 47	CMPSL	LDA (VON),Y	
65	F1 49		SBC (NACH),Y	
67	88		DEY	
68	D0 F9		BNE CMPSL	
6A	B1 47		LDA (VON),Y	
6C	F1 49		SBC (NACH),Y	CY=1, VON≥NACH
6E	60		RTS	CY=0, VON<NACH
6F	A2 05	PLUS	LDX #5	ADDITIONSRoutine
71	20 0C 09		JSR HOLPRM	VON = VON+NACH
74	18		CLC	
75	B1 47	PLUSL	LDA (VON),Y	
77	71 49		ADC (NACH),Y	
79	91 47		STA (VON),Y	
7B	88		DEY	
7C	10 F7		BPL PLUSL	
7E	60		RTS	
7F	A2 03	UEBTR	LDX #3	ÜBERTRAG VORNEHMEN
81	20 0C 09		JSR HOLPRM	
84	38		SEC	
85	B1 47	UEBL	LDA (VON),Y	
87	69 00		ADC #0	
89	91 47		STA (VON),Y	
8B	90 03		BCC ++3	
8D	88		DEY	
8E	10 F5		BPL UEBL, RTS	
90	60		RTS	

0991	20 EF 09	ENDPR1	JSR TST1	FELD=1 ?
94	90 57		BCC RTN	NEIN
96	68		PLA	RÜCHSPRUNGADRESSE VERÄNDERN
97	68		PLA	
98	4C D3 07		JMP RC1	RC=1 SETZEN UND ZURÜCK ZU BASIC
9B	20 EF 09	ENDPR2	JSR TST1	FELD=1 ?
9E	B0 4D		BCS RTN	JA, ZURÜCK
A0	A2 07		LDX #7	
A2	BD B0 02	ELOOP2	LDA ZAHL+8,X	ZAHL=TEILER ?
A5	DD B8 02		CMP TEILER,X	
A8	D0 06		BNE **6	
AA	CA		DEX	
AB	10 F5		BPL ELOOP2	
AD	4C ED 09		JMP RTN	JA, ZURÜCK
B0	20 34 09		JSR CLEAR	
B3	0F		.BYTE 15	TEILER QUADRIEREN
B4	24 00		.WORD DVDENT	
B6	20 41 09		JSR MOVE	
B9	07		.BYTE 7	
BA	B8 02		.WORD TEILER	
BC	34 00		.WORD REST	
BE	A0 40		LDY #64	
C0	84 4B		STY ZF1	
C2	20 F1 08	ELOOP3	JSR ROLDVD	
C5	20 03 09		JSR RLREST	
C8	90 10		BCC ETST	
CA	20 6F 09		JSR PLUS	
CD	07		.BYTE 7	
CE	2C 00		.WORD DVDENT+8	
D0	B8 02		.WORD TEILER	
D2	90 06		BCC ETST	
D4	20 7F 09		JSR UEBTR	
D7	07		.BYTE 7	
D8	24 00		.WORD DVDENT	
DA	C6 4B	ETST	DEC ZF1	
DC	D0 E4		BNE ELOOP3	
DE	20 5D 09		JSR CMPS	ZAHL < ALS TEILER <sup>2</sup> ?
E1	0F		.BYTE 15	
E2	A8 02		.WORD ZAHL	
E4	24 00		.WORD DVDENT	
E6	B0 05		BCS RTN	NEIN, ZURÜCK
E8	68		PLA	JA, ZAHL IST PRIMFAKTOR
E9	68		PLA	RÜCKSPRUNGADRESSE VERÄNDERN
EA	4C D0 07		JMP RC2	RC=2 SETZEN UND ZURÜCK ZU BASIC
ED	18	RTN	CLC	
EE	60		RTS	
EF	AD B7 02	TST1	ZAHL+15	
F2	C9 01		CMP #1	
F4	D0 F7		BNE RTN	
F6	20 4E 09		JSR CMPN	
F9	0E		.BYTE 14	
FA	A8 02		.WORD ZAHL	
FC	60		RTS	

(Anmerkung des Herausgebers: Dieses Programm dürfte sich ohne große Schwierigkeiten auch auf andere Systeme einrichten lassen, wenn man die Besonderheiten für Ein- und Ausgabe sowie für den SYS-Befehl berücksichtigt.)

**65<sub>xx</sub> MICRO MAG**

## VIDEO-EDIT

## GRAPHIK-GENERATOR FÜR PET 2001

Ing. (grad.) Uwe Kornnagel, Lahnstraße 6, D-6096 Raunheim/Main

E: This screen editor allows to deposit any image in a 1 kbyte at the top of RAM, where it can be protected. This image may be written to the screen repeatedly as a background and it may receive an overlay from printing statements of a BASIC program. The latter output may add movement to a constant background and vice versa.

Dieser Video-Editor ist als Maschinenprogramm zusammen mit dem Videodriver des Verfassers (Heft 7, Seite 38) zu betreiben. Das Programm legt in die letzten 1000 Bytes (entsprechend einem Bildschirminhalt) des 8k-PET ein Zeichenmuster nach Wahl des Betreibers ab, das jederzeit wieder auf den Bildschirm zurückgerufen werden kann und das durch seine Lage im Speicher gegen BASIC schützbar ist (memory size).

Unter Zuhilfenahme aller alphanumerischen und graphischen Darstellungsmöglichkeiten des PET nimmt der 'permanente Bildspeicher' beliebige Formulare oder Hintergrundkulissen in Spielen auf. Transportiert man seinen Inhalt mit SYS auf den Bildschirm und füllt hernach vorgesehene Freiräume mit PRINT-Statements eines beliebigen aufrufenden BASIC-Programmes, so generiert man ein bildliches Overlay mit Texten oder auch durch eingeblendete bewegliche Figuren. Die hervorragenden Möglichkeiten für z.B. Datenerfassung (Eintragen in Formulare) oder für Spiele sind damit aufgezeichnet. Verfasser und Herausgeber würden sich über weitere Anregungen und Berichte freuen.

Der Video-Editor entlastet den Betreiber vom mühseligen Aufbau eines Formulares/Hintergrundes mit PRINT-Statements. Mit den üblichen Cursorbewegungen gestaltet und korrigiert er in aller Ruhe bis zum Punkt der Zufriedenheit und verläßt dieses Dienstprogramm erst mit SHIFT/RETURN. Nach dem Eintippen und Erproben eines aufrufenden BASIC-Programmes kann er dieses dann zusammen mit dem 'permanenten Bildschirmspeicher' auf Magnetbandcassette abspeichern.

SYMBOLE:	CLMN	= 226	
	VIDP	= 224	
	FRM	= \$B0	
	TO	= \$B2	
	PRMT	= 1017	
	PRINT	= \$FFD2	; PRINTROUTINE DES PET
	GET	= \$FFE4	; GETROUTINE DES PET
	CLR	= \$E236	; BILDSCHIRMLÖSCHROUTINE DES PET
	NEW	= \$C553	; BASIC - NEWROUTINE DES PET
	WAIT	= 994	; UTILITY DES VIDEO DRIVER

DAS PROGRAMM IST ZUSAMMEN MIT DEM VIDEO DRIVER AUS HEFT 7 ZUM ZWECKE DES BILDAUFBAUES ZU BETREIBEN. LETZTERER ENTFÄLLT ZUR RUNTIME.

; PROGRAMMBEGINN

```
6950      VOUT      LDA # 128      ; BILDSCHIRMINHALT IN SPEICHER
          LDX # 28
          JMP VIDEX
```



Arbeiten mit VIDEOEDIT

## 1) Laden von VIDEOEDIT

Laden Sie VIDEO2 und VIDEOEDIT in den Arbeitsspeicher des PET.  
Mit dem Befehl SYS (7034) wird das Programm initialisiert.

## 2) Editing eines Bildschirminhaltes.

Mit SYS (6990) springen Sie in den EDITOR - MOD. Der PET nimmt alle Chr. über Tastatur an. Ebenfalls werden alle Cursorsteuerungen von der Maschine erkannt.  
Mit -SHIFT RETURN- können Sie den EDIT - MOD wieder verlassen.  
Der Bildschirminhalt ist jetzt extra abgespeichert.

## 3) Aufruf des Bildschirminhaltes.

Mit SYS (6957) wird der der geschützte Bildschirminhalt wieder auf den Bildschirm zurückgeschrieben.

## 4) Schützen eines Bildschirminhaltes.

Mit SYS (6950) wird der Inhalt des Bildschirms gesichert.

## 5) DUMP MEMORY ON TAPE # 1.

Der DUMP muß erst noch entwickelt werden, es gibt aber schon eine recht günstige Methode über das Betriebssystem des PET einen DUMP zu realisieren.

Wenn Sie ein Programm + Video - Driver + VIDEOEDIT + ein Hintergrundbild auf Kassette übernehmen möchten verfahren Sie wie folgt:

Die erste Programmzeile lautet:

```
1 POKE 124,PEEK(7050):POKE 125, PEEK(7051):CLR:POKE135,27
```

Das Abspeichern geschieht wie folgt:

```
VERIFY "Filename"
```

Bei "PRESS PLAY ON TAPE # 1" müssen Sie die STOP-Taste drücken. Danach müssen folgende POINTER gesetzt werden:

```
POKE 7050 ,PEEK(124): POKE 7051,PEEK(125)
POKE 247,58:POKE 248,3:POKE 229,245:POKE 230,31
Abgespeichert wird dann mit SYS(63153).
```

Der gesamte Bereich kann dann mittels LOAD wieder geladen werden.  
Der PET schreibt und lädt etwa 2,5 Minuten.

*Anmerkungen des Herausgebers: Hinsichtlich der eleganten Abspeicherung von Maschinenprogrammen auf Cassette sei auf die neueren Entwicklungen verwiesen, über die im Artikel 'Der PET-Assembler (2)' berichtet wird.*

## PET - PETITS

## REM-DELETER

Herr Karlheinz Lehner, Zeppelinstr. 29, 6236 Eschborn 1 sandte dieses kleine Dienstprogramm, das REM-Zeilen aus einem BASIC-Programm entfernt und damit Platz schafft. Man fügt den REM-DELETER am besten an das zu editierende Programm an und gibt RUN ab entsprechender Stelle.

```

10 DEFFNA(X)=PEEK(X)+256*PEEK(X+1)
20 Z=1025
25 IF PEEK(Z)=0 AND PEEK(Z+1)=0 THEN END
30 IF PEEK(Z+4)<143 THEN Z=FNA(Z):GOTO25
35 PRINT "♥FNA(Z+2):PRINT"RUN"           ♥ = Clear Screen
40 POKE527,145:POKE528,145:POKE529,145:POKE530,145:POKE531,13:POKE526,6:END

```

## Der PET-Assembler (2)

Dieser von Herrn Lehner verfaßte Assembler erfährt in gemeinsamen Bemühen weitere Abrundungen. Den Einsendern sei an dieser Stelle herzlich gedankt. Zunächst Herr Lehner: Bei Befehlen wie JMP und JSR zu einem Ziel in der Zeropage des Speichers stieg der Assembler bisher aus. Die Zeile 5110 muß daher wie folgt ergänzt werden:

```
5110 IF V<256 AND LEFT$(OP$,1)<>"J" THEN AD=2:GOTO5910
```

Zum Kommagebrauch bei INPUT: Unter Benutzung des Tastaturpuffers (527-536) läßt sich automatisch ein Anführungszeichen in einem INPUT eingeben. Die folgende Zeile soll dann unmittelbar auf einen INPUT folgen  
POKE 527,34:POKE 525,1. Während der Verarbeitung kann man dann Texte "blind" eingeben.

Die vom Herausgeber in Heft 7, S. 37 vorgeschlagene Form der Abspeicherung von Maschinenprogrammen konnte nur ein Provisorium sein. Ehe wir die von Herrn Kornnagel entwickelte sehr schöne Methode darlegen, noch Hinweise von Herrn Lehner zur Speicherung in DATA-Files: Wenn die Menge der wiedereinzulesenden Daten nicht genau deklariert ist, läßt man entweder unvollständig oder der PET hängt in einer Warteschleife auf, weil er auf 'mehr' wartet. Man verwende jungfräuliche Bänder, um nicht versehentlich auf Datenreste zu stoßen. Nach dem englischen PET User Manual schafft man wie folgt Raum zwischen records: POKE59411,53. Die Zeilen ab 8151 werden daher wie folgt zusammengefaßt

```
FOR Q=AN TO PC-1:PRINT#1, PEEK(Q):POKE59411,53:NEXT Q
```

Beim Wiedereinlesen hat man die zusätzliche Kontrolle, daß die letzte Zahl die Variable ST zu 64 setzt. Der Dataloader wird wie folgt geändert:

```
63090 INPUT#1,X:IF ST=64 THEN CLOSE1: ... FOLGESTATEMENTS
63100 POKE AN,X:PRINT AN,X:AN=AN+1:GOTO63090
```

Wie gelangt man nun zu 'selbstladenden' Programmen in Maschinsprache? Zu Programmen, die von den Laderoutinen des PET so geladen werden können wie ein BASIC-Programm? Im Zusammenhang mit Heft 7, Seite 37 gibt hier Herr Kornnagel folgende elegante Lösung, die beim Herausgeber erfolgreich nachvollzogen werden konnte:

```
8140 INPUT"FILENAME";FM$:AC%=256*PEEK(125)+PEEK(124):REM POINTER TO START
OF VARIABLES
8150 POKE247,WL%:POKE248,WH%:POKE229FNB(PC+1):POKE230,FNA(PC+1)
```



## GEDANKEN ZUM VIDEO-AIM

Michael Zimmermann, Eberstädter Str. 170, D-6102 Pfungstadt

E: The author defines the requirements for a TV-display to be utilized on an AIM 65, the best suiting display formats and examines several video-controllers for their aptness to work with AIM. For the most fitting MC6847 and the more versatile MC6845 he presents his tested hardware circuits and the supporting software to drive the TV display.

## 1. Einleitung

Mit Video-AIM bezeichnet der Verfasser einen AIM, dessen Ausgabeeinheit neben anderen ein Bildschirm ist. Man mag fragen "wozu ein Bildschirm?", der AIM hat doch eine 20-stellige alphanumerische Anzeige und einen Drucker und bietet damit doch viel bessere Voraussetzungen für die Ausgabe größerer Datenmengen als andere Systeme. - Gleichwohl ergibt sich besonders bei der Programmierung in BASIC eine starke Einschränkung durch die hierfür zu geringe Breite des Display mit seinen 20 Zeichen. Aber auch in vielen anderen Fällen möchte man mehr als 20 Zeichen auf einen Blick erfassen können und dabei nicht den Weg über den Drucker gehen, weil die Information nur kurzzeitig von Belang ist. Für diese genannten Fälle werden in diesem Artikel Entwicklungskriterien für eine Video-Ausgabe dargestellt.

## 2. Bildschirmformate

Dem Verfasser sind in seiner betrieblichen Praxis viele Bildschirmformate begegnet. Ihre Zeichenzahl variiert etwa um den Faktor 10. Generell ist bei der Programmierung eine größere Zeilenbreite erwünscht als bei der Datenerfassung, bei der in einer Zeile lediglich ein Merktext ausgegeben und die Eingabe eines Wertes erwartet wird. Die bei der Programmierung angestrebte große Zeilenbreite kann durch eine große Zeilenzahl und die Möglichkeit kompensiert werden, eine Programmzeile in mehreren Bildzeilen abzubilden.

Hier nun einige Formate und ihre Einsatzgebiete, als Ausgangspunkt für einen Video-AIM:

Zeichen	Format	Maschine	Einsatzgebiet
240	6x40	IBM 3741	Datenerfassung, nach Erfahrung des Verfassers etwas schmalbrüstig und auch bei den Erfassern nicht beliebt.
320	8x40	NIX 8811	Datatelefon, für einfache Abfragen und Erfassungen gerade ausreichend.
256	8x32	RUF 90	für Datenerfassung zu klein
480	12x40	NIX 8820	für Datenerfassung befriedigend, für Programmentwicklung gerade ausreichend.
512	16x32	diverse	in den USA beliebtes 'kleines' Bildschirmformat, durch größere Zeilenzahl auch für Programmentwicklung gerade ausreichend, durch Binärzähler einfach zu realisieren.
960	12x80	NIX 8820	guter Kompromiß zwischen Bildschirmbreite und Zeichenzahl, für Programmentwicklung gut.

---

## 65<sub>xx</sub> MICRO MAG

---

1000	25x40	PET 2001	für Programmentwicklung durch große Zeilenzahl befriedigend.
1024	16x64	diverse	in den USA beliebtes 'grobes' Bildschirmformat, für Programmentwicklung sehr befriedigend.
1920	24x80	diverse	Standardbildschirm in der 'großen' Datenverarbeitung, durch große Zeilenzahl und große Zeilenbreite für Programmierung keine Wünsche offen lassend.

Aus den oben kurz dargestellten Formaten ergibt sich, daß eine Bildschirmgröße zwischen 512 und 1024 Zeichen für die Programmierung ausreichend ist, ohne daß der Bildschirm zu groß wird und einen zu großen Aufwand erfordert, der naturgemäß mit der Bildschirmgröße wächst.

Wegen einfacherer Entwicklung sollte das Format eine 'gerade' Zahl sein. Als Zeilenbreiten kommen hierfür infrage: 32, 40, 48, 64 und 80 Zeichen. Günstige Zeilenzahlen sind: 8, 12, 16, 20 und 24 Zeilen. Für die endgültige Auswahl eines Bildschirmformates sind weitere Überlegungen erforderlich, die im nächsten Abschnitt erarbeitet werden sollen.

### 3. Einschränkungen

Es sollte Ziel der Entwicklung sein, das erzeugte Bildsignal auf einem Standard-Fernsehgerät wahlweise über den Video-Eingang oder über den Hochfrequenzeingang einspeisen zu können. Hierdurch ergibt sich eine Reihe von Einschränkungen, bedingt durch die vorherrschenden Fernsehnormen. In Deutschland hat das Fernsehbild eine Bandbreite von 4 MHz bei Farbgeräten. Das bedeutet: Bei einer Frequenz von mehr als 8 MHz ist für den einzelnen Bildpunkt keine befriedigende Darstellung mehr möglich (!).

Bei einer Zeilendauer von 64  $\mu$ sec sind nur etwa 2/3 für eine echte Informationsausgabe nutzbar, das entspricht 42  $\mu$ sec. Da ein Zeichen in der Horizontale aus mindestens 5 Bildpunkten und einem Zwischenraum besteht, ist eine Breite von 56 Zeichen möglich. - Für die amerikanische Fernsehnorm mit Bandbreite von nur 3 MHz liegt die Grenze bei 40 Zeichen/Zeile.

Ähnliche Überlegungen gelten für die Zeilenzahl. Bei 625 Zeilen je Bild ergibt sich durch Überlappung zweier Halbbilder eine echte Anzahl von 312 Zeilen, von denen wiederum nur 2/3 ausnutzbar sind. Bei einer Zeilenzahl von 7 pro Zeichen plus eine Zeile für den Zwischenraum sind 26 Zeilen möglich, so daß selbst größte Bildformate auf einem unmodifizierten TV-Gerät möglich wären. - Auch hier hat die amerikanische NTSC-Norm wegen ihrer 550 Zeilen ungünstigere Werte. Für sie können 22 Zeilen als Grenze angesehen werden.

Die vorstehenden Werte beruhen auf recht optimistischen Annahmen über das Betriebsverhalten der Fernsehgeräte. Zuma bei billigeren Geräten sollte man nicht von einer Ausnutzung von 2/3 der Bildfläche ausgehen, 1/2 dürfte hier realistischer sein. Zeichen mit nur einem Zwischenraum sowohl in horizontaler als auch vertikaler Richtung sind alles andere als gut lesbar. Ein Bildschirm mit weniger aber dafür gut lesbaren Zeichen dürfte allemal mehr bringen, als ein bis zu seiner theoretischen Grenze ausgereizter.

*Damit ergeben sich für die Zeilenlänge 64 oder 48 Zeichen und für die Zeilenzahl 16 bis 20, die nach europäischer Norm auf einem unmodifizierten Fernsehgerät darstellbar sind. Nach amerikanischer Norm dürfte es schwierig sein, mehr als 16x32 Zeichen auf den Bildschirm zu bringen.*

Auf Bildschirmgraphik wird hier bewußt verzichtet, 64 Zeichen, wie im AIM-Display, sollten als Zeichensatz ausreichen, und auch auf Farbe, entsprechend dem Motto 'if you want colours, buy crayons'. Wie auch bei der Hardware wird bei der Software auf Besonderheiten verzichtet. Sie soll einfache, aber wirkungsvolle Unterstützung gewähren.

Unabhängig vom endgültigen Format soll die Software folgende Funktionen realisieren:

1. Initialisieren des kompletten Video-Systems inkl. Umsetzen von DILINK (s. Heft 7) und Leeren des Bildschirminhaltes,
  2. Schreiben eines Zeichens in das Video-RAM in die Position des Cursors mit Weiterschalten des Cursors. Es soll immer nur in die letzte Zeile geschrieben werden. Ist diese voll, so rückt der Bildschirminhalt um eine Zeile nach oben und der Cursor springt in die erste Schreibstelle der neuen (noch leeren) Zeile.
  3. Erkennen und Ausführen eines Zeilenvorschubes durch Hochrücken des Bildschirminhaltes (scrolling).
4. Anschluß des Video-Teiles an den AIM

Dem Verfasser sind drei prinzipielle Möglichkeiten bekannt, einen Video-Teil an den AIM anzuschließen:

1. Über einen seriellen Ausgang,
2. Über einen parallelen Ausgang,
3. Als RAM, dessen Inhalt auf dem Bildschirm angezeigt wird.

Von diesen Möglichkeiten dürfte lediglich die dritte interessant sein, da die anderen nicht so schnell sind und auch nicht im entfernten die Möglichkeiten der Einflußnahme auf den Bildaufbau haben, wie die einer direkten RAM-Ausgabe. Aus diesen sicher guten Gründen sollten die beiden ersten Möglichkeiten nicht weiter beleuchtet werden.

Unabhängig vom hardwaremäßigen Anschluß ist die Verbindung des Video-Teiles mit dem Monitorprogramm zu betrachten. Die Entwickler des AIM-Monitors sind für ihre voraussehende Planung zu loben, die Zeichenausgabe auf dem LED-Display grundsätzlich mit einem indirekten Sprung mit Vektor im DILINK auszuführen. Es ist lediglich erforderlich, diesen Vektor, der bei der Systeminitialisierung auf OUTDIS gesetzt wird, so zu ändern, das jedes auszugebende Zeichen an der richtigen Stelle im Video-RAM abgelegt wird (s.a. Heft 7, Seite 35).

Wir fassen bis hierher noch einmal die Anschlußbedingungen zusammen:

*Der Video-AIM besteht aus einem Video-RAM, dessen Inhalt auf einem standardmäßigen Fernsehbildschirm angezeigt wird. Dafür ist ein Programm erforderlich, daß den Vektor DILINK auf eine Routine richtet, die ein Zeichen im Video-RAM ablegt, die evtl. erforderliche Verschiebungen (Scroll) durchführt und die die Kontrollzeichen der Bildschirm-Editierung erkennt und ausführt.*

5. Hardwaremäßige Realisierung des Video-RAMs und Auswahl des Video-Controllers.

Der erste Ansatz des Verfassers zur Realisierung war in der 'klassischen' Art: Ausgerüstet mit einer Vielzahl von Zählern und Multiplexern wurde ein

Speicher angesprochen, einmal direkt vom Prozessor, das andere Mal vom Video-AdreBregister, das mit der Zeichenfrequenz hochgezählt wurde. Das Ergebnis war eine doch sehr kleine 16x32-Zeichen-Ausgabe, die mit einem unverhältnismäßig großen Aufwand an Hardware erreicht wurde (TTL-Schaltkreise) und mehr zum Eierbacken und als Rechtfertigung für das nächste notwendige Atomkraftwerk eignete, denn als Video-RAM. Falls trotzdem an einem derartigen Schaltkreis, der nichtsdestoweniger viele Einsichten über die Wirkungsweise von Video-RAMs vermittelt, Interesse besteht, sei auf (9) verwiesen.

Eine logische Weiterentwicklung ist die vollständige Kontrolle der Signalerzeugung durch den Prozessor, wie von Don Lancaster (2) dargestellt. Dieser Weg scheidet für den AIM aus (im Gegensatz zum KIM), weil der hierfür erforderliche AdreBraun nicht vorhanden ist. Eine Weiterentwicklung der Zählerketten, verbunden mit Bereitstellung weiterer Funktionen, stellen die sogenannten Video- oder Cathode Ray Tube Controller (CRT) dar. Für einen entsprechenden Hinweis darf der Verfasser dem Herausgeber von 65<sub>xx</sub> MICRO MAG danken.

Weil diese Bausteine in Sichtgeräten einen immer bedeutenderen Platz einnehmen, soll in einem Exkurs näher auf sie eingegangen werden. Dabei erfolgt eine Auswahl der Einsatzmöglichkeiten am AIM. Als Vertiefung sei auf (3) hingewiesen, wobei der Verfasser betonen möchte, daß seine Ideen vollkommen unabhängig und ohne Wissen über diese Arbeit entwickelt wurden (Anmerkung des Herausgebers: Der Autor erhielt erst durch den Hrsg. entspr. Hinweise auf den Artikel). Das sehr ähnliche Ergebnis sollte in beiden Fällen auf ein den Anforderungen sehr angepaßtes Design deuten.

#### 6. Exkurs: Kurze Darstellung und Auswahl von Video-Controllern

Dieser Abschnitt soll dem Leser, ohne Anspruch auf Vollständigkeit, eine Übersicht über die erhältlichen Video-Controller geben und ihre Verwendbarkeit am AIM 65 beurteilen.

##### a) SF.F96364

Dieser Baustein der Firma Sescosem, der auch im Elekterminal (4) Verwendung findet, ist für bitparallele Zeichenübertragung in Zusammenarbeit mit einem UART ausgelegt. Hieraus resultieren die recht langen Zeiten für das Schreiben in den Speicher. Wegen der abweichenden Konzeption im Vergleich zum Video-AIM und weil er nur von einem Hersteller angeboten wird, scheint er weniger geeignet.

##### b) DP8350 (National) (3)

Bei diesem Baustein sind die Anzeigeparameter in einem Mask-PROM abgelegt. Er ist dadurch starr und dürfte deswegen nur in größeren Stückzahlen für professionelle Anwender interessant sein. Eine nähere Betrachtung soll daher nicht erfolgen.

##### c) SMC 5027 (3)

Dieser Chip scheidet aus der eingehenden Besprechung aus. Er benötigt zwei Spannungen, hat einen hohen Preis, außerdem entstehen Lücken im ansprechbaren Speicher bei nicht binären Bildschirmformaten.

##### d) Intel 8275 (10)

Dieser Controller verfügt über eine Vielzahl von Features und soll mit dem 8257 DMA-Controller zusammenwirken. Wegen seiner Ausrichtung auf die 8080-Architektur ist er nicht als Video-AIM geeignet. Es sei soviel gesagt, daß er zwei 80-stellige FIFOs enthält, von denen der eine die gerade ausgegebene Zeile aufnimmt und der andere die nächste.

e) MC 6847, S68047 (7), (8), (11)

Mit diesem Schaltkreis haben wir im Rahmen unseres Exkurses das erste Mitglied der 6800-Familie vor uns. Er ist mit den 65<sub>xx</sub>-Prozessoren kompatibel, ein Grund, diesen Controller näher zu betrachten. Seine Möglichkeiten für Graphik und Farben sollen hier nicht näher erörtert werden. Es sei aber erwähnt, daß ein vollständiges Farbsignal erzeugt wird.

Für uns ist interessanter, daß der 6847 im Alpha-Mode einen Speicher von 512 Bytes adressiert (16x32) und 'on-chip' einen Character-Generator sowie ein Schieberegister enthält. Er gibt ein Komplementär-Videosignal ab, dem alle Syncs bereits aufgeprägt sind. Als Nachteil ist zu erwähnen, daß die Video-Signale auch hinsichtlich Zeilenzahl und Chroma-Frequenz der amerikanischen Norm entsprechen. Für einen Anwender, der bewußt mit den Beschränkungen des 16x32-Formates zu leben bereit ist und der seinen Fernseher in der Zeilenfrequenz entsprechend umstellen kann, dürfte dies der gegebene Video-Baustein sein, weil auch die Bauteilezahl in der Video-Schaltung durch den integrierten Character-Generator sehr gering wird. Im pin-out besteht ein gewisser Unterschied zwischen den sonst gleichen MC6847 und S68047.

f) MC 6845, R6545 (5), (6)

Für dieses Bauteil gelten als Mitglied der 6800-Familie gleiche Überlegungen wie für den MC6847. Im Gegensatz zum vorgenannten ist dieser Chip aber mit einer Vielzahl von Kontrollmöglichkeiten ausgestattet, die dem 6847 fehlen. Im Gegenzuge fehlen hier Character-Generator und Schieberegister, die extern aufzubauen sind.

Besonderes Kennzeichen dieses Chips sind 17 Register, die die Kenndaten einerseits des Fernsehbildes (Zeilenlänge, Zeilenzahl) und zum anderen des anzuzeigenden RAM enthalten (ebenfalls Zeilenlänge und Zeilenzahl, Zahl der Zeichen pro Zeile u.v.a.). Eine auf diesem Controller basierende Video-Schaltung wird dadurch sehr vielseitig und kann bei gleichem Hardwareaufbau einer Vielzahl von Bildformaten und auch Gerätetypen angepaßt werden. - Einzige Bedingung ist, daß die entsprechenden Register mit den gewünschten Werten geladen werden.

Zur Vervollständigung sei erwähnt, daß der R6545 noch weitere Register enthält, die den Status angeben und die das Einspeichern von Werten in das Video-RAM mit geringerem Hardwareaufwand ermöglichen.

## 7. Beschreibung einer Video-Schaltung mit dem 6847

### a) Hardware

Einen Video-Controller wie den 6847 kann man sicherlich isoliert beschreiben. Wir folgen dem wohl besseren Ansatz, den Controller im Zusammenspiel mit den anderen Komponenten einer Schaltung darzustellen. Eine Übersicht der bereits im 6847 integrierten Funktionen sind dem Diagramm aus (7) in Abb. 1 zu entnehmen. Im einzelnen: Uhr und Herzschlag des gesamten Systems ist ein Quarz mit der NTSC-Chromafrequenz von 3,579545 MHz. Dieser Quarz wird mit einer konventionellen Schaltung mit 3 Invertern getrieben und gepuffert. Er erzeugt die Zeitbasis einerseits für die Sync-Signale nach NTSC-Norm und andererseits für die Zugriffe zum Video-RAM und dessen Abrasterung. Der Controller adressiert dabei jeweils 32 fortlaufende Bytes nacheinander und führt deren Inhalt dem Zeichengenerator und dem Schieberegister zu. Dieser Vorgang läuft für jede Datenzeile 12mal ab, entsprechend der Anzahl von Bildschirmzeilen, aus denen sich eine Datenzeile aufbaut.

Zwischen den einzelnen Durchläufen einer Datenzeile ist genug Zeit, in der

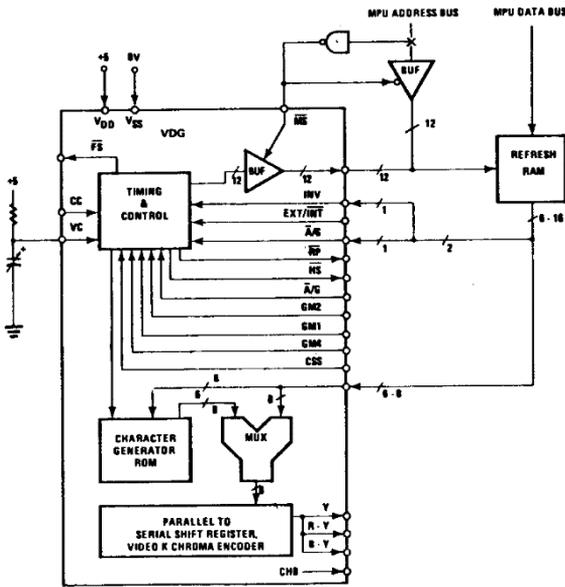


Abb. 1: Blockschaltbild MC 6847

keine Austastung des RAM vorgenommen wird, um dem Bildschirm den Rücksprung auf eine neue Bildschirmzeile ohne Datenverlust zu ermöglichen (Retrace).

Dieser Vorgang läuft für alle 16 Datenzeilen eines Bildschirm-inhaltes nacheinander ab und beginnt dann wieder von vorne. Zwischen den einzelnen Bildern ist auch hinreichend Leerzeit für einen Vertical Retrace.

In diese Austastlücken fallen die entsprechenden Sync-Signale.

Bisher haben wir nur über die Aktivitäten des Controllers berichtet. Sinn der Videoschaltung ist es aber, Informationen vom Prozessor an den Benutzer zu übermitteln. Dies wird dadurch bewirkt,

daß der Video-Speicher gleichermaßen vom Prozessor und vom Controller angesprochen werden kann und daß ein Zeichen, daß der Prozessor in diesen Speicher einschreibt, vom Controller beim nächsten Speicherdurchgang zur Anzeige gebracht wird.

Der Speicher ist quasi Diener zweier Herren. Damit entsteht das Problem, wer wann an das RAM darf. Diese Konkurrenz wird dadurch gelöst, daß die Busse des Controllers und des Prozessors voneinander getrennt werden. Dadurch können beide den Speicher unabhängig voneinander adressieren und Werte auslesen oder einschreiben (shared memory). Von der Prozessorseite wird die Trennung durch Daten- und Adreßtreiber vorgenommen, von der Controllerseite durch die Fähigkeit, die Adreßausgänge auf High Impedance zu setzen. Normalerweise wird der Speicher jetzt durch den Controller adressiert. Nur wenn der Prozessor in das RAM einschreiben oder dort auslesen will, werden die Ausgänge des Controllers inaktiviert, und der Prozessor kann seine Operation ausführen.

Während dieser Zeit laufen die internen Vorgänge des Controllers ohne Unterbrechung weiter, insbesondere das Abrastern des RAM (die Adressen werden nicht effektiv) und die Zeichenerzeugung zusammen mit der Zeichenausgabe. Dieser Vorgang ist insoweit unproblematisch und für den Benutzer nicht sichtbar, wenn er in der Austastlücke des Video-Bildes stattfindet. Lediglich beim Ansprechen des RAM im aktiven Bildteil kann es durch Fehlauftastung des RAM zu kurzzeitigem Schnee auf dem Bildschirm kommen. Das kann aber toleriert werden, da der Hard- und Softwareaufwand zur Verlegung aller Zugriffe in die Austastlücke in keinem Verhältnis zu den rein optischen Verbesserungen steht.

# 65. MICRO MAG

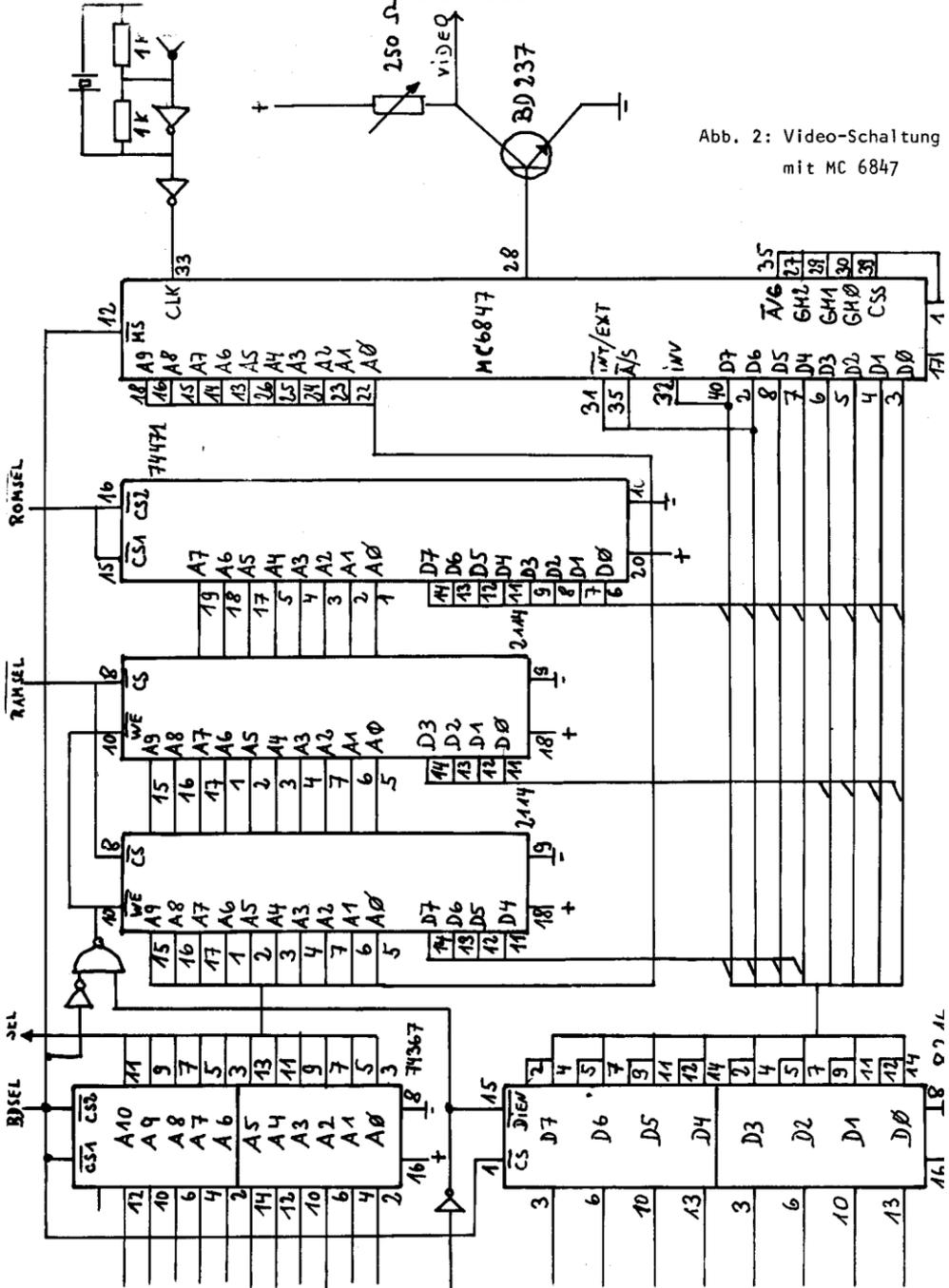


Abb. 2: Video-Schaltung mit MC 6847

---

## 65<sub>xx</sub> MICRO MAG

Als Video-RAM wurden 2 Speicher 2114 gewählt, zusätzlich ist ein ROM vorgesehen, das die erforderlichen Routinen für die Zugriffe auf das RAM enthalten kann, insbesondere für die Formatierung in Zeilen etc.. Solche Programme können selbstverständlich auch an anderer Stelle in einem ROM oder RAM stehen. Es dürfte eine gute Lösung sein, diese Programme in den noch freien Teil des Video-RAMs zu laden.

Bisher haben wir die Dekodierung von RAM und/oder ROM besprochen und die Zuweisung von Adressen. Die einfachste Methode, der auch der Verfasser gefolgt ist, besteht darin, einen auf dem AIM bereits dekodierten 4k-Block, z.B. \$9xxx zu nehmen. Hierbei werden zwar 3k Adreßraum verschenkt, aber die wenigsten Benutzer dürften gezwungen sein, mit ihrem Adreßraum besonders ökonomisch umgehen zu müssen. Alternativ könnte man das Display auf dem AIM entfernen und das Video-RAM in den Bereich AC00-AFFF legen. Hier ist dringend anzuraten, die Betriebsprogramme in einem ROM unterzubringen, da der AIM in diesem Falle blind hochgefahren werden muß. Für dieses ROM wird ein Teil des Adreßraumes des RAM geopfert. Man kann das im Interesse einer geringeren Chipzahl in Kauf nehmen.

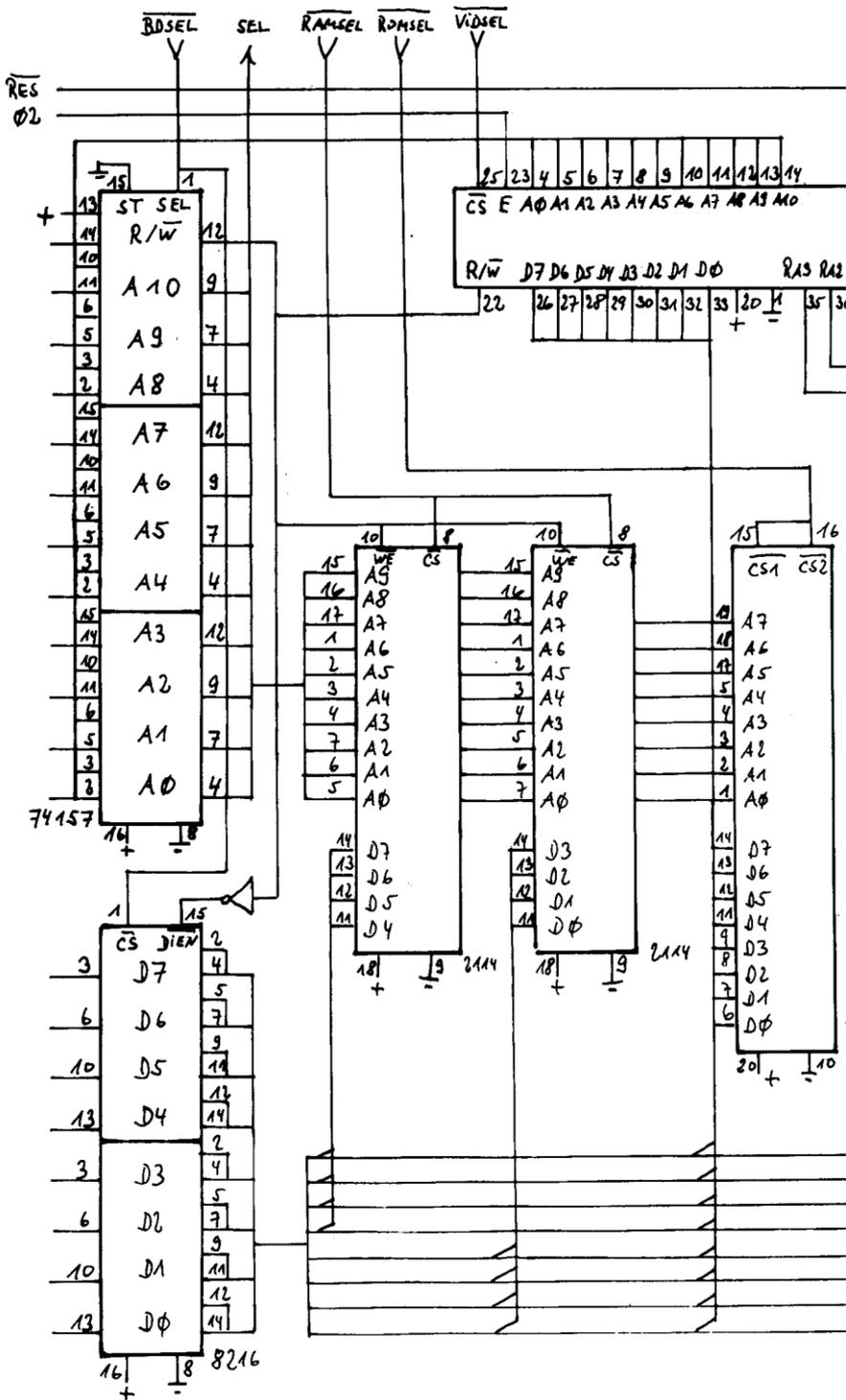
In der vorliegenden Zeichnung, Abb. 2, wurde ein bipolares PROM gewählt, da die erforderlichen Routinen kurz sind und weil dieser Promtyp für diese Größe am kostengünstigsten ist und außerdem nur eine Versorgungsspannung fordert. Legt man diesen Speicher mit seinen Treiberrountinen für das Video-RAM in jenen Speicherteil, der von Controller und Prozessor gemeinsam betrieben wird, so führt das wegen der häufiger notwendigen Zugriffe des Prozessors zu diesem Bereich auch zu mehr Fehlauastastungen. Nach den Erfahrungen des Verfassers ist auch das tolerierbar, da bei dieser Betriebsweise nur kurzzeitig Schnee auf dem Bildschirm erscheint, der nach Einschreiben oder Auslesen eines Zeichens sofort wieder verschwindet. Bei der Abfassung der Treiberrountinen muß man aber in jedem Fall vermeiden, daß sie in eine endlose Schleife im Video-RAM führen.

Der einfachste Teil der Schaltung ist der eigentliche Video-Treiber. Beim Verfasser reichten ein NPN-Transistor und ein Widerstand aus, um das invertierte Composite-Video-Signal umzudrehen und um ein gutes Bild bei Einspeisung in einen Monitor zu erzeugen. Zusätzlich sei auf den Chip MC1372 hingewiesen, der das Signal des 6847 modulieren kann und es damit zur Einspeisung in einen unmodifizierten Fernseher aufbereitet - dieses Gerät muß freilich der NTSC-Norm entsprechen.

Da der on-chip Character Generator des 6847 nur 64 Zeichen darstellen kann, werden nur 6 Bit eines jeden Byte im Video-RAM benötigt. Es liegt daher der Gedanke nahe, die restlichen 2 Bit für weitere Möglichkeiten des 6847 auszunutzen. Bit 6 wird hierbei als Schalter für Semi-Graphikdarstellung benutzt und Bit 7 als Inverterschalter im alphanumerischen Mode. Einzelheiten sind der Abb. 3 zu entnehmen. Auf die Darstellung weiterer Möglichkeiten, full graphic und Farbe, wurde verzichtet.

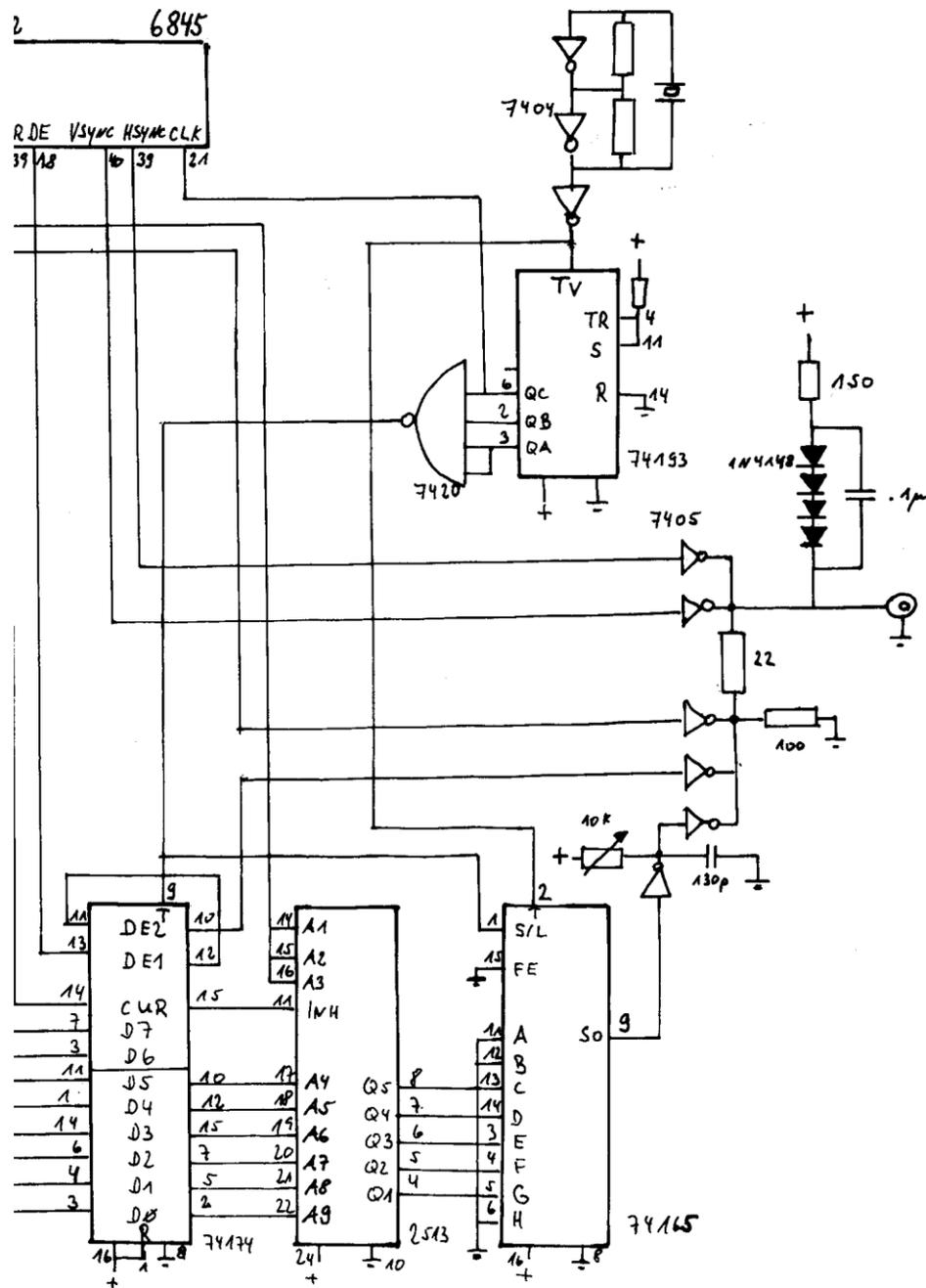
### b) Software

Welche Software ist nun erforderlich, um die Schaltung gem. Abb. 2 zu betreiben? Übereinstimmend mit dem Verfasser werden Sie wie bei der Hardware feststellen, es ist erstaunlich wenig.- Zuerst eine Anmerkung zum Format: Der Bildschirm soll nur eine Eingabezeile haben, die letzte. Alle Daten werden in diese Zeile eingegeben. Ist sie gefüllt oder wurde ein Zeilenvorschub ausgelöst, so wird der gesamte Bildschirminhalt nach oben verschoben (scroll). Dabei geht die bisherige oberste Zeile verloren die letzte Zeile wird für weitere Eingaben gelöscht.



## 65xx MICRO MAG

Abb. 4: Video-Interface mit MC 6845 (BDSEL=Boardselect)



D7	D6	D5	D4	D3	D2	D1	D0						
0	0	normale alphanum. Char.											
1	0	invertierte alphanum.Ch.											
0	1	Semigraphic-6											
1	1	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>D5</td> <td>D4</td> </tr> <tr> <td>D3</td> <td>D2</td> </tr> <tr> <td>D1</td> <td>D0</td> </tr> </table> <p style="margin-left: 40px;">Dn=0: schwarz Dn=1: weiß</p>						D5	D4	D3	D2	D1	D0
D5	D4												
D3	D2												
D1	D0												

Abb.3: Zeichen-Modes des 6847

Diese Form der Eingabe ähnelt derjenigen an einer Schreibmaschine, bei der man lediglich die letzten 16 Zeilen im Blickfeld hat. Nach Ansicht des Verfassers ist diese Formatierung der beste Kompromiß zwischen geringem Aufwand an Programmierung und sinnvoller Menge dargestellter Information.

Wegen Einzelheiten sei auf die dokumentierten Programme hingewiesen. In ihnen liegt der Bildschirmspeicher im Bereich 9000-91FF und das Kontrollprogramm bei 9300-93FF. Ebenso wie die Schaltung erheben auch diese Programme keinen Anspruch auf besondere Schönheit oder Eleganz, sie erfüllen aber ihren Zweck. Zur Übung des Lesers sind sie als Assemblerliste dargestellt.

Es ist jedem Benutzer überlassen, für seine Anforderungen entsprechende Routinen zu schreiben, die den Bildschirm als Gesamtheit betrachten und entsprechend ausgeklügelte Verarbeitungen ermöglichen. Hierbei wird dem Benutzer das Leben sehr leicht gemacht, weil jedes Zeichen auf dem Bildschirm eindeutig einer Speicheradresse entspricht.

Besonders ist das Umsetzen des Pointers im DILINK durch das Initialisierungsprogramm hervorzuheben. Hierdurch wird bewirkt, daß die Bildschirmausgabe in die Kette der Ausgabeprogramme eingefügt wird und daß jedes Zeichen, das auf dem AIM-Display erscheint, auch auf dem Bildschirm ausgegeben wird.

#### c) Zusammenfassung für MC 6847

Die dargestellte Video-Schaltung mit 6847 erscheint als eine gute Lösung für diejenigen Benutzer, die über den normalen AIM hinaus keinen großen Wert auf Editor, Assembler und BASIC legen. Durch die Zeilenlänge von 32 Zeichen wird das normale Ausgabeformat des AIM von 20 Zeichen in jedem Falle auch auf dem Bildschirm darstellbar.

Für anspruchsvollere Anwendungen in Richtung auf umfangreiche Editierung sind bei diesem Controller Grenzen gesetzt, da evtl. die Zahl der gleichzeitig auf dem Schirm darstellbaren Zeichen zu gering ist, um diese Arbeiten hinreichend komfortabel und ohne zuviel Umblättereie im Text durchführen zu können.

```

-----
DRIVER ROUTINES FOR
MC 6847

==0000
;DESCRIPTION OF VAR
TABLES

==0000 VIDRAM
==*$9000
; ADDRESS OF
VIDEO-RAM
==0000 VIDROM
==*$9200
; ADDRESS OF
DRIVER ROUTINES

```

65<sub>xx</sub> MICRO MAG

```

==0000 LASLIN
      =VIDRAM+$1E0
; ADDRESS OF LAST LINE
E
==0000 LINLEN
      =$20
; LINE LENGTH
==0000 XSAV
      =$0170
; LOCATION TO SAVE
X-REG
==0000 CURPOS
      =$171
; ADDRESS OF CURSOR
POSITION
==0000 RSET
      =$E0BF
; ADDRESS OF MONITOR
RESET ROUTINE
==0000 DILINK
      =$A406
; OUTPUT-ROUTINE
VECTOR
==0000 OUTDIS
      =$EF05
; ADDRESS OF DISPLAY
OUTPUT ROUTINE
-----
INITIALISATION-ROUTINE
==0000
      *=VIDROM
==9200 INIT
      ***
A010 LDY #$10
; SET COUNT FOR CONSECUTIVE SCROLLS
==9202 I10
      ***
; SCROLL UNTILL SCREEN IS CLEARED
208092 JSR SCROLL

88 DEY
DOFA BNE I10
A940 LDA #<CHAR
; CHANGE OUTPUT ROUTINE ADDRESS
8D06A4 STA DILINK
A992 LDA #>CHAR
8D07A4 STA DILINK+1
==9212
4CBFE0 JMP RSET
; GO TO RESET-ROUTINE
E

```

```

-----
CHARACTER OUTPUT ROUTINE
==9215
      *=VIDROM+$40
==9240 CHAR
      ***
48 PHA
; SAVE A
8E7001 STX XSAV
; AND X TOO
297F AND #$7F
; MASK OFF HIGHEST BIT
AE7101 LDX CURPOS
; LOAD ACTUAL CURSOR POSITION
C90D CMP #$0D
; CHECK FOR CARRY
F00C BEQ CR
293F AND #$3F
; STRIP OFF BIT 6
9DE091 STA LASLIN,X
; STORE CHARACTER AT CURSOR POSITION IN LAST LINE
==9252
E8 INX
; INCREASE CURSOR POSITION
E020 CPX #LINLEN
; AND CHECK FOR END OF LINE
D00C BNE NOSCRO
; NO END OF LINE, NO SCROLL
F005 BEQ SCRO
; END OF LINE, SCROLL
==9259 CR
      ***
; PROCESS CARRY, CLEAR CURSOR-POSITION AND SCROLL
A920 LDA #$20
9DE091 STA LASLIN,X
==925E SCRO
      ***
; SCROLL IF CARRY OR END OF LINE

208092 JSR SCROLL
A200 LDX #$00
; CLEAR CURSOR-POINT ER AND SAVE IT

```

```

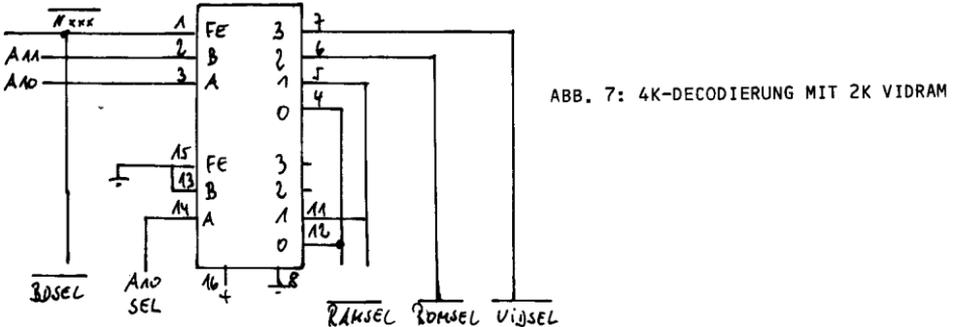
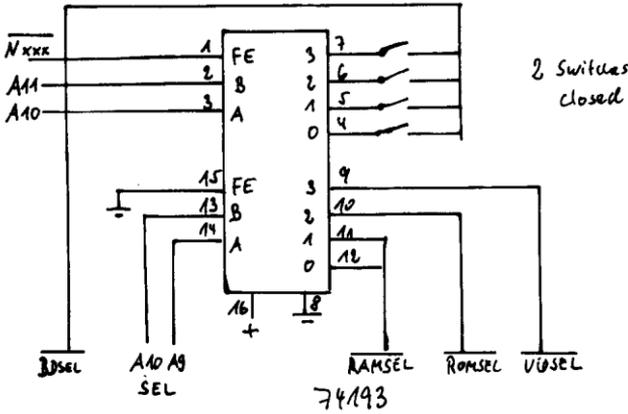
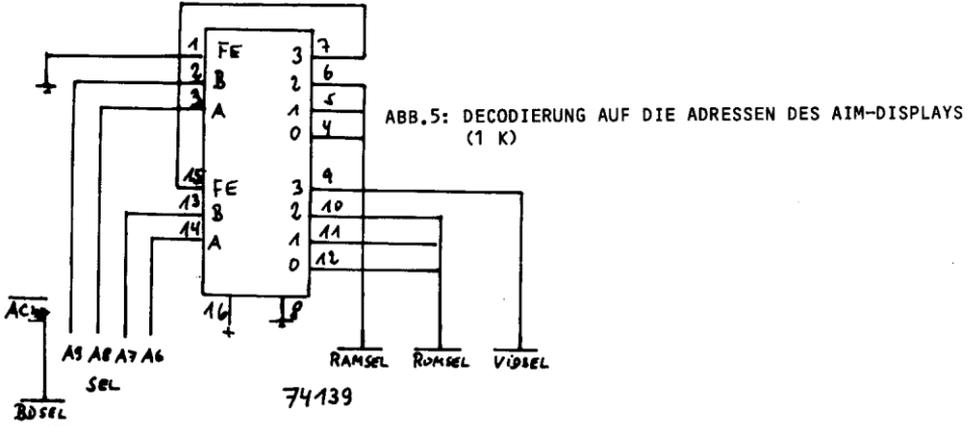
==9263 NOSCRO
      ***
8E7101 STX CURPOS
; SAVE CURSOR-POSITION
A9FF LDA #$FF
; LOAD CURSOR CHARACTER
9DE091 STA LASLIN,X
; AND STORE IT AT CURSOR POSITION
AE7001 LDX XSAV
; RELOAD SAVED X
68 PLA
; AND A
4C05EF JMP OUTDIS
; GO TO OUTPUT ROUTINE FOR DISPLAY
-----

```

```

SCROLL SCREEN AND CLEAR LAST LINE
==9272
      *=VIDROM+$80
==9280 SCROLL
      ***
A200 LDX #$00
==9282 S10
      ***
; SHIFT UP FIRST BLOCK OF VIDEO-RAM
BD2090 LDA VIDRAM+LINLEN,X
9D0090 STA VIDRAM,X
E8 INX
D0F7 BNE S10
==928B S20
      ***
; SHIFT UP SECOND BLOCK OF VIDEO-RAM
BD2091 LDA VIDRAM+LINLEN+256,X
9D0091 STA VIDRAM+256,X
E8 INX
D0F7 BNE S20
A920 LDA #$20
==9296 S30
      ***
; CLEAR LAST LINE OF VIDEO-RAM
9DE091 STA LASLIN,X
E8 INX
E020 CPX #LINLEN
D0F8 BNE S30
60 RTS
      .END
ERRORS= 0000

```



## 8. Aufbau einer Video-Schaltung mit dem MC 6845

## a) Hardware

Beim MC6847 ergaben sich durch das feste Bildschirmformat Einschränkungen, die in der nachfolgenden Schaltung überwunden werden. Die Vorteile der sehr freizügigen Gestaltung des Bildschirmformates werden allerdings mit einer höheren Chipzahl erkauft, da der MC6845 keinen Character-Generator 'on chip' hat. Unter Verzicht auf Grundlagen wird die Hardware hier nur insoweit beschrieben, als sich Unterschiede zur vorstehenden Schaltung ergeben.

Ausgangspunkt der Betrachtungen ist auch hier wieder der Quarz, der vom Verfasser mit 10 MHz gewählt wurde. Diese Frequenz bildet eine Grenze bei der Darstellung auf einem unmodifizierten Fernsehgerät. Sein Takt treibt das Shift-Register und gelangt andererseits auch auf einen Teiler durch 8. Dieser Teiler erzeugt die Zeichen-Taktimpulse für den Video-Controller und lädt in jedem 8. Takt auch das Data-Latch vor dem Zeichengenerator.

Die Taktfrequenz bewirkt im Video-Controller das Abrastern des Video-RAMs und ein Ansteuern der einzelnen Bildzeile im Zeichengenerator und die Erzeugung der Synchronisationsimpulse, die hier separat herausgeführt sind. Die Erzeugung des kompletten Videosignals erfolgt mit einer Schaltung, die (2) entnommen wurde. Für eine eingehende Beschreibung sei daher auf diese Quelle verwiesen.

Der Adreßbus des Controllers kann nicht direkt auf das RAM geführt werden, da die Ausgänge nicht hochohmig gesetzt werden können, vielmehr müssen die Adressen hier mit denen des Prozessors gemultiplext werden. Die dem RAM unter Steuerung des Controllers entnommenen Daten werden in einem Latch zwischengespeichert und über den Character-Generator dem Schieberegister zugeführt. Gleichzeitig werden das Cursor-Signal und das Display-Enable-Signal vom Controller mit diesem Latch verzögert, da diese durch die zweimalige Speicherung der Daten erst nach 2 Zeichenzeiten wirksam werden dürfen.

Wie im vorhergehenden Schaltungsbeispiel werden in der Schaltung nach Abb.4 normale TTL-Bausteine als Support-Chips verwendet, sie arbeitet aber auch mit TTL-LS-Bausteinen. Als Character-Generator wurde ein 2315 mit 5 V Spannung gewählt. Da hier in jedem Fall ein besonderer Zeichengenerator erforderlich ist, scheint es eine gute Idee, einen Baustein mit 128 darstellbaren Zeichen zu wählen. Da dem Verfasser ein solcher nicht verfügbar war und weil nur tatsächlich ausgeführte Schaltungen beschrieben werden sollen, wird auf die Betrachtung eines derartigen Bausteins verzichtet.

Wie in dem früheren Beispiel einer Videoschaltung (Abb.2), so sind auch hier wieder verschiedene Formen der Dekodierung von Speicher und Controller denkbar. Das Beispiel gem. Abb. 5 zielt auf den Fall hin, daß das AIM-Display ersetzt wird. Dabei ergeben sich keine weiteren Ansprüche an den Adreßraum, die nicht auch schon vom AIM erhoben worden sind. Darstellbar sind hier z.B. 16x48 oder 12x64-Formate.

Im Beispiel gem. Abb. 6 werden 2k benötigt, als Bildschirmgröße ist hier 16x64 das übliche Format.

Der dritte Fall (Abb. 7) erlaubt noch größere Bildschirmformate. Es ist z.B. ein 24x80-Format denkbar, hierfür sind aber weitere Speicher und ein Adreßraum von 4k erforderlich. Es bleibt aber zu bedenken: Diese Formate ergeben sich allein aus der Speichergröße und den Fähigkeiten des Controllers. Beschränkungen aus der Videobandbreite eines Monitors/Fernsehgerätes sind dabei unberücksichtigt.

## b) Software

Ebenso wie bei der Hardware sollen hier nur die Unterschiede zum MC6847 dargestellt werden. Wesentliches Merkmal und Schlüssel zur Flexibilität sind insgesamt 15 Register, in die geschrieben werden kann und die über ein Adreßregister indirekt adressiert werden. Ausgangspunkt für diese Register, die zumeist Zählerstände darstellen, ist die Zeichenclock, die dem Controller zugeführt wird. Diese Register enthalten:

R0	Anzahl der Zeichen in einer Bildschirmzeile des Monitors
R1	Anzahl der angezeigten Zeichen in einer Zeile
R2	Position des horizontalen Sync, bezogen auf R0
R3	Breite des horizontalen Sync
R4	Anzahl der Zeichenzeilen auf dem Bildschirm
R5	Adjustierung der Zeichenzeilen in Bildschirmzeilen
R6	Anzahl der angezeigten Bildschirmzeilen
R7	Position des vertikalen Sync, bezogen auf R4
R8	Schalter für Interlace-Mode
R9	Zahl der Bildschirmzeilen pro Zeichenzelle
R10	Erste Zeichenzeile des Cursors (Beeinflussung seines Aussehens)
R11	Letzte Zeichenzeile des Cursorzeichens
R12	Pointer High auf das erste anzuzeigende Zeichen des Video-RAM
R13	Pointer Low dito, Adresse, von der ab ausgelesen wird
R14	Cursorposition, Adresse High, Schreib- und Leseregister
R15	Dito, Adresse Low

Der vorstehenden Übersicht ist zu entnehmen, daß die Register zum einen Parameter für den Bildschirmmonitor enthalten, so z.B. R0, R2 und R3. Zum anderen beschreiben sie den Bildaufbau, wie R1 und R6. Und schließlich dienen R12 und R13 der Adressierung des Video-RAMs.

Wir geben ein Anwendungsbeispiel wiederum als Assembler-Listing. Es wird ein Bildschirm im Format 16x64 verwaltet.

## c) Zusammenfassung für MC 6845

Für den MC 6845 wurde eine universelle Videoschaltung beschrieben, die auf viele Benutzeranforderungen anpaßbar ist. Zeilenlänge und Zeilenzahl - und damit das Bildschirmformat - sind für viele gängige Formate einrichtbar. Bei entsprechend bemessenem Video-RAM und breitbandigem Bildschirmmonitor dürften kaum Wünsche offen bleiben, insbesondere auch, wenn eine graphische Datenverarbeitung, die mit diesem Controller ebenfalls möglich ist, in die Betrachtung einbezogen wird.

Die Schaltungen für den 6847 und für den 6845 wurden vom Verfasser ausgetestet. Sie haben sich bewährt. Sicher gibt es für Hard- und Software verbesserungsfähige Einzelheiten, die aber nicht den grundsätzlichen Aufbau berühren dürften. Der Verfasser hofft, daß er ein Spezialgebiet der Mikroelektronik auch den Lesern nahebringen konnte, die sich bisher noch nicht mit der besonderen Thematik befaßt haben.

## Literaturhinweise:

(1)	Don Lancaster	TV Typewriter Cookbook
(2)	Don Lancaster	The Cheap Video Cookbook
(3)	Bob Haas	Single Chip Video Controller, BYTE 5/79
(4)	N.N.	Elekterminal: Video-Interface, Elektor 12/78
(5)	Motorola	MC6845 CRT-Controller, AD1-465
(6)	Rockwell	R6545 CRT-Controller, 29000P55
(7)	AMI	S68047 Video Display Generator
(8)	AMI	68047 Video Display Generator Application Note

65<sub>xx</sub> MICRO MAG

(9) D. Amundson  
 (10) Intel  
 (11) Motorola  
 (12) Lesea/Zaks

Build a 850 TVT, KILOBAUD 2/78  
 Data Catalog 1977  
 MC6847 Video Display Generator, AD1-492  
 Microprocessor Interface Techniken

-----  
 DRIVER ROUTINES FOR  
 6845

```

==0000
; DESCRIPTION OF
VARIABLES
==0000 VIDRAM
      =$9000
; ADDRESS OF VIDEO-
RAM
==0000 VIDROM
      =$9E00
; ADDRESS OF DRIVER
ROUTINES
==0000 VIDADR
      =$9FFE
; ADDRESS OF CONTROL
LER ADDRESS REGISTER
==0000 VIDREG
      =$9FFF
; ADDRESS OF CONTROL
LER REGISTERS
==0000 LASLIN
      =VIDRAM+$300
; ADDRESS OF LAST
PAGE
==0000 LINLEN
      =$40
; LINE LENGTH
==0000 XSAV
      =$0170
; LOCATION TO SAVE
X-REG
==0000 CURPOS
      =$171
; ADDRESS OF CURSOR
POSITION
==0000 RSET
      =$E0BF
; ADDRESS OF MONITOR
RESET-ROUTINE
==0000 DILINK
      =$A406
; OUTPUT-ROUTINE
VECTOR
==0000 OUTDIS
      =$EF05
; ADDRESS OF DISPLAY
OUTPUT ROUTINE

```

-----  
 INITIALISATION-ROU-  
 TINE

```

==0000      *=VIDROM
==9E00 INIT
      ***
A00F LDY #50F
; SET COUNT FOR CONSE
CUTIVE SCROLLS
==9E02 I10
      ***
; SCOLL UNTILL SCREE
N IS CLEARED
8CFE9F STY VIDADR
; SET VIDEOADDRESS
REGISTER
B9309E LDA VIDTAB,Y
8DFF9F STA VIDREG
; TRANSFER PARAMETERS
INTO CONTROLLER
20809E JSR SCROLL
88 DEY
10F1 BPL I10
A940 LDA #<CHAR
; CHANGE OUTPUT-
ROUTINE ADDRESS
==9E13
8D06A4 STA DILINK
A99E LDA #>CHAR
8D07A4 STA DILINK+1
4CBFE0 JMP RSET
; GO TO RESET-
ROUTINE
==9E1E
      *=VIDROM+$30
==9E30 VIDTAB
      ***
58 .BYT $58
; MONITOR LINE LENGTH
40 .BYT $40
; DISPLAYED LINE
LENGTH
4A .BYT $4A
; SYNC POSITION
05 .BYT $05
; SYNC LENGTH

```

```

1C .BYT $1C
; MONITOR LINE NUMBER
13 .BYT $13
; VERTICAL ADJUST
10 .BYT $10
; VERTICAL DISPLAYED
LINES
17 .BYT $17
; VERTICAL SYNC PO-
SITION
00 .BYT $00
; NO INTERLACE
0C .BYT $0C
12 LINES PER CHA-
RACTER
00 .BYT $00
; CURSOR START LINE
07 .BYT $07
: CURSOR END LINE
00 .BYT $00
; START ADDRESS HIGH
00 .BYT $00
: START ADDRESS LOW
03 .BYT $03
; CURSOR ADDRESS HIGH
C0 .BYT $C0
; CURSOR ADDRESS LOW

```

-----  
 CHARACTER-OUTPUT-ROU-  
 TINE

```

==9E40
      *=VIDROM*$40
==9E40 CHAR
      ***
48 PHA
; SAVE A
8E7001 STX XSAV
; AND X
297F AND #57F
; MASK OFF HIGHEST
BIT
A20F LDX #50F
8E9E9F STX VIDADR
AEFF9F LDX VIDREG
; LOAD CURSOR PO-
SITION
C9 0D CMP #50D
; CHECK FOR CR

```

```

==9E50
F008 BEQ CR
293F AND #3F
; STRIP OFF BIT 6
9D0093 STA LASLIN,X
;STORE CHARACTER AT
CURSOR POSITION
IN LAST LI
E8 INX
;INCREASE CURSOR PO
SITION
D005 BNE NOSCR0
; NO END OF LINE, NO
SCROLL
==9E5A CR
***
; PROCESS CR, CLEAR
CURSOR POSITION
AND SCROLL
==9E5A SCRO
***
;SCROLL IF CR OR END
OF LINE
20809E JSR SCROLL
A2C0 LDX #3C0
;CURSOR START ADDRESS
==9E5F NOSCR0
***
A90F LDA #30F
8DFE9F STA VIDADR
8EFF9F STX VIDREG
;RESTORE CURSOR
POINTER
AE7001 LDX XSAV
; RELOAD SAVED X
68 PLA
: AND A
4C05EF JMP OUTDIS
; GO TO OUTPUT ROU-
TINE FOR DISPLAY
-----
SCROLL SCREEN AND
CLEAR LAST LINE
==9E6E
**=VIDROM+#80
==9E80 SCROLL
***
A200 LDX #300
==9E82 S10
***
;SHIFT UP FIRST
BLOCK OF VIDEORAM
BD4090 LDA VIDRAM+LI
NLEN,X
9D0090 STA VIDRAM,X
E8 INX
DOF7 BNE S10
==9E8B S20
***
;SHIFT UP SECOND
BLOCK OF VIDEORAM
BD4091 LDA VIDRAM+LI
NLEN+256,X
9D0091 STA VIDRAM+25
6,X
E8 INX
DOF7 BNE S20
==9E94 S30
***
;SHIFT UP 3. BLOCK
BD4092 LDA VIDRAM+51
2+LINLEN,X
9D0092 STA VIDRAM+51
2,X
E8 INX
DOF7 BNE S30
==9E9D S40
***
;SROLL LAST BLOCK
OF VIDRAM
BD4093 LDA VIDRAM+76
8+LINLEN,X
9D0093 STA VIDRAM+76
8,X
E8 INX
DOF7 BNE S40
AE3F9E LDX VIDTAB+$0
F
A920 LDA #320
==9EAB S90
***
;CLEAR LAST LINE OF
VIDRAM
9D0093 STA LASLIN,X
E8 INX
DOFA BNE S90
60 RTS
.END
ERRORS= 0000
-----

```

VIDEO-NEWS Die von der Firma Dohmann, Gütersloh, speziell für AIM entwickelte Videokarte gibt ihr Debut auf der Berliner Funkausstellung am Stande der Firma Körting: Betriebssystem im Festwertspeicher, beliebige Zeichensätze, voller Anschluß an die Prozessorbusse, 2 KB RAM Bildspeicher, hochwertiger HF-Generator für Anschluß an ein TV-Gerät. Mit Kontrollcodes kann von/auf Grafik umgestellt werden.

Eine TV-Karte ist auch von der Firma GWK in Herzogenrath (neue Anschrift!) angekündigt worden. Das VIDEO PLUS ist seit einiger Zeit beim Micro-Shop-Bodensee einschl. Betriebssystem und ausführlicher Dokumentation erhältlich. MC6845, Busanschluß, 128 Zeichen im character generator ROM. Bildspeicher 2-4 kB, 2 kB character generator RAM möglich (freie Zeichensätze), serieller Anschluß und Tastatur über VIA möglich. Der Steckplatz für eine 6502-CPU ermöglicht stand alone-Betrieb. - Zwei Videokarten liefert die Fa. Eltec Elektronik GmbH, Neubrunnenstr. 10, 65 Mainz. VIC 1 hat 7-bit-breiten Eingang und stellt 64 ASCII-Zeichen dar. Grafik 1 ermöglicht hochauflösende Grafik und hat 256x256 einzeln adressierbare Bildpunkte. Mit Zusatzgeräten sind Graustufungen oder Farben möglich. - 'A CRT Monitor or TV Interface for AIM 65' heißt eine 15-seitige application note von Rockwell (Juni 1979): Kompletter Schaltungsvorschlag und Betriebssoftware sind enthalten, die CPU greift bei Ø2 und der Controller bei Ø1 auf das Video-RAM zu.

## OSZILLOGRAPH ALS BILDSCHIRM FÜR DEN AIM

Dr. Fritz Mayer-Lindenberg, Nadefeld 4, D-4800 Bielefeld 16

E: A simple interface and the program below will output from the editor of the AIM 65 to the screen of an oscilloscope.

Das hier beschriebene Programm/Interface gestattet die Darstellung von 16 Zeilen un je 32 Zeichen auf einem Oszillographen bei einer Wiederholffrequenz von ca. 20 Hz. Bild 1 zeigt das Interface, welches an den VIA 6522 angeschlossen wird. Es wird angenommen, daß der Oszillograph einen TTL-kompatiblen Eingang zur Z-Modulation besitzt, wie etwa die HAMBG-Typen HM412 und HM512 (sonst ist eine Verstärkungsstufe erforderlich).

Das Y-Signal ist invertiert darzustellen. PBO triggert die Zeitablenkung des Oszillographen. Die an PB1 bis PB4 angeschlossenen Widerstände bilden einen einfachen D/A-Wandler zur Erzeugung der 16 Zeilen. Zur Zeichenerzeugung wird der Zeichengenerator im Monitor-ROM verwendet. Die Zeichenbytes werden in das Schieberegister des VIA geladen, dessen Ausgang CB2 den Kathodenstrahl helltastet. Bei jedem Laden des Schieberegisters wird durch den CSA-Impuls über T der Kondensator C geladen, wodurch am Kollektor von T eine Sägezahnspannung entsteht (siehe dazu Bild 2).

Das Programm ist ab Stelle 0000 geschrieben, es kann aber an jeder Stelle des Speichers niedergelegt werden, da es nur relative Sprünge enthält. Das Programm besteht aus einer vom Timer T2 periodisch ausgelösten Interrupt-routine, die jeweils eine Zeile auf den Bildschirm schreibt. Dadurch kann während der Bilddarstellung der AIM-Monitor bedient werden. Für die Zusammenarbeit mit dem Editor des AIM enthält das Programm einen Abschnitt (0069, 0075-OA0E), in dem jeweils die 16. Zeilenadresse mit der aktiven Zeilenadresse des Editors verglichen wird. Bei Ungleichheit wird die Bildanfangesadresse korrigiert, so daß die aktive Zeile den gezeigten Ausschnitt den gezeigten Ausschnitt des Textspeichers bestimmt (automatisches scrolling). Der Textspeicher muß dazu auf einem Seitenanfang beginnen. Bei anderen Anwendungen des Bildschirms (z.B. zum Display von 128 Speicherstellen oder von 16 disassemblierten Befehlen) kann dieser Programmteil natürlich fortgelassen werden.

Zur Initialisierung müssen folgende Register in der angegebenen Reihenfolge geladen werden:

```
A400 00      IRQ-ADRESSE
A401 00      "
A002 1F
A000 FD
A00B 18
A009 FF
.....
BILDSCHIRM EIN:
A00E A0
BILDSCHIRM AUS:
A00E 20
```

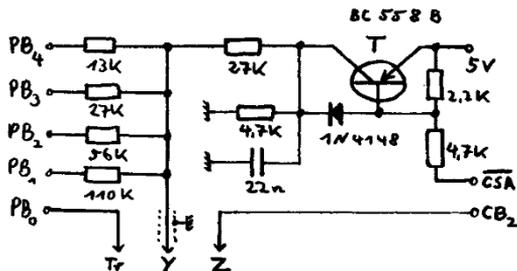


Bild 1: Das Hardware-Interface

## Kommentare zum Programm:

OODA und OODB enthalten die Bildanfangsadresse  
 OODC und OODD enthalten die jeweilige Zeilenanfangsadresse  
 OO05 Y indiziert die Zeichen einer Zeile  
 OO0A der Wert OE bestimmt die Zeilenfrequenz  
 OO0F die Zeitablenkung wird getriggert  
 OO14 ist (Zeichen), Y das Stop-Zeichen?  
 OO1D OO28, OO32, OO3D, OO49: Hier werden die 5 Kolonnen eines Zeichens ins Schieberegister geladen.  
 OO1C durch die ASL's ist das jeweils letzte vom Schieberegister ausgegebene Bit eine 0 (Dunkeltastung)  
 OO21 die NOP's sind für das Timing wichtig  
 OO42 keine weiteren Zeichen abbilden, wenn Zeichen Y+1 ein CR ist.  
 OO4F der Wert ist die Zeilenlänge  
 OO53-OO5B: letzten Zeichenindex zur Zeilenanfangsadresse addieren  
 OO65 vergleiche Zeilennummer mit 16, gehe nach RTI, falls kleiner  
 OO6B falls größer: setze Zeilenanfangsadresse auf Bildanfangsadresse  
 OO75 16. Zeile. Vergleiche Zeilenanfangsadresse mit Adresse der aktiven (offenen) Zeile in OODF/EO  
 OO7F ist gleich, RTI  
 OO81-OOAE Korrektur der Bildadresse; zähle von aktiver Zeile um 15 Zeilen zurück, aber höchstens bis zum Textanfang.

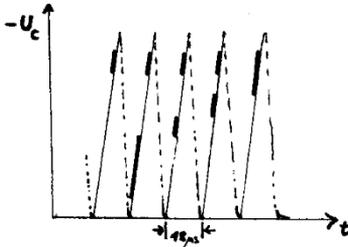


Bild 2: Zur Zeichenerzeugung

```

0000 48 PHA
0001 8A TXA
0002 48 PHA
0003 98 TYA
0004 48 PHA
0005 A0 LDY #00
0007 8C STY A008
000A A9 LDA #0E
000C 8D STA A009
000F EE INC A000
0012 B1 LDA (DC),Y
0014 F0 BEQ OO53
0016 29 AND #3F
0018 AA TAX
0019 BD LDA F2E1,X
001C 0A ASL .A
001D 8D STA A00A
0020 C8 INY
0021 EA NOP
0022 EA NOP

```

Der Drucker darf während des Bildschirmbetriebes nicht verwendet werden, da der periodische Interrupt sein Timing stört (Überhitzung). Sollen gleichzeitig andere Programme auf dem AIM laufen, so dürfen diese nicht mit dem BRK-Kommando abgeschlossen werden, weil danach der Monitor gegen Interrupt maskiert ist. Dagegen ist Laden von und auf Tonbandcassette möglich. Viel Spaß!

```

0023 EA NOP
0024 BD LDA F321,X
0027 0A ASL .A
0028 8D STA A00A
002B B1 LDA (DC),Y
002D 48 PHA
002E BD LDA F361,X
0031 0A ASL .A
0032 8D STA A00A
0035 68 PLA
0036 C9 CMP #0D
0038 08 PHP
0039 BD LDA F3A1,X
003C 0A ASL .A
003D 8D STA A00A
0040 28 PLP
0041 08 PHP
0042 D0 BNE OO45
0045 C8 INY
0048 BD LDA F3E1,X

```

65<sub>xx</sub> MICRO MAG

0049	8D STA A00A	0083	A9 LDA #00
004C	28 PLP	0085	85 STA DA
004D	F0 BEQ 0053	0087	A5 LDA E0
004F	C0 CPY #20	0089	85 STA DB
0051	30 BMI 0012	008B	A9 LDA #0D
0053	98 TYA	008D	A4 LDY DF
0054	18 CLC	008F	D0 BNE 009B
0055	65 ADC DC	0091	A4 LDY DB
0057	85 STA DC	0093	C4 CPY E4
0050	90 BCC 005D	0095	F0 BEQ 00B0
005B	E6 INC DD	0097	A0 LDY #FF
005D	EE INC A000	0099	C6 DEC DB
0060	AD LDA A000	009B	D1 CMP (DA),Y
0063	29 AND #1F	009D	D0 BNE 00A2
0065	C9 CMP #10	009F	CA DEX
0067	90 BCC 00B0	00A0	F0 BEQ 00A9
0069	F0 BEQ 0075	00A2	88 DEY
006B	A5 LDA DA	00A3	C0 CPY #FF
006D	85 STA DC	00A5	D0 BNE 009B
006F	A5 LDA DB	00A7	F0 BEQ 0091
0071	85 STA DD	00A9	C8 INY
0073	B0 BCS 00B0	00AA	84 STY DA
0075	A5 LDA DC	00AC	D0 BNE 00B0
0077	C5 CMP DF	00AE	E6 INC DB
0079	D0 BNE 0081	00B0	68 PLA
007B	A5 LDA DD	00B1	A8 TAY
007D	C5 CMP E0	00B2	68 PLA
007F	F0 BEQ 00B0	00B3	AA TAX
0081	A2 LDX #10	00BA	68 PLA
		00B5	40 RTI

★ ★

EINLADUNG ZUM ERSTEN OFFENEN  
65XX COMPUTERMEETING  
13.-14 OKT.79, SA/SO 14-11 h  
Hotel Sonnenberg, Weillrodt-Riedel-  
bach, Camberger Weg 6, 30 km NW  
Frankfurt, BAB-Abf. Camberg. Tel.:  
06083-400. Hotelbuchung sofort  
vornehmen, begrenzte Kapazität!  
Organisationsbeitrag DM 8,-/Teiln.  
*Eine große Zahl hervorragender  
Fachleute beschäftigt sich unab-  
hängig und weitgehend ohne Wissen  
voneinander mit der Hard- und  
Softwareentwicklung für 65xx. Der  
Herausgeber lädt sie zu o.a. Profi-  
meeting ein, zum Kennenlernen, zum  
Gedankenaustausch. Rahmenprogramm  
mit Überblicken. Weitere Beiträge  
sehr willkommen. Programme, Demos,  
Geräte zugelassen. Arbeitsgesprä-  
che, keine Einführung für Anfänger.  
Verschiedene Fachleute haben ihre  
Teilnahme bereits angekündigt. Ge-  
ben Sie bitte bei Teilnahme Meldung  
nach hier. Fahrgemeinschaften ggfs.  
zu vermitteln.* R.L.

ERZEUGUNG QUASISTATISTISCHEN RAUSCHENS  
DURCH PSEUDO-ZUFALLSZAHLNFOLGEN (2)

Zu diesem Artikel in Heft 7 macht Herr Dr. Mayer-Lindenberg, Bielefeld, folgende Anmerkungen:

Während die Rückführung von der letzten und viertletzten Stufe bei Schieberegistern der Länge 7, 10, 17, 20, 25, 28, 31 Zufallsfolgen maximaler Länge ergibt, ist dies bei Registern z.B. der Längen 8, 11, 16, 48 nicht der Fall. Es gibt für Register der Länge 16 überhaupt keine zweifache Rückführung mit dieser Eigenschaft. Das Schieberegister der Länge 48 zerfällt bei der angegebenen Rückführung in 3 gleichartige der Länge 16 mit Periode 255, die nacheinander bedient werden. Die Periode ist daher nur 765!

Das nachfolgende Programm verwendet ein 24-Bit-Schieberegister in den Speicherstellen 00, 01, 02 mit einer andersartigen Rückführung. Die Periode ist  $2^{24} - 1$ , was für viele Anwendungen genügen wird.

0003	A5 LDA 00	000A	90 BCC 000E
0005	0A ASL .A	000C	49 EOR #87
0006	26 ROL 01	000E	85 STA 00
0008	26 ROL 02	0010	60 RTS

Der Herausgeber hat diesen elegant-kurzen Algorithmus für Zufallszahlen erfolgreich auf den PET-Bildschirm arbeiten lassen. Man sieht dabei sehr schön den stochastischen Ablauf.

## AUSKUNFTSSYSTEM MIT AIM 65

E: In two main bodies of the program data storage and data retrieval are incorporated to build up an information system. The user may create mass storage files with addresses and other information on tape and may scan the whole blocked file for any item (string) when retrieving data automatically. The AIM Monitor/Editor is modified for this aim.

Wie alle gut dokumentierten Programme, so läßt sich auch das des Editors in Abwandlungen benutzen. Das Auskunftssystem verwendet wesentliche Bestandteile der Editor-Commands LIST (Ausgabe), READ (Einlesen) und FIND (Suchen einer Zeichenkette). Während man den AIM auf der Ausgabeseite über das DILINK beeinflussen kann, so ist das auf der Eingabeseite mangels eines veränderlichen Vektors nicht möglich. Es war daher eine externe Überlagerung des Editors zu schaffen, die z.T. Bestandteile der erwähnten Commando-Routinen enthält.. Zum Zwecke der Automatisierung der Abläufe war dabei insbesondere zu vermeiden, daß man nach Abarbeitung von Programmteilen in die ENDERR-Routine des Editors gelangt, die jede Beeinflussung von außen ausschließt und die nur auf die Betätigung irgendeiner Kommandotaste des Editors in Schleife wartet.

Das Auskunftssystem eignet zur Anlage jedweder ASCII-codierter DATAFILES auf Tonbandcassette. Zu denken ist an Adreßdateien, Literaturverzeichnisse, Inhaltsverzeichnisse mit Verweisungen für beliebige Sachgebiete, Termindateien mit Hinweisen, Telefonverzeichnisse etc.. Das mitgeteilte Programm gliedert sich in zwei Abschnitte: Datenerfassung und Suchroutine. Beide sind automatisiert.

Programmablauf und Datenerfassung: Das Programm benötigt seinen eigenen Speicherplatz, der mit \*0A00 vom Assembler zugewiesen wurde. START also an dieser Adresse. Man landet direkt in der Eingabe eines frisch in Adressen 0C00 bis 0EFF initialisierten Editors (Parametervorgabe in Tabelle TAD). Man gibt wie im Editor üblich ein. Nachdem man bis zu 768 Zeichen für einen Block gefüllt hat, schließt man den Editor mit zweimaligem Betätigen der RETURN-Taste ab ('END') und verläßt den Editor mit 'Q' oder 'ESC'. Das Startprogramm hat bereits die Benutzertasten F1, F2 und F3 initialisiert, und zwar durch die durch Tabelle TVEC vermittelten Parameter. Es hat auch bereits die Bezeichnung des DATAFILES und der Magnetbandeinheit abgefragt.

Ein Betätigen der Taste F2 führt jetzt zur Aufzeichnung der erfaßten Daten auf Magnetband, das nach den erforderlichen Bandsätzen wieder anhält (Fernsteuerung erforderlich). Es erfolgt Wiedereintritt in einen frisch initialisierten (leeren) Editor, in dem man die nächste Informationseinheit erfaßt. Das Wechselspiel der Funktionstasten ist nicht anders als an einem Datenerfassungsplatz. - Ein FILE muß für den Computer eine erkennbare Begrenzung enthalten, damit man beim Suchlauf nicht über das END OF FILE ins Leere stößt und vergeblich weitersucht. In unserem Auskunftssystem wird daher das END OF FILE durch einen Datensatz gekennzeichnet, der als erstes und einziges Zeichen den Klammeraffen enthält, Schreibweise '@'. Damit ist die Datenerfassung abgeschlossen.

Inbetriebnahme des Auskunftssystems: Es sind die Leistungen des FIND-Befehles im Editor implementiert. Man kann also nach einer beliebigen Zeichenkette suchen. Diese wird nach Betätigen der Anwendertaste F3 nach dem angezeigten PROMPT eingegeben. Danach deklariert man FILENAME und Bandeinheit. Das Programm sucht nun die Datenblöcke im Start-Stop-Betrieb nach

65<sub>xx</sub> MICRO MAG

der Zeichenkette ab. Ist sie nicht im DATAFILE enthalten, so erscheint am Ende des FILES die Anzeige 'END@'. Wenn sie jedoch gefunden wurde, so wird sie mit ggfs. ergänzender Information auf dem Display angezeigt. In diesem Falle ist der gesamte Datenblock im Editor resident. Mit Befehlen wie UP, DOWN, TOP, BOTTOM, FIND etc. kann man ggfs. weitere ergänzende Information zur Anzeige bringen. Das Auskunftssystem beinhaltet also für das Beispiel einer Adressendatei folgende Möglichkeiten: Man kann nach Anrede, Vornamen, Familiennamen, PLZ, Ort etc. absuchen und findet im zutreffenden Falle auch die ergänzende Information in den Zeilen oberhalb oder unterhalb der gefundenen Zeile.

Dieses Auskunftssystem kann nur ein Anfang sein. Erstens ist an Beschleunigung zu denken, und zwar durch Anschluß an den HAMMING-WAY in Heft 6 oder an Diskettensysteme. Zweitens: Das Suchwort wird bis maximal 60 Bytes in Speicherzellen ab 0000 niedergelegt. Man könnte aber auch zwei alternative Suchworte vorgeben, die nicht einen zusammenhängenden String bedeuten. Man könnte dann logisch ODER suchen. Dafür wären die Programmabschnitte zur Stringvorgabe und zur Stringprüfung entsprechend aneinanderzureihen. Eine weitere Ergänzung würde darin bestehen, daß das Programm nicht an der ersten identischen Stringkette aussteigt, sondern noch alle folgenden identischen Strings erfaßt und ausdrückt. Dazu müßte man auf dem externen Kommando-Level verbleiben. Und schließlich ist daran zu denken, daß man eine externe Statistik über die Suchhäufigkeit aufstellt und ein darauf abgestelltes Sortierprogramm aufbaut, das die Zugriffe in zeitlicher Hinsicht optimiert. Die letzte Überlegung bezieht sich auf kommerziell einzusetzende Verfahren.

0A00	D8	CLD		0A51	20	JSR	F709		UPNO
01	20	JSR	E8A2	FNAM	54	20	JSR	F8E9	
04	A2	LDX	#0C	TRANSPOR-	57	90	BCC	0A46	
06	BD	LDA	0A23,X	TIERE VEKTO-	59	20	JSR	F73F	
09	9D	STA	0108,X	RENTABELLE	5C	20	JSR	FF0D	
0C	CA	DEX			5F	68	PLA		ADJUST
0D	10	BPL	0A06		60	68	PLA		SP
0F	20	JSR	0A6A	INIT EDITOR	61	68	PLA		
12	A9	LDA	#0D	KEYBOARD	62	A9	LDA	#04	STOP TAPE
14	8D	STA	A412	INFLAG	64	8D	STA	A800	TO PORT
17	4C	JMP	F7D5	EDITOR READ	67	4C	JMP	0A0F	NACHSTE EINGABE
<M>	=0A1A	XX	00 0C 00	VEKTOREN FUER	6A	A2	LDX	#07	EDITOR VEK-
< >	0A1E	0C	00 0C FF	EDITOR-ANLAGE	6C	BD	LDA	0A1B,X	TOREN INI-
< >	0A22	0E			6F	95	STA	DF,X	TIALISIEREN
<M>	=0A23	C7	0F 30 0A	USER INPUT	71	CA	DEX		
< >	0A27	4C	00 0A 4C	OUTPUT VEKTOREN,	72	10	BPL	0A6C	
< >	0A2B	30	0A 4C 7A	F1, F2, F3	74	A0	LDY	#00	LEERER
< >	0A2F	0A		TASTENBELEGUNG	76	98	TYA		EDITOR MIT
0A30	A9	LDA	#2C	DUMP MIT TASTE F2	77	91	STA	(DF),Y	00 ALS 1.
0A32	8D	STA	A419	COUNT	79	60	RTS		ZEICHEN
0A35	20	JSR	EA13	CRLW	-----				
38	20	JSR	F8BC	TOPNO, 1. ZEILE	0A7A	A0	LDY	#00	EINGABE NACH*F3*
3B	A9	LDA	#54	*T*, TAPE	7C	20	JSR	E7BD	TASTE.
3D	8D	STA	A413	OUTFLAG	7F	20	JSR	E95F	ABLAUF WIE IM
40	20	JSR	E56F	DUMPTA	82	C9	CMP	#0D	EDITOR BEI FCH
43	4C	JMP	0A4E		84	D0	BNE	0A90	
46	20	JSR	E907	RCHEK, WIE IN LST	86	C0	CPY	#00	ANLAGE DES
49	20	JSR	E790		88	D0	BNE	0A90	SUCHSTRINGS
4C	FO	BEQ	0A59		8A	20	JSR	F709	
4E	20	JSR	F727	PLNE, SHOW	8D	4C	JMP	0AA8	

0A90	99 STA 0000,Y	BIS ZU 60	0B01	EE INC A415	
93	C9 CMP #0D	ZEICHEN AB	04	4C JMP 0AED	
95	F0 BEQ 0A9F	ADRESSE 0000	07	C8 INY	
97	C8 INY		08	E8 INX	
98	C0 CPY #60		09	EC CPX A429	STIY+2
9A	D0 BNE 0A7F		0C	D0 BNE 0AF2	N. ZEICHEN
9C	4C JMP FA72	ERROR	0E	60 RTS	
9F	20 JSR EA24		0F	20 JSR F709	UPNO
A2	8C STY A429	COUNT	12	4C JMP 0AE8	
A5	20 JSR E8A2	FNAM	15	20 JSR F8B6	WIE INL
A8	20 JSR 0A6A	INIT EDITOR	18	A0 LDY #00	ABER GEKÜRZT
AB	A0 LDY #54	T=TAPE	1A	84 STY E9	
AD	8C STY A412	INFLG	1C	20 JSR E78D	
B0	20 JSR E32F	LOADTA	1F	20 JSR EB44	
B3	20 JSR 0B15	ENTSPR. INL	22	20 JSR E993	
B5	20 JSR 0ABC	ENTSPR. UPNO	25	20 JSR FEF8	
B9	4C JMP 0AB3	WIE INPU1	28	C9 CMP #0A	
BC	A0 LDY #00	NEUES 'UPNO'	2A	F0 BEQ 0B22	
BE	20 JSR F8E9		2C	C9 CMP #0D	
C1	90 BCC 0AC9		2E	F0 BEQ 0B48	
C3	A2 LDX #FF	ADJUST SP	30	C0 CPY #3C	
C5	9A TXS		32	B0 BCS 0B3C	
C6	4C JMP 0ACC	NUN VERGLEICH	34	99 STA A438,Y	
C9	4C JMP F713	REST VON UPNO	37	C8 INY	
CC	A9 LDA #04	STOP TAPE	38	C0 CPY #3C	
CE	8D STA A800	PORT	3A	D0 BNE 0B22	
D1	20 JSR F8BC	TOPNO, 1. ZEILE	3C	A0 LDY #3C	
D4	A0 LDY #00	1. ZEICHEN EIN	3E	A9 LDA #01	
D6	B1 LDA (DF),Y	@ ?	40	0D ORA A411	
D8	C9 CMP #40		43	8D STA A411	
DA	D0 BNE 0ADF	NEIN	46	8C STY A415	
DC	4C JMP FA5C	DISPLAY 'END@'	49	D0 BNE 0B22	
DF	20 JSR 0AE8	VERGLEICHE	4B	84 STY EA	
E2	20 JSR F80F	DISPLAY GEFUNDENE	4D	C0 CPY #00	
E5	4C JMP FA78	ZEILE, GOTO EDITOR	4F	D0 BNE 0B66	
E8	A0 LDY #00	VERGLEICH ENTSP.	51	AD LDA A419	
EA	8C STY A415	FC5	54	D0 BNE 0B66	
ED	AC LDY A415		56	20 JSR EA24	
F0	A2 LDX #00		59	20 JSR FF03	
F2	B1 LDA (DF),Y	VERGL. AUF ENDE	5C	20 JSR F727	
F4	D0 BNE 0AF9	DES EDITORS	5F	20 JSR 0ABC	
F6	4C JMP 0AA8	ENDE, N. BLOCK	62	20 JSR F608	
F9	C9 CMP #0D	ZEILENENDE?	65	00 BRK	
FB	F0 BEQ 0B0F	JA	66	20 JSR F93F	
FD	D5 CMP 00,X	VERGL. STRINGS	69	4C JMP EA24	
FF	F0 BEQ 0B07	IDENT., WEITER			

R.L.

## KLEINANZEIGEN DER LESER

Drei AIM 65-Programme: Baudot-Ausgabe über TTY (Postfernschreiber), Schreibautomat mit dito, Morsecode aus Texteditor, zus. für DM 38,- Vorkasse als Cassette oder Listing. Hans-Joachim Regge, Fesenfeld 57, 28 Bremen 1. Postscheckkonto Berlin 205337-100. BLZ 100 100 10.

## AIM-TASTATUR MIT KLEINSCHREIBUNG!

E: Upper and lower case characters from the AIM keyboard.

Unter AIM 65 Spezial (2) in Heft 7, S. 35 zeigten wir den Weg auf, wie man die AIM-Tastatur über das DILINK aktiv hält, auch wenn man mit einem Video-Terminal oder mit TTY als Ausgabe zusammenarbeitet. Einige Leser fanden etwas mühsam zur richtigen Implementierung. Daher folgende Hinweise: Die Initialisierung verändert den Adreßvektor im DILINK A406/07. Sie kann nicht per Tastatur vorgenommen werden, weil der Monitor sonst aufhängt. Nach dem Beispiel in Heft 6 setzt man die Sternadresse (\*) auf die Zeile 09B7 und initialisiert mit der Taste G=G0. Das DILINK weist jetzt auf die Adresse 09A0. Bei der Initialisierung gem. Beispiel wird die Baudrate auf 300 eingesetzt. Für Interfaces mit anderen Übertragungsgeschwindigkeiten ist die Baudrate gem. Anwenderhandbuch, Seite 9-31 einzurichten.

Vorstehende Manipulationen betrafen die Ausgabeseite, für die die Verfasser des Monitorprogrammes vorsorglich den veränderlichen Vektor im DILINK vorgesehen haben. Für die Eingabeseite besteht leider keine entsprechende Eingriffsmöglichkeit. Dabei hätte man ohne Mehraufwand auch Kleinschreibung über das AIM-Keyboard vorsehen können. Man hat sie einfach 'verboten', wohl wegen der beschränkten Darstellungsmöglichkeiten von Display und Drucker. Bei Interesse studiere man das nachfolgende Programm hierzu genauer.

Dem Herausgeber ist es mit einer User Input-Routine gleichwohl gelungen, Kleinschreibung auf dem AIM-Keyboard zu implementieren. Die Utility arbeitet mit EDITOR und auch mit BASIC zusammen, wenn man will, sogar auch mit dem Assembler. Für den Editor konnten die Stringfunktionen F und C (FIND und CHANGE) allerdings noch nicht auf Kleinschreibung ausgedehnt werden. Es folgen aber Anregungen. Damit sind nunmehr Eingabe- und Ausgabeseite des AIM beeinflussbar geworden, und es eröffnen sich breite Perspektiven für die Anwendung. Vor allem scheint auch der Kostengesichtspunkt wichtig. Man benötigt keinerlei Terminaltastatur mehr, für die ja immerhin auch ab etwa DM 200,- verlangt werden. Das erarbeitete Konzept spricht umso mehr für den 'Stand der Technik' mit Anschluß eines Video-Interfaces an die Prozessorbusse (also kein TTY-Betrieb), dargestellt besonders auch im Artikel 'Der Video-AIM' und realisiert im VIDEO-PLUS des Micro-Shop-Bodensee (Nedela) und in der Video-Karte der Firma Dohmann in Gütersloh mit ihrem beliebigen Zeichensatz.

Das nachfolgende Programm entspricht dem Abschnitt GETK00 bis GETK10 im Monitor-Listing, natürlich mit Veränderungen. Man vergesse nicht, den USER INPUT VECTOR in 0108/09 auf die Anfangsadresse 0FC7 zu richten, in der Reihenfolge Low/High. Das Programm wurde am Ende des 4k-Speichers angesiedelt, die Manipulierung des DILINK und der Baudrate konnte gleich angeschlossen werden. Hier das Listing ohne Kommentare (s. Monitor-Listing).

0F00 A9 LDA #00	0F19 A2 LDX #03	0F2A CA DEX
0F02 8D STA A477	0F1B A9 LDA #7F	0F2B D0 BNE 0F1D
0F05 20 JSR F050	0F1D 38 SEC	0F2D F0 BEQ 0F0B
0F08 20 JSR ECEF	0F1E 6A ROR .A	0F2F 68 PLA
0F0B 20 JSR ED2A	0F1F 48 PHA	0F30 AD LDA A42B
0F0E A9 LDA #8F	0F20 20 JSR ED0B	0F33 49 EOR #FF
0F10 8D STA A480	0F23 AD LDA A482	0F35 AA TAX
0F13 AD LDA A482	0F26 4A LSR .A	0F36 EE INC A42A
0F16 4A LSR .A	0F27 90 BCC 0F2F	0F39 20 JSR ED05
0F17 B0 BCS 0F39	0F29 68 PLA	0F3C 88 DEY

OF3D	D0 BNE	OF48	OF76	F0 BEQ	OF86	OFB7	68 PLA
OF3F	AD LDA	A42B	OF78	C9 CMP	#0C	OFB8	E0 CPX #20
OF42	C9 CMP	#F7	OF7A	80 BCS	OF81	OFBA	A0 BPL OF87
OF44	80 BCS	OF4A	OF7C	68 PLA		OFBC	C9 CMP #20
OF46	90 BCC	OF0B	OF7D	29 AND	#EF	OFBE	90 BCC OF87
OF48	30 BMI	OF0B	OF7F	D0 BNE	OF87	OFCC	09 ORA #20
OF4A	20 JSR	ED2C	OF81	68 PLA		OFCD	D0 BNE OF87
OF4D	98 TYA		OF82	09 ORA	#10	OFCE	80 BCS OFC7
OF4E	0A ALS	.A	OF84	D0 BNE	OF87	OFD0	60 RTS
OF4F	0A ASL	.A	OF86	68 PLA		OFD3	20 JSR EB9E
OF50	0A ASL	.A	OF87	C9 CMP	#60	OFD4	20 JSR OF08
OF51	A8 TAY		OF89	D0 BNE	OF91	OFD5	20 JSR EBAC
OF52	AD LDA	A42B	OF8B	E0 CPX	#00	OFD6	20 JSR E97A
OF55	4A LSR	.A	OF8D	F0 BEQ	OFB4	OFD7	60 RTS
OF56	90 BCC	OF5B	OF8F	29 AND	#4F	OFD8	8D STA 0170
OF58	C8 INY		OF91	C9 CMP	#1C	OFD9	29 AND #7F
OF59	D0 BNE	OF55	OF93	D0 BNE	OFA9	OFDB	F0 BEQ OFE0
OF5B	B9 LDA	F421,Y	OF95	20 JSR	E6E1	OFDD	4C JMP EEA8
OF5E	48 PHA		OF98	A0 LDY	#01	OFDE	A9 LDA #0A
OF5F	8A TXA		OF9A	B9 LDA	A415,Y	OFDF	20 JSR EEA8
OF60	F0 BEQ	OFB7	OF9D	38 SEC		OFE0	A9 LDA #0A
OF62	29 AND	#10	OF9E	E9 SBC	#03	OFE2	AD JSR EEA8
OF64	F0 BEQ	OF6C	OFA0	99 STA	A415,Y	OFE5	D0 LDA 0170
OF66	68 PLA		OFA3	88 DEY		OFEB	4C JMP EEA8
OF67	29 AND	#3F	OFA4	10 BPL	OF9A	OFED	A9 LDA #D4
OF69	4C JMP	OF87	OFA6	4C JMP	OF08	OFEE	8D STA A406
OF6C	68 PLA		OFA9	C9 CMP	#5C	OFF0	A9 LDA #0F
OF6D	48 PHA		OFAB	D0 BNE	OFB3	OFF2	8D STA A407
OF6E	29 AND	#40	OFAD	20 JSR	F04A	OFF5	A9 LDA #0C
OF70	D0 BNE	OFB7	OFB0	4C JMP	OF08	OFF7	8D STA A417
OF72	68 PLA		OFB3	60 RTS		OFFA	A9 LDA #C2
OF73	48 PHA		OFB4	4C JMP	OF00	OFFC	8D STA A418
OF74	29 AND	#0F				OFFF	00 BRK

<M>=0108 C7 OF USER INPUT VECTOR

Die Initialisierung des DILINK-Vektors erfolgt in obigem Programm in Stelle OFEB. Der Vektor wird auf OFD4 gerichtet (Zwischenroutine). Die Baudrate ist auf 300 gesetzt.

Hinsichtlich der Editor-Funktionen F und C wird man darauf achten müssen (für Kleinschreibung), daß man nicht auf ENDERR stößt, sondern in eine anwenderdefinierte Routine zurückkehrt. Hierzu liegen zwar schon Überlegungen, auch von Herrn Dipl.-Ing. Wöflingseder vor, sie konnten aber noch nicht erprobt werden.

Nun zur Anwendung: Der Editor wird wie üblich und gewünscht eröffnet. Bei der Frage IN= tippt man 'U', schon hat man wahlfreie Kleinschreibung. Das Gleiche gilt für das Hinzuladen von Zeilen im Editor mit R=READ. Man tippt IN=U. Eine READ-Anweisung wird mit CTRL/Z (Drücken zweier Tasten zum Erzeugen eines Kontrollzeichens) abgeschlossen. Dabei wird hexadezimal '0A' erzeugt. Die gleiche Prozedur ist auf BASIC anzuwenden. Man schreibt: LOAD IN=U und schließt die Eingabe ebenso mit CTRL/Z ab. In BASIC ist Kleinschreibung vor allem für PRINT-Statements interessant, gleichwohl dekodiert der Interpreter auch etliche kleingeschriebene Kommandos in den Statements. Auch das Assemblerprogramm (ROM) 'frißt' IN=U, man ist allerdings gezwungen, gleichlautend und sehr akkurat den Quelltext zweimal hintereinander für die PASSES einzugeben. Sicher keine Empfehlung, aber eine Möglichkeit. Verschiedene Kontrollzeichen für Video-Interfaces sind im Programm eingeschlossen.

## 65xx MICRO MAG

### AIM SPEZIAL (3)

Beim Abschluß dieses Heftes wurde die Fertigstellung des deutschen Anwenderhandbuches für den AIM täglich erwartet. Ein verzögerndes Problem stellte die Wiedergabe der Listings auf Thermalstreifen durch den Drucker dar.

Nach verschiedenen Beobachtungen, auch durch den Herausgeber, mißt der AIM die Baudrate beim Verkehr mit einem als TTY betriebenen Terminal nicht richtig, man muß in gewissen Fällen die Baudrate manuell in die Speicherstellen A417/18 lt. Anweisung im Handbuch, S. 9-26, eintragen. Der Fehler liegt offensichtlich in der Behandlung des Übertrages im Programmteil PATCH1.

Für das AIM-BASIC liegt ein ausführliches memory map noch nicht vor. Nicht alle Leser haben die nachgelieferte BASIC Reference Card erhalten, der wir folgende Parameter für die Zeropage entnehmen:

USR Routine address	hex	04,05	dezimal	04,05
Line width, Zeilenbreite	12		18	
Eingabepuffer	14-50		20-88	
Pointer f. Programmstart, L/H	73,74		115,116	
Start der Variablen, L/H	75,76		117,118	
Pointer auf Anfang von arrays L/H	77,78		119,120	
Pointer to top of used mem L/H	79,7A		121,122	
Pointer für Freiraum nach oben L/H	7B,7C		123,124	
Pointer f. BASIC Speicherbegrenzung	7F,80		127,128	
Fließkomma-Akku	A9-AE		169-174	
Fließkomma Argument-Register	B1-B6		177-182	

Im einzelnen ist mitzuteilen, daß die BASIC Zeilen-Nr. in den Zellen (hex) 14/15 abgelegt wird. Darauf folgt ein künstlicher Operationscode für die BASIC-Anweisung sowie in ASCII der Text der Anweisung. Abschluß mit 00 xx 00. Die Pointer für die Variablen werden erst mit RUN belegt. Beim Schreiben oder Laden eines Programmes ist der Pointer für Variable identisch mit dem Text-Ende der BASIC-Statements. Ab Speicherstelle BF ist eine Abfrage-routine in die Zeropage projiziert. Beim Initialisieren des BASIC werden einige Parameter in die Speicherstellen ab 0200 projiziert, deren Bedeutung nicht ersichtlich ist.

Zum Aufbau der BASIC-Statements im Speicher ist zu erklären: Speicherstelle 0211 (hex) wird als Anfang mit 00 initialisiert. Danach folgt das 'LINK', das Bindeglied zur nächsten Anweisung. Wegen der unterschiedlichen Länge der vom Benutzer eingegebenen Anweisungen muß der Interpreter wissen, an welcher Speicherstelle das LINK der nächsten Anweisung steht. Das LINK enthält also in der Reihenfolge LOW/HIGH die Adresse des nächsten LINK - und damit den Beginn der nächsten Anweisung. Trifft der Interpreter auf ein LINK, das in beiden Bytes 00 00 enthält, so ist es für ihn das untrügliche Zeichen, das hier das Ende des Quellentextes vorliegt. M.a.W. Bei der zeilenweisen Eingabe eines BASIC-Programmes schiebt das Editierungsprogramm die 00-00-Kombination jeweils um ein Statement nach hinten.

Auf das LINK folgt in hexadezimaler Zählung die Zeilennummer in der Reihenfolge LOW/HIGH. Für Zeile 10 z.B. also 0A 00. Das 5. Byte stellt den vom Interpreter auszudeutenden Operationscode dar oder bei Variablen ihren Namen. Darauf folgt in ASCII-Codierung der Rest der Anweisung. Diese wird jeweils durch ein Byte 00 abgeschlossen. In der Erklärung dieses grundsätzlichen Aufbaues liegt der Schlüssel für Manipulationen des BASIC, z.B. für ein Renumbering (Zuweisung neuer Zeilennummern) oder für einen REM-DELETER, der in dieser Ausgabe für den PET abgedruckt wurde.

In der für das AIM-BASIC bisher verfügbaren Dokumentation ist nicht angegeben, wie Variable, Daten, per Programm zur dauerhaften Speicherung auf andere Peripheriegeräte als Display und Printer ausgegeben werden können und wie Daten einem BASIC-Programm z.B. von einer Magnetbandaufzeichnung (einem DATAFILE) zur Verfügung gestellt werden können. Ersichtlich ist bisher nur die Tastatureingabe mit INPUT oder mit GET. Man kann sich nicht vorstellen, daß der für den Verkehr mit beliebigen Peripherieeinheiten vorbereitete AIM hinsichtlich seines BASIC nur als Tischrechner betrieben werden könnte. Gemeint ist also nicht das Programmieren und das Abspeichern von Programmen, sondern die Datenverarbeitung gespeicherter Datensätze, z.B. das Absuchen einer Adreßdatei oder die Verarbeitung von Meßwerten.

Es fehlt hier mindestens an ausreichender Dokumentation, und es ist den zahllosen Anwendern nicht zuzumuten, daß sie diese Dienstleistungen in mühsamer Kleinarbeit selbst erarbeiten. Die konkrete Frage lautet, wie kann man eine Variable A oder A\$ dauerhaft abspeichern, wie kann man (außer von der Tastatur her) in eine solche Variable hineinlesen?

Sofern diese Dinge nicht dokumentiert oder vorgesehen sind, wird ein Ansatzpunkt möglicherweise in der Beeinflussung der In- und Outflags in A412/13 mittels POKE-Befehlen liegen. Hinsichtlich der Eingabe von BASIC-Statements mittels LOAD-Befehlen ist Beweglichkeit gegeben. So kann man z.B. mit LOAD IN=U das Programm zur Kleinschreibung in der AIM-Tastatur (in diesem Heft) benutzen und z.B. in PRINT"..."-Befehlen Kleinschreibung vorsehen. Eine Eingabe mit IN=U wird abgeschlossen durch das gleichzeitige Drücken der Tasten CTRL und Z. In diesem Zusammenhang ist zu erwähnen, das das Kleinschreibprogramm noch keinen Softwareschalter für SHIFT LOCK hat, zur Umschaltung grundsätzlich auf Großschreibung, die für die meisten BASIC-Statements erforderlich ist. Allerdings ist es ganz lustig zu beobachten, daß auch einige kleingeschriebene Anweisungen korrekt ausgeführt werden.

Eine Eingabemöglichkeit für BASIC-Programme besteht auch vom Editor her. Dokumentiert ist das Einlesen einer vom Editor mit LIST OUT=T beschriebenen Tonbandcassette. Es gibt aber einen zweiten Weg, direkt aus dem im Memory residenten Editor, so wie es das Assembler-ROM bei IN=M macht. Nach einigen telefonischen Anfragen stellte der Herausgeber fest, daß auch hierfür Interesse besteht. Also den Vektor DO FA in dieser Reihenfolge als User Input-Vector nach 0108/0109 bringen. In BASIC sagt man dann LOAD IN=U. Zwei Besonderheiten sind nur zu beachten: Im Editor-Quellentext muß das erste Zeichen ein Zwischenraum sein, SPACE, denn dieses wird beim Laden abgeschnitten. Die letzte Eingabezeile des Editors muß wiederum das Abschlußzeichen CTRL/Z enthalten. Das Leseprogramm des BASIC filtert leider die Eingabe. Es unterdrückt Steuerzeichen z.B. für das Bildschirmdisplay, die der Cursor- oder Hell/Dunkelsteuerung dienen. Man kann (anders als beim PET) solche Steuerzeichen nicht in PRINT"..."-Anweisungen aufnehmen. Bei Bedarf besteht der Ausweg in CHR\$( ).  
*Beim Einlesen wie vor zunächst den Editor mit 'T' auf oberste Zeile stellen!*  
 Am Ende des ersten Monitor-ROMs ab Adresse EFB2 ist ein kryptographischer Text wie folgt enthalten: N IT IN SIGN OF A (MSB) in der Folgezeile dann RDBIT LDA TSPEED ;ARE WE IN O OR 1,6 SPEED letzte Zeile BMI <EA>;RDBIT4H. Man hätte eigentlich gewünscht, daß dieser Platz nicht verschenkt und z.B. für die Implementierung der Kleinschreibung benutzt worden wäre.

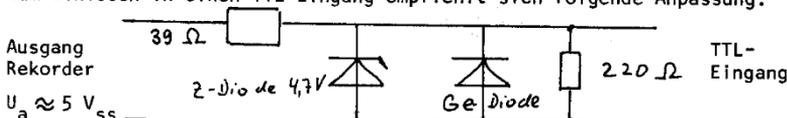
Wichtiger Hinweis! Viele Leser hatten Schwierigkeiten mit dem Assembler und mit BASIC, weil sie das Tape-GAP in Adresse A409 nicht manuell ausreichend auf hex 20 bis 80 einsetzten (vor dem Bandschreiben!). Also bitte das GAP beachten! R.L.

65<sub>xx</sub> MICRO MAG

Datenaustausch zwischen KIM-1 und TRS-80

Im Anschluß an seinen Artikel in Heft 7 gibt Herr Dr. Wünsche die Empfehlung, Anpassungsschaltungen zum Schutz der Ein- und Ausgänge zu verwenden. Bei direkter Ankoppelung des Kassettenrekorderausganges an TTL tritt eine Anhebung des Nullpegels ein, weil der Strom von 1 mA bei Low nicht abfließen kann. Man verwende hier einen Spannungsteiler, der dem Rekorder anzupassen ist.

Zum Einlesen in einen TTL-Eingang empfiehlt sich folgende Anpassung:



Die Z-Diode hält positive Spannungsspitzen unter 5 V, die Ge-Diode begrenzt die negativen auf 0,3 V. Die 220 Ω verhindern eine unzulässig hohe Anhebung des Nullpegels bei Low. Ausreichende Ausgangsspannung des Rekorders vorausgesetzt.

## P A S C A L - C O M P I L E R F U E R M C 6502

IMPLEMENTIERT IST EINE UNTERMENGE VON STANDARD-PASCAL;  
LAUFFAEHIG AUF 6502-SYSTEMEN (KIM-1) DIE UEBER  
16-KB-RAM-SPEICHER AB ADRESSE '2000 VERFUEGEN.

### KURZDATEN:

- => ERZEUGUNG VON MASCHINENCODE IN EINEM DURCHGANG  
(GESCHWINDIGKEIT CA. 30 STATEMENTS PRO SEKUNDE)
- => GENAUIGKEIT: 12 STELLEN BEI DEN GRUNDRECHENARTEN
- => VARIABLENTYPEN: INTEGER, REAL, FELDER
- => ANWEISUNGEN: REPEAT, WHILE, FOR, IF
- => SPEICHERN UND LADEN VON PROGRAMMEN MIT NAMEN UEBER KASSETTE
- => DIALOGFAEHIG DURCH INTEGRIERTEN EDITOR

### ZUM LIEFERUMFANG GEHOEREN:

- => COMPILER 8-KB
- => BETRIEBSSYSTEM 1,5-KB
- => ASSEMBLER 2-KB
- => BESCHREIBUNG DER SYNTAX
- => BENUTZERANLEITUNG
- => SPEICHERBELEGUNGSTABELLE

AUF KASSETTE IM KIM-FORMAT

PREIS: DM 320,00

### INFORMATIONEN VON:

MANFRED FLAKOWSKI : UNTERE GASSE 10 : 7411-TROCHTELFINGEN-1  
TEL.: DUISBURG 0203/660867 AB 19.00 UHR  
( MO - DO ) ( BIS 14.09.79 )  
TEL.: TROCHTELFINGEN 07124/868 AB 19.00 UHR  
TEL.: VON 8.00 - 17.00 0208/4434304 (BIS 14.09.1979)  
07121/82333 (AB 16.09.1979)

## AJWJ Gehäuse

### ENCLOSURE

DM 148,-

### POWER SUPPLY

DM 365,-

INPUT: 230 VAC 50 Hz

OUTPUT: +5V @ 5A

+24V @ 1A

DM 1050,-

AIM (1k) DM 875,-

(4k) DM 1050,-

additional board

MEMORY PLUS

FOR

AIM/SYMIKIM

8K STATIC RAM

LOW POWER

Sockets for 8K Eprom

6522 1/0 Port

ON BOARD REGULATORS

EPROM

PROGRAMMER

MEMORY PLUS: DM 668,- (ONLY ASSEMBLED AND TESTED)

Preise +12% MwSt.

Gebrauchte TTY (ASR33) DM 750,-

M & R Nedela  
Postfach 1122  
Marktstraße 3  
7778 Markdorf

**MSB** Micro-Shop-Bodensee

# INSTANT SOFTWARE

Peterborough, New Hampshire 03458

**INSTANT SOFTWARE**

**Business Package II**

**INSTANT SOFTWARE**

**Car Race Anti-Aircraft Rat Trap**

Would you like to be able to graph the sales of your firm... or a particular product? This program will do this automatically, and save the data for later use or modification. It will also calculate and graph the average sales for a year period, month by month... and calculate and graph the increase or decrease in sales. This program alone is worth the price of the entire package. **DM 69,- 90**

In Car Race you and a friend can have fun using your data on a checkered flag. It works with your TRS 80. In Rat Trap you must trap the cat in the grid using your trap. Car Race Aim your gun and shoot down the plane in the Antiaircraft game. **DM 24,- 90**

Händleranfragen willkommen.

M & R Nedela  
Postfach 1122  
Marktstraße 3  
7778 Markdorf

**MSB** Micro-Shop-Bodensee

32768 × 8 - 1 KBYTE FÜR DM 25,-

ANSCHLUSSFERTIGE SPEICHERKARTEN,  
BESTÜCKT UND GETESTET MIT 32 KBYTE  
IM "EUROPAFORMAT" 100 × 160 mm  
FÜR AIM 65, KIM-1, SYM, ALPHA, PET2001, MC6800

VERSORGUNGSSPANNUNG +5V (1A)

Adressen wählbar in unabhängigen 2k-Blöcken im 64k-Bereich.  
Keine Wait- oder Refresh-Zyklen.

Durch Verwendung hochzuverlässiger RAMs und erstmalig eingesetzte strukturierte Testverfahren, die Hard- und Soft-Errors erfassen, ergibt sich außer der auf 1 Jahr verlängerten Garantie noch ein erheblicher Preisvorteil:

- \* DM 595,- 6502/6800-Version für KIM, AIM 65, SYM, ALPHA und für 6800-Systeme, bestückt mit 16 kB.
- \* DM 896,- dito, mit 32 kB bestückt.  
Spätere Erweiterung einschl. Test: Differenzpreis + DM 30,-.
- \* DM 878,- für PET2001, ergänzt den PET-RAM auf 32 kB mit passender Adressierung.

Nähere Informationen kostenlos - bitte anfragen.

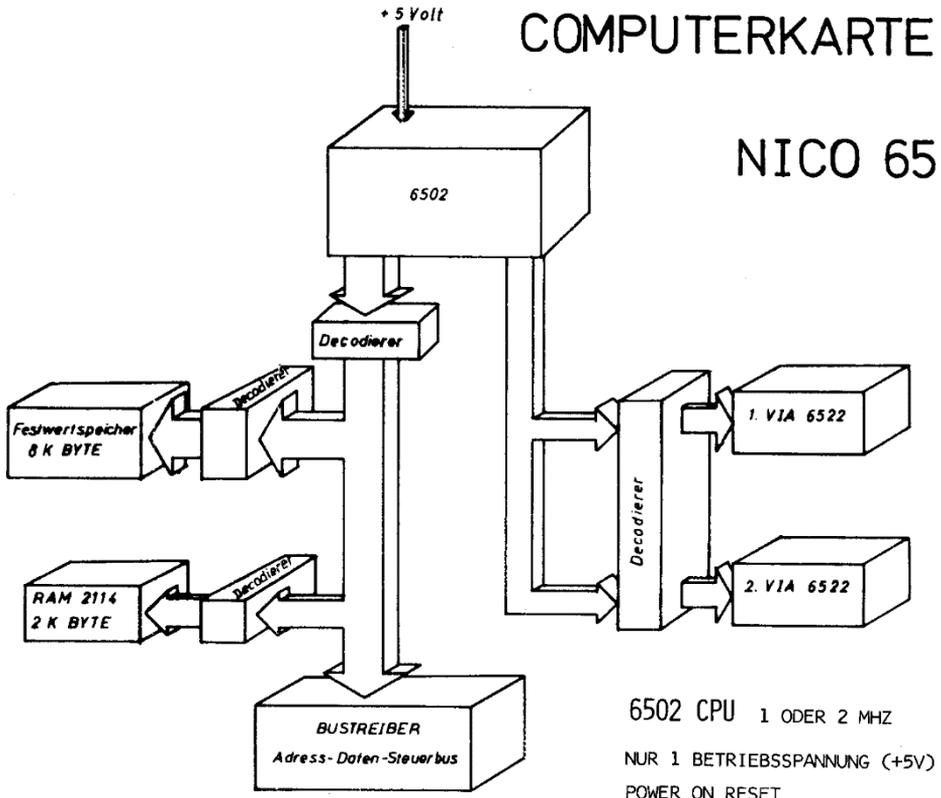
**DIPL.-ING. HORST NEUDECKER**  
**INGENIEURBÜRO FÜR MESSTECHNIK**

MEHRINGPLATZ 13, POSTFACH 151  
1000 BERLIN 61

TEL: 030 - 61 48 900  
030 - 25 12 000

# COMPUTERKARTE

## NICO 65



6502 CPU 1 ODER 2 MHz

NUR 1 BETRIEBSSPANNUNG (+5V)

POWER ON RESET

8 K EPROM mit Anwenderprogramm, Adreßbereich über Lötbrücke festlegbar

2 K RAM, statisch

2 INTERFACEBAUSTEINE 6522 VIA mit 40 I/O-Leitungen, 4 Timern, 2 Zählern, Handshake-Betrieb, 2 Schieberegistern, 7 Interruptmöglichkeiten

TREIBER FÜR ADRESS-, DATEN- UND STEUERBUS

VG64 STECKERLEISTE für Erweiterungen

Leiterbahnen glanzverzinnt, Stecksockel. Expandierbar auf gesamten Prozessorbereich, gleich 64 k.

PREIS: DM 390,- +MwST, in Grundausstattung bestückt

ALS ERWEITERUNG SIND LIEFERBAR:

Combo-Karte mit 8 k RAM und 2 Stück 6522 VIA (siehe oben),  
TV-Interfacekarte mit 2 k Bildschirmspeicher in Vorbereitung

8 Bit A/D Wandler

8 Bit D/A Wandler

Motherboard für VG64 (aller Systeme)  
mit 8 Einschüben lieferbar.

### I. DOHMANN

ELEKTRONISCHE BAUGRUPPEN

IM WIEHAGEN 15

4830 GÜTERSLOH 12

TEL.: 05241 - 67 480

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

## Rockwell AIM 65

Das vollständige Klein-  
Entwicklungssystem erlaubt  
erstmalig alphanumerische  
Ein-/Ausgabe ohne Verwen-  
dung 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
- Lieferbar

### 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 Versandkosten.

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

# GWK

FÜR TECHNISCHE ELEKTRONIK mbH.

GESELLSCHAFT

# 65<sub>xx</sub> 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: DRUCKARTIST GERHARD M. MEIER, HAMBURG 70

**BEZUGSBEDINGUNGEN:** Abonnement für 6 Hefte im Inland DM 40,- (Endpreis). Ausland/Foreign: DM 46,- (surface mail). Firmen erhalten Rechnung. Rechnungserteilung sonst nur auf Wunsch. Richten Sie bitte Ihre Überweisung/Euroscheck an:

Roland Löhr, Konto 20/01121, Vereins- und Westbank, BLZ 200 300 00.

Alle früher erschienenen Hefte dieser Zeitschrift sind weiterhin lieferbar. Einzelne Hefte können zu DM 7,- inkl. Porto nachbezogen werden.

AIM 65/PC100: WOCHENENDWORKSHOP (3 TAGE INTENSIV) 28.-30.9.79  
IN FRIEDRICHSDORF BEI FRANKFURT UNTER DER LEITUNG DES HERAUSGEBERS. 1/3 GRUNDLAGEN UND HARDWARE, 2/3 PROGRAMMIERUNG AM GERÄT. BILDSCHIRMMONITORE AM AIM DER DOZENTEN. SONDERPROSPEKT MIT PROGRAMM, AUSKUNFT UND ANMELDUNG BEIM HERAUSGEBER.

OFFENES 65XX COMPUTERMEETING AM 13./14. OKTOBER 1979 BEI FRANKFURT. NÄHERES AUF SEITE 35.



THERMOPAPIER FÜR DEN AIM 65

MIT QUALITÄTSFREIGABE DURCH ROCKWELL, CA.

Lösen Sie die Nachschubfrage bequem auf dem Postwege:

1 Packung (Versandeinheit) mit 8 Thermorollen in kontrastreicher Spitzenqualität, 65 m je Rolle, zus. ca. 520 m Streifenlänge, 57 mm breit, inkl. Versandkosten und USt.

Vorkassepreis DM 50,85  
als Nachnahme DM 52,35



AIM 65 - ANWENDERHANDBUCH IN DEUTSCHER SPRACHE

Original Rockwell-Handbuch mit über 100 Verbesserungen und Nachträgen gegenüber der englischen Vorlage, daher noch zuverlässiger. Erscheinungstermin etwa Ende Juni 1979.  
Preise inkl. Versandkosten und USt.

Vorkassepreis (Überweisung/Scheck)) DM 32,10  
Nachnahmeversand DM 33,60

ROLAND LÖHR. ANSCHRIFT UND KONTO WIE OBEN.