

ANHANG B

Spezial-Adressen

Tabelle 1: Adressen des Tastenfeldes

Adresse		Beschreibung
Hexa	Dezimal	
\$C000	49152 -16384	Tastendruckspeicher
\$C010	49168 -16368	Tastendruck löschen

Tabelle 4: Bildschirm-Speicherbereiche

Modus	Seite	Beginn		Ende	
		Hexa	Dezimal		
Text/Lo-Res	1	\$400	1024	\$7FF	2047
	2	\$800	2048	\$BFF	3071
Hi-Res	1	\$2000	8192	\$3FFF	16383
	2	\$4000	16384	\$5FFF	24575

Tabelle 5: Bildmodus-Schalter

Adresse		Beschreibung
Hexa	Dezimal	
\$C050	49232 -16304	Anzeige auf GRAPHIK-Modus.
\$C051	49233 -16303	Anzeige auf TEXT-Modus.
\$C052	49234 -16302	Anzeige auf: nur TEXT oder nur GRAPHIK.
\$C053	49235 -16301	Gemischte Anzeige: TEXT und GRAPHIK.
\$C054	49236 -16300	Anzeige der 1. Seite.
\$C055	49237 -16299	Anzeige der 2. Seite.
\$C056	49238 -16298	Anzeige auf Lo-Res GRAPHIK.
\$C057	49239 -16297	Anzeige auf Hi-Res GRAPHIK.

Tabelle 9: Adressen für die Signal-Ausgänge				
Nr.	Zustand	Adresse		Hexa
		Dezimal	Hexa	
Ø	aus	49240	-16296	§CØ58
	an	49241	-16295	§CØ59
1	aus	49242	-16294	§CØ5A
	an	49243	-16293	§CØ5B
2	aus	49244	-16292	§CØ5C
	an	49245	-16291	§CØ5D
3	aus	49246	-16290	§CØ5E
	an	49247	-16289	§CØ5F

Tabelle 10: Ein-/Ausgabe-Adressen				
Funktion	Adresse			Lesen/Schreiben
	Dezimal	Hexa	Hexa	
Lautsprecher	492ØØ	-16336	§CØ3Ø	L
Kassetten-Eingang	49256	-16288	§CØ6Ø	L
Kassetten-Ausgang	49184	-16352	§CØ2Ø	L
Signal-Ausgänge	4924Ø	-16296	§CØ58	L/S
	bis	bis	bis	
	49247	-16289	§CØ5F	
Ein-Bit-Eingänge	49249	-16287	§CØ61	L
	4925Ø	-16286	§CØ62	L
	49251	-16285	§CØ63	L
Analog-Eingänge	49252	-16284	§CØ64	L
	49253	-16283	§CØ65	
	49254	-16282	§CØ66	
	49255	-16281	§CØ67	
Analog Rücksetzen	49264	-16272	§CØ7Ø	L/S
Impuls-Ausgang	49216	-1632Ø	§CØ4Ø	L

Tabelle 11: Speicherstellen für das Text-Fenster				
Funktion	Adresse		Minimum/Normal/Maximum	
	Dezimal	Hexa	Dezimal	Hexa
Linke Kante	32	\$20	0/0/39	\$0/\$0/\$17
Breite	33	\$21	0/40/40	\$0/\$28/\$28
Obere Kante	34	\$22	0/0/24	\$0/\$0/\$18
Untere Kante	35	\$23	0/24/24	\$0/\$18/\$18

Tabelle 12: Ausgabekontrolle durch Speicherstelle 50		
Wert		Effekt
Dezimal	Hexa	
255	\$FF	COUT gibt Zeichen im "NORMAL"-Modus aus.
63	\$3F	COUT gibt Zeichen im "INVERSE"-Modus aus.
127	\$7F	COUT gibt Buchstaben im "FLASH"-Modus aus und alle anderen Zeichen im "INVERSE"-Modus.

Tabelle 13: Spezielle Adressen für den Autostart-ROM		
Adresse		Inhalt
Dezimal	Hexa	
1010	\$3F2	Eingangsadresse. Diese zwei Speicherstellen enthalten die Adresse für den Sprung in die Sprache, die gerade benutzt wird (Normalerweise ist es \$E003).
1011	\$3F3	
1012	\$3F4	Einschalt-Byte. (Normalerweise ist es \$45) Siehe unten.
64367 -1169	\$FB6F	Dies ist die Startadresse eines Unterprogramms in Maschinensprache, das das Einschaltbyte initialisiert.

Tabelle 14: Monitor-Adressen auf Seite 3

Adresse		Verwendung	
Dezimal	Hexa	Monitor-ROM	Autostart-ROM
1008	3F0	keine	Enthält die Adresse des Unterprogramms, das "BRK"-Nachfragen behandelt (normal: 3FA59).
1009	3F1		
1010	3F2	keine	Eingangsadresse in die benutzte Sprache.
1011	3F3		
1012	3F4	keine	Einschalt-Byte
1013	3F5	Enthält einen JMP (Sprung)-Befehl zu dem Unterprogramm, das Applesoft II "&"-Kommandos behandelt*. (Normal: 4C 58 FF)	
1014	3F6		
1015	3F7		
1016	3F8	Enthält einen JMP-Befehl zu dem Unterprogramm, das "USER" (CTRL Y)-Kommandos behandelt.	
1017	3F9		
1018	3FA		
1019	3FB	Enthält einen JMP-Befehl zu dem Unterprogramm, das nichtmaskierbare Interrupts behandelt.	
1020	3FC		
1021	3FD		
1022	3FE	Enthält die Adresse des Unterprogramms, das Interrupt-Nachfragen (IRQ) behandelt.	
1023	3FF		

Tabelle 22: Eingebaute Ein-/Ausgabe-Adressen

	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
\$C000	Tastendruckspeicher															
\$C010	Tastelöschimpuls															
\$C020	Kassettenausgang															
\$C030	Lautsprecher															
\$C040	Impulsausgang des Spielanschlusses															
\$C050	gr	tx	un	mi	er	zw	lo	hi	sig0	sig1	sig2	sig3				
\$C060	ke	d1	d2	d3	a0	a1	a2	a3	wie \$C060-\$C067							
\$C070	Spielsteuerrücksetzimpuls															

Erläuterung der Abkürzungen:

gr	setze Graphik-Zustand	tx	setze Text-Zustand
un	nur Text oder nur Graphik	mi	setze Text und Graphik gemischt
er	Anzeige der ersten Seite	zw	Anzeige der zweiten Seite
lo	Anzeige auf Lo-Res Graphik	hi	Anzeige auf Hi-Res Graphik
sig	Signalausgänge	d	Drucktasteneingänge
a	Analogeingänge	ke	Kassetteneingang

Tabelle 23: Ein-/Ausgabebereiche für Geräte-Anschlußkarten

	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
\$C080	Ein-/Ausgabebereiche für Karten-Nummer															
\$C090																
\$C0A0																
\$C0B0																
\$C0C0																
\$C0D0																
\$C0E0																
\$C0F0																

Tabelle 24: PROM-Bereiche für Ein-/Ausgabe-Gerätekarten															
	\$00	\$10	\$20	\$30	\$40	\$50	\$60	\$70	\$80	\$90	\$A0	\$B0	\$C0	\$E0	\$F0
\$C100														1	
\$C200														2	
\$C300														3	
\$C400	PROM-Speicherbereich für Karten-Nummer													4	
\$C500														5	
\$C600														6	
\$C700														7	

Tabelle 25: Ein-/Ausgabe-Basisadressen								
Basis-Adresse	Einsteckschlitz							
	0	1	2	3	4	5	6	7
\$C080	\$C080	\$C090	\$C0A0	\$C0B0	\$C0C0	\$C0D0	\$C0E0	\$C0F0
\$C081	\$C081	\$C091	\$C0A1	\$C0B1	\$C0C1	\$C0D1	\$C0E1	\$C0F1
\$C082	\$C082	\$C092	\$C0A2	\$C0B2	\$C0C2	\$C0D2	\$C0E2	\$C0F2
\$C083	\$C083	\$C093	\$C0A3	\$C0B3	\$C0C3	\$C0D3	\$C0E3	\$C0F3
\$C084	\$C084	\$C094	\$C0A4	\$C0B4	\$C0C4	\$C0D4	\$C0E4	\$C0F4
\$C085	\$C085	\$C095	\$C0A5	\$C0B5	\$C0C5	\$C0D5	\$C0E5	\$C0F5
\$C086	\$C086	\$C096	\$C0A6	\$C0B6	\$C0C6	\$C0D6	\$C0E6	\$C0F6
\$C087	\$C087	\$C097	\$C0A7	\$C0B7	\$C0C7	\$C0D7	\$C0E7	\$C0F7
\$C088	\$C088	\$C098	\$C0A8	\$C0B8	\$C0C8	\$C0D8	\$C0E8	\$C0F8
\$C089	\$C089	\$C099	\$C0A9	\$C0B9	\$C0C9	\$C0D9	\$C0E9	\$C0F9
\$C08A	\$C08A	\$C09A	\$C0AA	\$C0BA	\$C0CA	\$C0DA	\$C0EA	\$C0FA
\$C08B	\$C08B	\$C09B	\$C0AB	\$C0BB	\$C0CB	\$C0DB	\$C0EB	\$C0FB
\$C08C	\$C08C	\$C09C	\$C0AC	\$C0BC	\$C0CC	\$C0DC	\$C0EC	\$C0FC
\$C08D	\$C08D	\$C09D	\$C0AD	\$C0BD	\$C0CD	\$C0DD	\$C0ED	\$C0FD
\$C08E	\$C08E	\$C09E	\$C0AE	\$C0BE	\$C0CE	\$C0DE	\$C0EE	\$C0FE
\$C08F	\$C08F	\$C09F	\$C0AF	\$C0BF	\$C0CF	\$C0DF	\$C0EF	\$C0FF

Ein-/Ausgabe-Adressen

Tabelle 26: Ein-/Ausgabe-Zwischenspeicher-Adressen							
Basis-Adresse	Anschlußnummer						
	1	2	3	4	5	6	7
\$0470	\$0479	\$047A	\$047B	\$047C	\$047D	\$047E	\$047F
\$04F0	\$04F9	\$04FA	\$04FB	\$04FC	\$04FD	\$04FE	\$04FF
\$0570	\$0579	\$057A	\$057B	\$057C	\$057D	\$057E	\$057F
\$05F0	\$05F9	\$05FA	\$05FB	\$05FC	\$05FD	\$05FE	\$05FF
\$0670	\$0679	\$067A	\$067B	\$067C	\$067D	\$067E	\$067F
\$06F0	\$06F9	\$06FA	\$06FB	\$06FC	\$06FD	\$06FE	\$06FF
\$0770	\$0779	\$077A	\$077B	\$077C	\$077D	\$077E	\$077F
\$07F0	\$07F9	\$07FA	\$07FB	\$07FC	\$07FD	\$07FE	\$07FF

ANHANG C

Rom-Programme

Autostart-Rom
Monitor-Rom

143
169

AUTOSTART ROM

```
0000:          2 *****
0000:          3 4
0000:          4 * APPLE II
0000:          5 * MONITOR II
0000:          6 *
0000:          7 * COPYRIGHT 1978 BY
0000:          8 * APPLE COMPUTER, INC.
0000:          9 *
0000:         10 * ALL RIGHTS RESERVED
0000:         11 *
0000:         12 * STEVE WOZNIAK
0000:         13 *
0000:         14 *****
0000:         15 *
0000:         16 * MODIFIED NOV 1978
0000:         17 * BY JOHN A
0000:         18 *
0000:         19 *****
FB00:         20          DRG $FB00
FB00:         21          OBJ $2000
FB00:         22 *****
FB00:         23 LOCO    EQU $00
FB00:         24 LOC1    EQU $01
FB00:         25 WNDLFT  EQU $20
FB00:         26 WNDWDTH EQU $21
FB00:         27 WNDTOP  EQU $22
FB00:         28 WNDBTM  EQU $23
FB00:         29 CH     EQU $24
FB00:         30 CV     EQU $25
FB00:         31 GBASL  EQU $26
FB00:         32 GBASH  EQU $27
FB00:         33 BASL   EQU $28
FB00:         34 BASH  EQU $29
FB00:         35 BAS2L  EQU $2A
FB00:         36 BAS2H  EQU $2B
FB00:         37 H2    EQU $2C
FB00:         38 LMNEM  EQU $2C
FB00:         39 V2    EQU $2D
FB00:         40 RMNEM  EQU $2D
FB00:         41 MASK  EQU $2E
FB00:         42 CHKSUM EQU $2E
FB00:         43 FORMAT EQU $2E
FB00:         44 LASTIN EQU $2F
FB00:         45 LENGTH EQU $2F
FB00:         46 SIGN  EQU $2F
FB00:         47 COLOR  EQU $30
FB00:         48 MODE  EQU $31
FB00:         49 INVFLG EQU $32
FB00:         50 PROMPT EQU $33
FB00:         51 YSAV  EQU $34
FB00:         52 YSAV1 EQU $35
FB00:         53 CSWL  EQU $36
FB00:         54 CSWH  EQU $37
FB00:         55 KSWL  EQU $38
FB00:         56 KSWH  EQU $39
FB00:         57 PCL   EQU $3A
FB00:         58 PCH   EQU $3B
FB00:         59 A1L   EQU $3C
FB00:         60 A1H   EQU $3D
FB00:         61 A2L   EQU $3E
FB00:         62 A2H   EQU $3F
FB00:         63 A3L   EQU $40
FB00:         64 A3H   EQU $41
FB00:         65 A4L   EQU $42
FB00:         66 A4H   EQU $43
FB00:         67 A5L   EQU $44
FB00:         68 A5H   EQU $45
```

F800	69	ACC	EGU \$45	; N0TE 0VERLAP WITH A5H!	*1
F800	70	XREG	EGU \$46		
F800	71	YREG	EGU \$47		
F800	72	STATUS	EGU \$48		
F800	73	SPNT	EGU \$49		
F800	74	RNDL	EGU \$4E		
F800	75	RNDH	EGU \$4F		
F800	76	PICK	EGU \$95		
F800	77	IN	EGU \$0200		
F800	78	BRKV	EGU \$3F0	; NEW VECTOR FOR BRK	*2
F800	79	SOFTEV	EGU \$3F2	; VECTOR FOR WARM START	*3
F800	80	PWREDUP	EGU \$3F4	; THIS MUST = EDR **A5 OF SOFTEV+1	*4
F800	81	AMPERV	EGU \$3F5	; APPLESOFT & EXIT VECTOR	*5
F800	82	USRADR	EGU \$03FB		
F800	83	NMI	EGU \$03FB		
F800	84	IRQL0C	EGU \$3FE		
F800	85	LINE1	EGU \$400		
F800	86	MSLOT	EGU \$07FB		
F800	87	IDADR	EGU \$C000		
F800	88	KBD	EGU \$C000		
F800	89	KBDSTRB	EGU \$C010		
F800	90	TAPE0UT	EGU \$C020		
F800	91	SPKR	EGU \$C030		
F800	92	TXTCLR	EGU \$C050		
F800	93	TXTSET	EGU \$C051		
F800	94	MIXCLR	EGU \$C052		
F800	95	MIXSET	EGU \$C053		
F800	96	LOWSCR	EGU \$C054		
F800	97	HISCR	EGU \$C055		
F800	98	LORES	EGU \$C056		
F800	99	HIRES	EGU \$C057		
F800	100	SETAN0	EGU \$C058		
F800	101	CLRAN0	EGU \$C059		
F800	102	SETAN1	EGU \$C05A		
F800	103	CLRAN1	EGU \$C05B		
F800	104	SETAN2	EGU \$C05C		
F800	105	CLRAN2	EGU \$C05D		
F800	106	SETAN3	EGU \$C05E		
F800	107	CLRAN3	EGU \$C05F		
F800	108	TAPEIN	EGU \$C060		
F800	109	PADDL0	EGU \$C064		
F800	110	PTRIG	EGU \$C070		
F800	111	CLRR0M	EGU \$CFFF		
F800	112	BASIC	EGU \$E000		
F800	113	BASIC2	EGU \$E003		
F800	114		PAGE		
F800	4A	115	PLOT	LSR A	
F801	08	116		PHP	
F802	20 47 FB	117		JSR GBASCALC	
F805	28	118		PLP	
F806	49 0F	119		LDA #\$0F	
F808	90 02	120		BCC RTMASK	
F80A	69 E0	121		ADC #\$E0	
F80C	85 2E	122		RTMASK STA MASK	
F80E	B1 2E	123		PLOT1 LDA (GBASL),Y	
F810	45 30	124		EDR COLDR	
F812	25 2E	125		AND MASK	
F814	51 26	126		EDR (GBASL),Y	
F816	91 26	127		STA (GBASL),Y	
F818	60	128		RTS	
F819	20 00 FB	129		HLINE JSR PLOT	
F81C	C4 2C	130		HLINE1 CPY H2	
F81E	B0 11	131		BCS RTS1	
F820	C8	132		INY	
F821	20 0E FB	133		JSR PLOT1	
F824	90 F6	134		BCC HLINE1	
F826	65 01	135		VLINEZ ADC #\$01	
F828	4E	136		VLINE PHA	
F829	20 00 FB	137		JSR PLOT	
F82C	68	138		PLA	
F82D	C5 2D	139		CMP V2	
F82F	90 F5	140		BCC VLINEZ	
F831	60	141		RTS1 RTS	

1. Beachten Sie die Übertagerung mit A5H!
2. Neue Adresse für BRK.
3. Verzweigungsadresse für den Warmstart.
4. Der Inhalt muß SOFTEV+1 EOR #\$A5 sein.
5. Eingangsadresse für Applesoft oder andere Sprachen.

F832:	A0 2F	142	CLRSCR	LDY	##2F
F834:	DC 02	143		BNE	CLRSC2
F836:	A0 27	144	CLRTOP	LDY	##27
F83B:	B4 2D	145	CLRSC2	STY	V2
F83A:	A0 27	146		LDY	##27
F83C:	A9 00	147	CLRSC3	LDA	##00
F83E:	B5 30	148		STA	COLOR
F840:	20 2B FB	149		JSR	VLINE
F843:	B8	150		DEY	
F844:	10 F6	151		BPL	CLRSC3
F846:	60	152		RTS	
F847:		153		PAGE	
F847:	4B	154	GBASCALC	PHA	
F848:	4A	155		LSR	A
F849:	29 03	156		AND	##03
F84B:	09 04	157		ORA	##04
F84D:	B5 27	158		STA	GBASH
F84F:	6E	159		PLA	
F850:	29 1B	160		AND	##1B
F852:	90 02	161		BCC	GBCALC
F854:	69 7F	162		ADC	##7F
F856:	B5 26	163	GBCALC	STA	GBASL
F858:	0A	164		ASL	A
F859:	0A	165		ASL	A
F85A:	05 26	166		ORA	GBASL
F85C:	B5 26	167		STA	GBASL
F85E:	60	168		RTS	
F85F:	A5 30	169		LDA	COLOR
F861:	1B	170		CLC	
F862:	69 03	171		ADC	##03
F864:	29 0F	172	SETCOL	AND	##0F
F866:	B5 30	173		STA	COLOR
F868:	0A	174		ASL	A
F869:	0A	175		ASL	A
F86A:	0A	176		ASL	A
F86B:	0A	177		ASL	A
F86C:	05 30	178		ORA	COLOR
F86E:	B5 30	179		STA	COLOR
F870:	60	180		RTS	
F871:	4A	181	SCRN	LSR	A
F872:	0B	182		PHP	
F873:	20 47 FB	183		JSR	GBASCALC
F876:	B1 26	184		LDA	(GBASL), Y
F878:	2B	185		PLP	
F879:	90 04	186	SCRN2	BCC	RTMSKZ
F87B:	4A	187		LSR	A
F87C:	4A	188		LSR	A
F87D:	4A	189		LSR	A
F87E:	4A	190		LSR	A
F87F:	29 0F	191	RTHSKZ	AND	##0F
F881:	60	192		RTS	
F882:		193		PAGE	
F882:	A6 3A	194	INSDS1	LDX	PCL
F884:	A4 3B	195		LDY	PCH
F886:	20 96 FD	196		JSR	PRYX2
F889:	20 4B F9	197		JSR	PRBLNK
F88C:	A1 3A	198	INSDS2	LDA	(PCL, X)
F88E:	AB	199		TAY	
F88F:	4A	200		LSR	A
F890:	90 09	201		BCC	IEVEN
F892:	6A	202		RDR	A
F893:	B0 10	203		BCS	ERR
F895:	C9 A2	204		CMP	##A2
F897:	F0 0C	205		BEG	ERR
F899:	29 B7	206		AND	##B7
F89B:	4A	207	IEVEN	LSR	A
F89C:	AA	208		TAX	
F89D:	BD 62 F9	209		LDA	FMT1, X
F8A0:	20 79 FB	210		JSR	SCRN2
F8A3:	D0 04	211		BNE	GETFMT
F8A5:	A0 B0	212	ERR	LDY	##B0
F8A7:	A9 00	213		LDA	##00
F8A9:	AA	214	GETFMT	TAX	

F8AA:	BD A6 F9	215		LDA FMT2, X
F8AD:	85 2E	216		STA FORMAT
F8AF:	29 03	217		AND ##03
F8B1:	85 2F	218		STA LENGTH
F8B3:	98	219		TYA
F8B4:	29 BF	220		AND ##BF
F8B6:	AA	221		TAX
F8B7:	98	222		TYA
F8B8:	A0 03	223		LDY ##03
F8BA:	E0 8A	224		CPX ##8A
F8BC:	F0 0B	225		BEG MNNDX3
F8BE:	4A	226	MNNDX1	LSR A
F8BF:	90 0B	227		BCC MNNDX3
F8C1:	4A	228		LSR A
F8C2:	4A	229	MNNDX2	LSR A
F8C3:	09 20	230		ORA ##20
F8C5:	88	231		DEY
F8C6:	D0 FA	232		BNE MNNDX2
F8C8:	C8	233		INY
F8C9:	88	234	MNNDX3	DEY
F8CA:	D0 F2	235		BNE MNNDX1
F8CC:	60	236		RTS
F8CD:	FF FF FF	237		DFB \$FF, \$FF, \$FF
F8D0:		238		PAGE
F8D0:	20 82 FB	239	INSTDSP	JSR INSDS1
F8D3:	48	240		PHA
F8D4:	B1 3A	241	PRNTOP	LDA (PCL), Y
F8D6:	20 DA FD	242		JSR PRBYTE
F8D9:	A2 01	243		LDX ##01
F8DB:	20 4A F9	244	PRNTBL	JSR PRBL2
F8DE:	C4 2F	245		CPY LENGTH
F8E0:	C8	246		INY
F8E1:	90 F1	247		BCC PRNTOP
F8E3:	A2 03	248		LDX ##03
F8E5:	C0 04	249		CPY ##04
F8E7:	90 F2	250		BCC PRNTBL
F8E9:	68	251		PLA
F8EA:	AB	252		TAY
F8EB:	B9 C0 F9	253		LDA MNEML, Y
F8EE:	85 2C	254		STA LMNEM
F8F0:	B9 00 FA	255		LDA MNEMR, Y
F8F3:	85 2D	256		STA RMNEM
F8F5:	A9 00	257	NXTCOL	LDA ##00
F8F7:	A0 05	258		LDY ##05
F8F9:	06 2D	259	PRMN2	ASL RMNEM
F8FB:	26 2C	260		RDL LMNEM
F8FD:	2A	261		RDL A
F8FE:	88	262		DEY
F8FF:	D0 FB	263		BNE PRMN2
F901:	69 BF	264		ADC ##BF
F903:	20 ED FD	265		JSR COUT
F906:	CA	266		DEX
F907:	D0 EC	267		BNE NXTCOL
F909:	20 48 F9	268		JSR PRBLNK
F90C:	A4 2F	269		LDY LENGTH
F90E:	A2 06	270		LDX ##06
F910:	E0 03	271	PRADR1	CPX ##03
F912:	F0 1C	272		BEG PRADR5
F914:	06 2E	273	PRADR2	ASL FORMAT
F916:	90 0E	274		BCC PRADR3
F918:	BD B3 F9	275		LDA CHAR1-1, X
F918:	20 ED FD	276		JSR COUT
F91E:	BD B9 F9	277		LDA CHAR2-1, X
F921:	F0 03	278		BEG PRADR3
F923:	20 ED FD	279		JSR COUT
F926:	CA	280	PRADR3	DEX
F927:	D0 E7	281		BNE PRADR1
F929:	60	282		RTS
F92A:	88	283	PRADR4	DEY
F92B:	30 E7	284		BMI PRADR2
F92D:	20 DA FD	285		JSR PRBYTE
F930:	A5 2E	286	PRADR5	LDA FORMAT
F932:	C9 EB	287		CMP ##EB

F934:	B1 3A	288	LDA (PCL), Y
F936:	90 F2	289	BCC PRADR4
F938:		290	PAGE
F938:	20 56 F9	291	RELADR JSR PCADJ3
F93D:	AA	292	TAX
F93C:	EB	293	INX
F93D:	D0 01	294	BNE PRNTYX
F93F:	CB	295	INY
F940:	9B	296	PRNTYX TYA
F941:	20 DA FD	297	PRNTAX JSR PRBYTE
F944:	8A	298	PRNTAX TXA
F945:	4C DA FD	299	JMP PRBYTE
F948:	A2 03	300	PRBLNK LDX #03
F94A:	A9 A0	301	PRBL2 LDA #A0
F94C:	20 ED FD	302	PRBL3 JSR COUT
F94F:	CA	303	DEX
F950:	D0 FB	304	BNE PRBL2
F952:	60	305	RTS
F953:	3B	306	PCADJ SEC
F954:	A5 2F	307	PCADJ2 LDA LENGTH
F956:	A4 3B	308	PCADJ3 LDY PCH
F95B:	AA	309	TAX
F959:	10 01	310	BPL PCADJ4
F95B:	8E	311	DEY
F95C:	65 3A	312	PCADJ4 ADC PCL
F95E:	90 01	313	BCC RTS2
F960:	CB	314	INY
F961:	60	315	RTS2 RTS
F962:	04	316	FMT1 DFB \$04
F963:	20	317	DFB \$20
F964:	54	318	DFB \$54
F965:	30	319	DFB \$30
F966:	0D	320	DFB \$0D
F967:	80	321	DFB \$80
F968:	04	322	DFB \$04
F969:	90	323	DFB \$90
F96A:	03	324	DFB \$03
F96B:	22	325	DFB \$22
F96C:	54	326	DFB \$54
F96D:	33	327	DFB \$33
F96E:	0D	328	DFB \$0D
F96F:	80	329	DFB \$80
F970:	04	330	DFB \$04
F971:	90	331	DFB \$90
F972:	04	332	DFB \$04
F973:	20	333	DFB \$20
F974:	54	334	DFB \$54
F975:	33	335	DFB \$33
F976:	0D	336	DFB \$0D
F977:	80	337	DFB \$80
F978:	04	338	DFB \$04
F979:	90	339	DFB \$90
F97A:	04	340	DFB \$04
F97B:	20	341	DFB \$20
F97C:	54	342	DFB \$54
F97D:	3B	343	DFB \$3B
F97E:	0D	344	DFB \$0D
F97F:	80	345	DFB \$80
F980:	04	346	DFB \$04
F981:	90	347	DFB \$90
F982:	00	348	DFB \$00
F983:	22	349	DFB \$22
F984:	44	350	DFB \$44
F985:	33	351	DFB \$33
F986:	0D	352	DFB \$0D
F987:	CB	353	DFB \$CB
F988:	44	354	DFB \$44
F989:	00	355	DFB \$00
F98A:	11	356	DFB \$11
F98B:	22	357	DFB \$22
F98C:	44	358	DFB \$44
F98D:	33	359	DFB \$33
F98E:	0D	360	DFB \$0D

F98F:	CB	361		DFB	*CB
F990:	44	362		DFB	*44
F991:	A9	363		DFB	*A9
F992:	01	364		DFB	*01
F993:	22	365		DFB	*22
F994:	44	366		DFB	*44
F995:	33	367		DFB	*33
F996:	0D	368		DFB	*0D
F997:	80	369		DFB	*80
F998:	04	370		DFB	*04
F999:	90	371		DFB	*90
F99A:	01	372		DFB	*01
F99B:	22	373		DFB	*22
F99C:	44	374		DFB	*44
F99D:	33	375		DFB	*33
F99E:	0D	376		DFB	*0D
F99F:	80	377		DFB	*80
F9A0:	04	378		DFB	*04
F9A1:	90	379		DFB	*90
F9A2:	26	380		DFB	*26
F9A3:	31	381		DFB	*31
F9A4:	87	382		DFB	*87
F9A5:	9A	383		DFB	*9A
F9A6:	00	384	FMT2	DFB	*00
F9A7:	21	385		DFB	*21
F9A8:	81	386		DFB	*81
F9A9:	82	387		DFB	*82
F9AA:	00	388		DFB	*00
F9AB:	00	389		DFB	*00
F9AC:	59	390		DFB	*59
F9AD:	4D	391		DFB	*4D
F9AE:	91	392		DFB	*91
F9AF:	92	393		DFB	*92
F9B0:	86	394		DFB	*86
F9B1:	4A	395		DFB	*4A
F9B2:	85	396		DFB	*85
F9B3:	9D	397		DFB	*9D
F9B4:	AC	398	CHAR1	DFB	*AC
F9B5:	A9	399		DFB	*A9
F9B6:	AC	400		DFB	*AC
F9B7:	A3	401		DFB	*A3
F9B8:	AB	402		DFB	*AB
F9B9:	A4	403		DFB	*A4
F9BA:	D9	404	CHAR2	DFB	*D9
F9BB:	00	405		DFB	*00
F9BC:	DB	406		DFB	*DB
F9BD:	A4	407		DFB	*A4
F9BE:	A4	408		DFB	*A4
F9BF:	00	409		DFB	*00
F9C0:	1C	410	MNEML	DFB	*1C
F9C1:	8A	411		DFB	*8A
F9C2:	1C	412		DFB	*1C
F9C3:	23	413		DFB	*23
F9C4:	5D	414		DFB	*5D
F9C5:	8B	415		DFB	*8B
F9C6:	1B	416		DFB	*1B
F9C7:	A1	417		DFB	*A1
F9C8:	9D	418		DFB	*9D
F9C9:	8A	419		DFB	*8A
F9CA:	1D	420		DFB	*1D
F9CB:	23	421		DFB	*23
F9CC:	9D	422		DFB	*9D
F9CD:	8B	423		DFB	*8B
F9CE:	1D	424		DFB	*1D
F9CF:	A1	425		DFB	*A1
F9D0:	00	426		DFB	*00
F9D1:	29	427		DFB	*29
F9D2:	19	428		DFB	*19
F9D3:	AE	429		DFB	*AE
F9D4:	69	430		DFB	*69
F9D5:	AB	431		DFB	*AB
F9D6:	19	432		DFB	*19
F9D7:	23	433		DFB	*23

F9DB:	24	434	DFB	*24
F9DD:	53	435	DFB	*53
F9DA:	1B	436	DFB	*1B
F9DB:	23	437	DFB	*23
F9DC:	24	438	DFB	*24
F9DD:	53	439	DFB	*53
F9DE:	19	440	DFB	*19
F9DF:	A1	441	DFB	*A1
F9E0:	00	442	DFB	*00
F9E1:	1A	443	DFB	*1A
F9E2:	5B	444	DFD	*5B
F9E3:	5B	445	DFB	*5B
F9E4:	A5	446	DFB	*A5
F9E5:	69	447	DFB	*69
F9E6:	24	448	DFB	*24
F9E7:	24	449	DFB	*24
F9E8:	AE	450	DFB	*AE
F9E9:	AE	451	DFB	*AE
F9EA:	A8	452	DFD	*A8
F9EB:	AD	453	DFB	*AD
F9EC:	29	454	DFB	*29
F9ED:	00	455	DFB	*00
F9EE:	7C	456	DFB	*7C
F9EF:	00	457	DFB	*00
F9F0:	15	458	DFB	*15
F9F1:	9C	459	DFB	*9C
F9F2:	6D	460	DFB	*6D
F9F3:	9C	461	DFB	*9C
F9F4:	A5	462	DFB	*A5
F9F5:	69	463	DFB	*69
F9F6:	29	464	DFD	*29
F9F7:	53	465	DFB	*53
F9F8:	84	466	DFB	*84
F9F9:	13	467	DFB	*13
F9FA:	34	468	DFB	*34
F9FB:	11	469	DFB	*11
F9FC:	A5	470	DFB	*A5
F9FD:	69	471	DFB	*69
F9FE:	23	472	DFB	*23
F9FF:	A0	473	DFB	*A0
FA00:	DB	474	MNEMR	DFB *DB
FA01:	62	475	DFB	*62
FA02:	5A	476	DFB	*5A
FA03:	4B	477	DFB	*4B
FA04:	26	478	DFB	*26
FA05:	62	479	DFB	*62
FA06:	94	480	DFB	*94
FA07:	8B	481	DFB	*8B
FA08:	54	482	DFB	*54
FA09:	44	483	DFB	*44
FA0A:	C8	484	DFB	*C8
FA0B:	54	485	DFB	*54
FA0C:	6B	486	DFB	*6B
FA0D:	44	487	DFB	*44
FA0E:	EB	488	DFB	*EB
FA0F:	94	489	DFB	*94
FA10:	00	490	DFB	*00
FA11:	B4	491	DFB	*B4
FA12:	0B	492	DFB	*0B
FA13:	84	493	DFB	*84
FA14:	74	494	DFB	*74
FA15:	B4	495	DFB	*B4
FA16:	2B	496	DFB	*2B
FA17:	6E	497	DFB	*6E
FA18:	74	498	DFB	*74
FA19:	F4	499	DFB	*F4
FA1A:	CC	500	DFB	*CC
FA1B:	4A	501	DFB	*4A
FA1C:	72	502	DFB	*72
FA1D:	F2	503	DFB	*F2
FA1E:	A4	504	DFB	*A4
FA1F:	8A	505	DFB	*8A
FA20:	00	506	DFB	*00

FA21:	AA	507		DFB	*AA
FA22:	A2	508		DFB	*A2
FA23:	A2	509		DFB	*A2
FA24:	74	510		DFB	*74
FA25:	74	511		DFB	*74
FA26:	74	512		DFB	*74
FA27:	72	513		DFB	*72
FA28:	44	514		DFB	*44
FA29:	68	515		DFB	*68
FA2A:	B2	516		DFB	*B2
FA2B:	32	517		DFB	*32
FA2C:	B2	518		DFB	*B2
FA2D:	00	519		DFB	*00
FA2E:	22	520		DFB	*22
FA2F:	00	521		DFB	*00
FA30:	1A	522		DFB	*1A
FA31:	1A	523		DFB	*1A
FA32:	26	524		DFB	*26
FA33:	26	525		DFB	*26
FA34:	72	526		DFB	*72
FA35:	72	527		DFB	*72
FA36:	88	528		DFB	*88
FA37:	C8	529		DFB	*C8
FA38:	C4	530		DFB	*C4
FA39:	CA	531		DFB	*CA
FA3A:	26	532		DFB	*26
FA3B:	48	533		DFB	*48
FA3C:	44	534		DFB	*44
FA3D:	44	535		DFB	*44
FA3E:	A2	536		DFB	*A2
FA3F:	CB	537		DFB	*CB
FA40:		538		PAGE	
FA40:	B5 45	539	IRQ	STA	ACC
FA42:	68	540		PLA	
FA43:	48	541		PHA	
FA44:	0A	542		ASL	A
FA45:	0A	543		ASL	A
FA46:	0A	544		ASL	A
FA47:	30 03	545		BMI	BREAK
FA49:	6C FE 03	546		JMP	(IRQLQC)
FA4C:	28	547	BREAK	PLP	
FA4D:	20 4C FF	548		JSR	SAV1
FA50:	68	549		PLA	
FA51:	B5 3A	550		STA	PCL
FA53:	68	551		PLA	
FA54:	B5 3B	552		STA	PCH

FA56:	6C F0 03	553	JMP (BRKV) ; BRKV WRITTEN OVER BY DISK BOOT	*1
FA59:	20 82 FB	554	OLDBRK JSR INSDS1	
FA5C:	20 DA FA	555	JSR RQDSP1	
FA5F:	4C 65 FF	556	JMP MDN	
FA62:	08	557	RESET CLD ; DO THIS FIRST THIS TIME	*2
FA63:	20 84 FE	558	JSR SETNORM	
FA66:	20 2F FD	559	JSR INIT	
FA69:	20 93 FE	560	JSR SETVID	
FA6C:	20 89 FE	561	JSR SETKBD	
FA6F:	AD 58 C0	562	INITAN LDA SETANO ; AN0 = TTL HI	
FA72:	AD 5A C0	563	LDA SETAN1 ; AN1 = TTL HI	
FA75:	AD 5D C0	564	LDA CLRAN2 ; AN2 = TTL LO	
FA78:	AD 5F C0	565	LDA CLRAN3 ; AN3 = TTL LO	
FA7B:	AD FF CF	566	LDA CLRROM ; TURN OFF EXTNSN ROM	*3
FA7E:	2C 10 C0	567	BIT KBDSTRB ; CLEAR KEYBOARD	*4
FAB1:	08	568	NEWMON CLD	
FAB2:	20 3A FF	569	JSR BELL ; CAUSES DELAY IF KEY BOUNCES	*5
FAB5:	AD F3 03	570	LDA SOFTEV+1 ; IS RESET HI	*6
FAB8:	49 A5	571	EOR ##A5 ; A FUNNY COMPLEMENT OF THE	*7
FABA:	CD F4 03	572	CMP PWRDUP ; PWR UP BYTE ???	
FABD:	00 17	573	BNE PWRUP ; NO SO PWRUP	*8
FABF:	AD F2 03	574	LDA SOFTEV ; YES SEE IF COLD START	*9
FA92:	00 0F	575	BNE NOFIX ; HAS BEEN DONE YET?	
FA94:	A9 E0	576	LDA ##E0 ; ??	
FA96:	CD F3 03	577	CMP SOFTEV+1 ; ??	
FA99:	00 08	578	BNE NOFIX ; YES SO REENTER SYSTEM	*10
FA9B:	A0 03	579	FIXSEV LDY #3 ; NO SO POINT AT WARM START	*11

1. BRK-Verzweigungsadresse wird beim Initialisieren des DOS geändert.
2. RESET-Adresse beim ersten Mal.
3. Die Erweiterungs-ROMs abschalten.
4. Tastendruck löschen.
5. Verzögerung, falls eine Taste gedrückt wird.
6. RESET-Adresse : höherwertiges Byte.
7. Einschaltkontrolle mit dem POWERED UP-Byte.
8. Sprung zum Kaltstart, wenn das Gerät eben eingeschaltet wurde.
9. Wurde der Kaltstart schon durchgeführt!
10. Ja : Sprung zurück ins System.
11. Nein : Verzweigungsadresse für den Warmstart einrichten.

FA9D:	8C F2 03	580	STY SOFTEV ; FOR NEXT RESET	
FAA0:	4C 00 E0	581	JMP BASIC ; AND DO THE COLD START	*1
FAA3:	6C F2 03	582	NOFIX JMP (SOFTEV) ; SOFT ENTRY VECTOR	*2
FAA6:		583	*****	
FAA6:	20 60 FB	584	PWRUP JSR APPLEII	
FAA9:		585	SETPG3 EQU * ; SET PAGE 3 VECTORS	*3
FAA9:	A2 05	586	LDX #5	
FAAB:	BD FC FA	587	SETPLP LDA PWRCON-1,X ; WITH CNTRL B ADRS	*4
FAAE:	9D EF 03	588	STA BRKV-1,X ; OF CURRENT BASIC	
FAB1:	CA	589	DEX	
FAB2:	D0 F7	590	BNE SETPLP	
FAB4:	A9 C8	591	LDA #*CB ; LOAD HI SLOT +1	*5
FAB6:	86 00	592	STX LOCO ; SETPG3 MUST RETURN X=0	*6
FABB:	85 01	593	STA LOC1 ; SET PTR H	
FABA:	A0 07	594	SLOOP LDY #7 ; Y IS BYTE PTR	
FABC:	C6 01	595	DEC LOC1	
FABE:	A5 01	596	LDA LOC1	
FAC0:	C9 C0	597	CMP #*CO ; AT LAST SLOT YET?	*7
FAC2:	F0 D7	598	BEG FIXSEV ; YES AND IT CANT BE A DISK	*8
FAC4:	8D FB 07	599	STA MSLOT	
FAC7:	31 00	600	NXTBYT LDA (LOCO),Y ; FETCH A SLOT BYTE	*9
FAC9:	D9 01 FB	601	CMP DISKID-1,Y ; IS IT A DISK ??	*10
FACC:	D0 EC	602	BNE SLOOP ; NO SO NEXT SLOT DOWN	*11
FACE:	88	603	DEY	
FACF:	88	604	DEY ; YES SO CHECK NEXT BYTE	*12
FAD0:	10 F5	605	BPL NXTBYT ; UNTIL 4 CHECKED	
FAD2:	6C 00 00	606	JMP (LOCO)	
FAD5:	EA	607	NOP	
FAD6:	EA	608	NOP	
FAD7:		609	* REGDSP MUST DRG *FAD7	
FAD7:	20 BE FD	610	REGDSP JSR CROUT	
FADA:	A9 45	611	RGDSP1 LDA #*45	
FADC:	B5 40	612	STA A3L	
FADE:	A9 00	613	LDA #*00	
FAE0:	85 41	614	STA A3H	
FAE2:	A2 FB	615	LDX #*FB	
FAE4:	A9 A0	616	RDSP1 LDA #*A0	
FAE6:	20 ED FD	617	JSR COUT	
FAE9:	BD 1E FA	618	LDA RTBL-251,X	
FAEC:	20 ED FD	619	JSR COUT	
FAEF:	A9 BD	620	LDA #*BD	
FAF1:	20 ED FD	621	JSR COUT	
FAF4:		622	* LDA ACC+5,X	
FAF4:	B5 4A	623	DFB #B5,#4A	
FAF6:	20 DA FD	624	JSR PRBYTE	
FAF9:	EB	625	INX	
FAFA:	30 EB	626	BMI RDSP1	
FAFC:	60	627	RTS	
FAFD:	59 FA	628	PWRCON DW OLDBRK	
FAFF:	00 E0 45	629	DFB #00,#E0,#45	
FB02:	20 FF 00			
FB05:	FF	630	DISKID DFB #20,#FF,#00,#FF	
FB06:	03 FF 3C	631	DFB #03,#FF,#3C	
FB09:	C1 D0 D0	632	TITLE DFB #C1,#D0,#D0	
FB0C:	CC C5 A0	633	DFB #CC,#C5,#A0	
FB0F:	DD DB	634	DFB #DD,#DB	
FB11:		635	XLTLBL EQU *	
FB11:	C4 C2 C1	636	DFB #C4,#C2,#C1	
FB14:	FF C3	637	DFB #FF,#C3	
FB16:	FF FF FF	638	DFB #FF,#FF,#FF	
FB19:		639	* MUST DRG #FB19	
FB19:	C1 D8 D9	640	RTBL DFB #C1,#D8,#D9	
FB1C:	D0 D3	641	DFB #D0,#D3	
FB1E:	AD 70 C0	642	PREAD LDA PTRIG	
FB21:		643	LST ON	
FB21:	A0 00	644	LDY #*00	
FB23:	EA	645	NOP	
FB24:	EA	646	NOP	
FB25:	BD 64 C0	647	PREAD2 LDA PADDLO,X	
FB28:	10 04	648	BPL RTS2D	
FB2A:	CB	649	INY	
FB2B:	D0 FB	650	BNE PREAD2	
FB2D:	88	651	DEY	

1. BASIC-Kaltstart.
2. Sprung zurück in die aktuelle Sprache.
3. Setzt die Verzweigungsadressen auf Seite Drei.
4. Adresse des aktuellen BASIC für **CTRLB**.
5. Peripherie-Karten-Adresse + 1.
6. Hier ist $X = 0$.
7. Ist schon der letzte Einsteckschlitz geprüft worden?
8. Ja. Es kann keine Disk angeschlossen sein.
9. Holt ein Byte aus dem Peripherie-Anschluß.
10. Ist es eine Disk?
11. Nein : Den nächsten Anschluß prüfen.
12. Testet das nächste Byte, bis vier Bytes getestet sind.

FB2E:	60	652	RTS2D	RTS			
FB2F:	A9 00	2	INIT	LDA	**00		
FB31:	85 48	3		STA	STATUS		
FB33:	AD 56 C0	4		LDA	LORES		
FB36:	AD 54 C0	5		LDA	LOWSCR		
FB39:	AD 51 C0	6	SETTXT	LDA	TXTSET		
FB3C:	A9 00	7		LDA	**00		
FB3E:	F0 0D	8		BEG	SETWND		
FB40:	AD 50 C0	9	SETGR	LDA	TXTCLR		
FB43:	AD 53 C0	10		LDA	MIXSET		
FB46:	20 36 FB	11		JSR	CLRTOP		
FB49:	A9 14	12		LDA	**14		
FB4B:	85 22	13	SETWND	STA	WNDTOP		
FB4D:	A9 00	14		LDA	**00		
FB4F:	85 20	15		STA	WDLFT		
FB51:	A9 28	16		LDA	**28		
FB53:	85 21	17		STA	WNDWDTH		
FB55:	A9 18	16		LDA	**18		
FB57:	85 23	19		STA	WNDBTM		
FB59:	A9 17	20		LDA	**17		
FB5B:	85 25	21	TABV	STA	CV		
FB5D:	4C 22 FC	22		JMP	VTAB		
FB60:	20 58 FC	23	APPLEII	JSR	HOME	; CLEAR THE SCRN	*1
FB63:	A0 08	24		LDY	#8		
FB65:	B9 08 FB	25	STITLE	LDA	TITLE-1,Y ; GET A CHAR		*2
FB68:	99 0E 04	26		STA	LINE1+14,Y		
FB6B:	86	27		DEY			
FB6C:	DO F7	28		BNE	STITLE		
FB6E:	60	29		RTS			
FB6F:	AD F3 03	30	SETPWRC	LDA	SOFTEV+1		
FB72:	49 A5	31		EOR	**A5		
FB74:	8D F4 03	32		STA	PWREDUP		
FB77:	60	33		RTS			
FB7B:	34	34	VIDWAIT	EGU	*	; CHECK FOR A PAUSE	*3
FB7B:	C9 8D	35		CMP	**8D	; ONLY WHEN I HAVE A CR	
FB7A:	DO 18	36		BNE	NOWAIT	; NOT SO, DO REGULAR	*4
FB7C:	AC 00 C0	37		LDY	KBD	; IS KEY PRESSED?	*5
FB7F:	10 13	38		BPL	NOWAIT	; NO	
FB81:	C0 93	39		CPY	**93	; IS IT CTL B ?	*6
FB83:	DO 0F	40		BNE	NOWAIT	; NO SO IGNORE	*7
FB85:	2C 10 C0	41		BIT	KBDSTRB	; CLEAR STROBE	*8
FB8B:	AC 00 C0	42	KBDWAIT	LDY	KBD	; WAIT TILL NEXT KEY TO RESUME	*9
FB8B:	10 FB	43		BPL	KBDWAIT	; WAIT FOR KEYPRESS	
FB8D:	C0 B3	44		CPY	**B3	; IS IT CONTROL C ?	*10
FB8F:	F0 03	45		BEG	NOWAIT	; YES SO LEAVE IT	*11
FB91:	2C 10 C0	46		BIT	KBDSTRB	; CLR STROBE	*12
FB94:	4C FD FB	47	NOWAIT	JMP	VIDDUT	; DO AS BEFORE	*13
FB97:		48		PAGE			
FB97:	3B	49	ESCOLD	SEC		; INSURE CARRY SET	*14
FB9B:	4C 2C FC	50		JMP	ESC1		
FB9B:	AB	51	ESCNOW	TAY		; USE CHAR AS INDEX	*15
FB9C:	B9 4B FA	52		LDA	XLTLB-#C9,Y ; XLATE IJKM TO CBAD		
FB9F:	20 97 FB	53		JSR	ESCOLD	; DO THIS CURSOR MOTION	*16
FBA2:	20 0C FD	54		JSR	RDKEY	; AND GET NEXT	
FBA5:	C9 CE	55	ESCNEW	CMP	**CE	; IS THIS AN N ?	*17
FBA7:	DO EE	56		BCS	ESCOLD	; N OR GREATER DO IT	*18
FBA9:	C9 C9	57		CMP	**C9	; LESS THAN I ?	*19
FBAB:	90 EA	58		BCC	ESCOLD	; YES SO OLD WAY	*20
FBAD:	C9 CC	59		CMP	**CC	; IS IT A L ?	*21
FBAF:	F0 E6	60		BEG	ESCOLD	; DO NORMAL	*22
FBB1:	DO EB	61		BNE	ESCNOW	; GO DO IT	
FBB3:	EA	62		NOP			
FBB4:	EA	63		NOP			
FBB5:	EA	64		NOP			
FBB6:	EA	65		NOP			
FBB7:	EA	66		NOP			
FBB8:	EA	67		NOP			
FBB9:	EA	68		NOP			
FBBA:	EA	69		NOP			

1. Bildschirm löschen.
2. Ein Zeichen holen.
3. Soll ein CR (carriage return) ausgegeben werden?
4. Nein : normal fortfahren.
5. Ist eine Taste gedrückt?
6. War es **CTRL S**?
7. Nein : Andere Zeichen werden ignoriert.
8. Löscht den Tastendruck.
9. Auf den nächsten Tastendruck warten.
10. Ist es **CTRL C**?
11. Ja : gibt das **CTRL C** an das laufende Programm weiter.
12. Löscht den Tastendruck.
13. Das Programm arbeitet normal weiter.
14. Carry muß gesetzt sein.
15. Das Zeichen dient als Index.
16. Führt die Zeiger-Bewegung aus und holt den nächsten Tastendruck.
17. Ist es ein „N“?
18. „N“ oder größer : führt die Zeigerbewegung aus.
19. Kleiner als „I“?
20. Ja : führt es in der alten Weise aus.
21. Ist es ein „L“?
22. Ja : normale Ausführung.
23. Nein : Ausführung der neuen ESC-Einrichtungen.

FBBB:	EA	70		NOP		
FBBC:	EA	71		NOP		
FBBD:	EA	72		NOP		
FBBE:	EA	73		NOP		
FBBF:	EA	74		NOP		
FBC0:	EA	75		NOP		
FBC1:		76	*	MUST ORQ	*FBC1	
FBC1:	4B	77		BASCALC	PHA	
FBC2:	4A	78		LSR	A	
FBC3:	29 03	79		AND	**03	
FBC5:	09 04	80		DRA	**04	
FBC7:	85 29	81		STA	BASH	
FBC9:	68	82		PLA		
FBCA:	29 18	83		AND	**18	
FBCC:	90 02	84		BCC	BASCLC2	
FBCF:	69 7F	85		ADC	**7F	
FBD0:	85 28	86	BASCLC2	STA	BASL	
FBD2:	0A	87		ASL	A	
FBD3:	0A	88		ASL	A	
FBD4:	05 28	89		ORA	BASL	
FBD6:	85 28	90		STA	BASL	
FBD8:	60	91		RTS		
FBD9:	C9 87	92	BELL1	CMP	**87	
FBD8:	D0 12	93		BNE	RTS2B	
FBD9:	A9 40	94		LDA	**40	
FBD9:	20 A8 FC	95		JSR	WAIT	
FBE2:	A0 C0	96		LDY	**C0	
FBE4:	A9 0C	97	BELL2	LDA	**0C	
FBE6:	20 A8 FC	98		JSR	WAIT	
FBE9:	AD 30 C0	99		LDA	SPKR	
FBEC:	88	100		DEY		
FBED:	D0 F5	101		BNE	BELL2	
FBEE:	60	102	RTS2B	RTS		
FBF0:		103		PAGE		
FBF0:	A4 24	104	STORADV	LDY	CH	
FBF2:	91 28	105		STA	(BASL),Y	
FBF4:	E6 24	106	ADVANCE	INC	CH	
FBF6:	A5 24	107		LDA	CH	
FBF8:	C5 21	108		CMP	WNDWDTH	
FBFA:	B0 64	109		BCS	CR	
FBFC:	60	110	RTS3	RTS		
FBFD:	C9 A0	111	VIDOUT	CMP	**A0	
FBFF:	B0 EF	112		BCS	STORADV	
FC01:	A8	113		TAY		
FC02:	10 EC	114		BPL	STORADV	
FC04:	C9 BD	115		CMP	**BD	
FC06:	F0 5A	116		BEG	CR	
FC08:	C9 8A	117		CMP	**8A	
FC0A:	F0 5A	118		BEG	LF	
FC0C:	C9 88	119		CMP	**88	
FC0E:	D0 C9	120		BNE	BELL1	
FC10:	C6 24	121	BS	DEC	CH	
FC12:	10 E8	122		BPL	RTS3	
FC14:	A5 21	123		LDA	WNDWDTH	
FC16:	B5 24	124		STA	CH	
FC18:	C6 24	125		DEC	CH	
FC1A:	A5 22	126	UP	LDA	WNDTOP	
FC1C:	C5 25	127		CMP	CV	
FC1E:	B0 08	128		BCS	RTS4	
FC20:	C6 25	129		DEC	CV	
FC22:	A5 25	130	VTAB	LDA	CV	
FC24:	20 C1 FB	131	VTABZ	JSR	BASCALC	
FC27:	65 20	132		ADC	WNLFT	
FC29:	85 28	133		STA	BASL	
FC2B:	60	134	RTS4	RTS		
FC2C:	49 C0	135	ESC1	EOR	**C0 ; ESC @ ?	
FC2E:	F0 28	136		BEG	HOME ; IF SO DO HOME AND CLEAR	*1
FC30:	69 FD	137		ADC	**FD ; ESC-A OR B CHECK	*2
FC32:	90 C0	138		BCC	ADVANCE ; A, ADVANCE	
FC34:	F0 DA	139		BEG	BS ; B, BACKSPACE	
FC36:	69 FD	140		ADC	**FD ; ESC-C OR D CHECK	*3
FC38:	90 2C	141		BCC	LF ; C, DOWN	
FC3A:	F0 DE	142		BEG	UP ; D, GO UP	

1. ESC - : Bildschirm löschen und Zeiger nach links oben.
2. ESC-A- und ESC-B-Test : A : Zeiger nach rechts. B : Zeiger nach links.
3. ESC-C- und ESC-D-Test : C : Zeiger nach unten. D : Zeiger nach oben.

FC3C:	69	FD	143		ADC	##FD	; ESC-E OR F CKECK
FC3E:	90	5C	144		BCC	CLREOL	; E. CLEAR TO END OF LINE
FC40:	D0	E9	145		BNE	RTS4	; ELSE NOT F. RETURN
FC42:	A4	24	146	CLREOP	LDY	CH	; ESC F IS CLR TO END OF PAGE
FC44:	A3	25	147		LDA	CV	
FC46:	4B		148	CLEOP1	PHA		
FC47:	20	24	149		JSR	VTABZ	
FC4A:	20	9E	150		JSR	CLEOLZ	
FC4D:	A0	00	151		LDY	##00	
FC4F:	68		152		PLA		
FC50:	69	00	153		ADC	##00	
FC52:	C5	23	154		CMP	WNDBTM	
FC54:	90	F0	155		BCC	CLEOP1	
FC56:	B0	CA	156		BCS	VTAB	
FC58:	A5	22	157	HOME	LDA	WNDDTOP	
FC5A:	B5	25	158		STA	CV	
FC5C:	A0	00	159		LDY	##00	
FC5E:	B4	24	160		STY	CH	
FC60:	F0	E4	161		BEG	CLEOP1	
FC62:			162		PAGE		
FC62:	A9	00	163	CR	LDA	##00	
FC64:	B5	24	164		STA	CH	
FC66:	E6	25	165	LF	INC	CV	
FC68:	A5	25	166		LDA	CV	
FC6A:	C5	23	167		CMP	WNDBTM	
FC6C:	90	B6	168		BCC	VTABZ	
FC6E:	C6	25	169		DEC	CV	
FC70:	A5	22	170	SCROLL	LDA	WNDDTOP	
FC72:	4B		171		PHA		
FC73:	20	24	172		JSR	VTABZ	
FC76:	A5	28	173	SCRL1	LDA	BASL	
FC78:	B5	2A	174		STA	BAS2L	
FC7A:	A5	29	175		LDA	BASH	
FC7C:	B5	2B	176		STA	BAS2H	
FC7E:	A4	21	177		LDY	WNDDWTH	
FC80:	8B		178		DEY		
FC81:	68		179		PLA		
FC82:	69	01	180		ADC	##01	
FC84:	C5	23	181		CMP	WNDBTM	
FC86:	B0	0D	182		BCS	SCRL3	
FC88:	4B		183		PHA		
FC89:	20	24	184		JSR	VTABZ	
FC8C:	B1	2B	185	SCRL2	LDA	(BASL),Y	
FC8E:	91	2A	186		STA	(BAS2L),Y	
FC90:	8B		187		DEY		
FC91:	10	F9	188		BPL	SCRL2	
FC93:	30	E1	189		BMI	SCRL1	
FC95:	A0	00	190	SCRL3	LDY	##00	
FC97:	20	9E	191		JSR	CLEOLZ	
FC9A:	B0	86	192		BCS	VTAB	
FC9C:	A4	24	193	CLREOL	LDY	CH	
FC9E:	A9	A0	194	CLEOLZ	LDA	##A0	
FCA0:	91	2B	195	CLEOL2	STA	(BASL),Y	
FCA2:	CB		196		INY		
FCA3:	C4	21	197		CPY	WNDDWTH	
FCA5:	90	F9	198		BCC	CLEOL2	
FCA7:	60		199		RTS		
FCA8:	3B		200	WAIT	SEC		
FCA9:	4B		201	WAIT2	PHA		
FCAA:	E9	01	202	WAIT3	SBC	##01	
FCAC:	D0	FC	203		BNE	WAIT3	
FCAE:	6B		204		PLA		
FCAF:	E9	01	205		SBC	##01	
FCB1:	D0	F6	206		BNE	WAIT2	
FCB3:	60		207		RTS		
FCB4:	E6	42	208	NXTA4	INC	A4L	
FCB6:	D0	02	209		BNE	NXTA1	
FCBB:	E6	43	210		INC	A4H	
FCBA:	A5	3C	211	NXTA1	LDA	A1L	
FCBC:	C5	3E	212		CMP	A2L	
FCBE:	A5	3D	213		LDA	A1H	
FCC0:	E5	3F	214		SBC	A2H	
FCC2:	E6	3C	215		INC	A1L	

1. ESC-E- und ESC-F-Test : E : Rest der Zeile löschen.
F : Rest des Bildes löschen.

FCC4:	D0 02	216	BNE RTS4B
FCC6:	E6 3D	217	INC A1H
FCCB:	60	218	RTS4B RTS
FCC9:		219	PAGE
FCC9:	A0 4B	220	LDY ##4B
FCCB:	20 DB FC	221	JSR ZERDLY
FCCE:	D0 F9	222	BNE HEADR
FCD0:	69 FE	223	ADC ##FE
FCD2:	80 F5	224	BDS HEADR
FCD4:	A0 21	225	LDY ##21
FCD6:	20 DB FC	226	WRBIT JSR ZERDLY
FCD9:	CB	227	INY
FCDA:	CB	228	INY
FCDB:	BB	229	ZERDLY DEY
FCDC:	D0 FD	230	BNE ZERDLY
FCDE:	90 05	231	BCC WRTAPE
FCE0:	A0 32	232	LDY ##32
FCE2:	8B	233	ONEDLY DEY
FCE3:	D0 FD	234	BNE ONEDLY
FCE5:	AC 20 CO	235	WRTAPE LDY TAPEOUT
FCEB:	A0 2C	236	LDY ##2C
FCEA:	CA	237	DEX
FCEB:	60	238	RTS
FCEC:	A2 08	239	RDBYTE LDX ##0B
FCEE:	4B	240	RDBYT2 PHA
FCEF:	20 FA FC	241	JSR RD2BIT
FCF2:	6B	242	PLA
FCF3:	2A	243	ROL A
FCF4:	A0 3A	244	LDY ##3A
FCF6:	CA	245	DEX
FCF7:	D0 F5	246	BNE RDBYT2
FCF9:	60	247	RTS
FCFA:	20 FD FC	248	RD2BIT JSR RDBIT
FCFD:	8B	249	RDBIT DEY
FCFE:	AD 60 CO	250	LDA TAPEIN
FD01:	45 2F	251	EOR LASTIN
FD03:	10 FB	252	BPL RDBIT
FD05:	45 2F	253	EOR LASTIN
FD07:	85 2F	254	STA LASTIN
FD09:	C0 80	255	CPY ##80
FD0B:	60	256	RTS
FD0C:	A4 24	257	RDKEY LDY CH
FD0E:	B1 2B	258	LDA (BASL), Y
FD10:	4B	259	PHA
FD11:	29 3F	260	AND ##3F
FD13:	09 40	261	ORA ##40
FD15:	91 2B	262	STA (BASL), Y
FD17:	6B	263	PLA
FD18:	6C 3B 00	264	JMP (KSWL)
FD1B:	E6 4E	265	KEYIN INC RNDL
FD1D:	D0 02	266	BNE KEYIN2
FD1F:	E6 4F	267	INC RNDH
FD21:	2C 00 CO	268	KEYIN2 BIT KBD ; READ KEYBOARD
FD24:	10 F5	269	BPL KEYIN
FD26:	91 2B	270	STA (BASL), Y
FD28:	AD 00 CO	271	LDA KBD
FD2B:	2C 10 CO	272	BIT KBDSTRB
FD2E:	60	273	RTS
FD2F:	20 0C FD	274	ESC JSR RDKEY
FD32:	20 A5 FB	275	JSR ESCNEW
FD35:	20 0C FD	276	RDCHAR JSR RDKEY
FD3B:	C9 9B	277	CMP ##9B
FD3A:	F0 F3	278	BEG ESC
FD3C:	60	279	RTS
FD3D:		280	PAGE
FD3D:	A5 32	281	NOTCR LDA INVFLG
FD3F:	4B	282	PHA
FD40:	A9 FF	283	LDA ##FF
FD42:	85 32	294	STA INVFLG
FD44:	BD 00 02	285	LDA IN, X
FD47:	20 ED FD	286	JSR CDUT
FD4A:	6B	287	PLA
FD4B:	85 32	288	STA INVFLG

1. Tastenfeld lesen.

FD4D:	BD 00 02	289	LDA IN, X
FD50:	C9 BB	290	CMP ##88
FD52:	F0 1D	291	BEG BCKSPC
FD54:	C9 98	292	CMP ##98
FD56:	F0 0A	293	BEG CANCEL
FD58:	E0 FB	294	CPX ##FB
FD5A:	90 03	295	BCC NOTCR1
FD5C:	20 3A FF	296	JSR BELL
FD5F:	E8	297	NOTCR1 INX
FD60:	D0 13	298	BNE NXTCHAR
FD62:	A9 DC	299	CANCEL LDA ##DC
FD64:	20 ED FD	300	JSR COUT
FD67:	20 BE FD	301	GETLNZ JSR CROUT
FD6A:	A5 33	302	GETLN LDA PROMPT
FD6C:	20 ED FD	303	JSR COUT
FD6F:	A2 01	304	LDX ##01
FD71:	8A	305	BCKSPC TXA
FD72:	F0 F3	306	BEG GETLNZ
FD74:	CA	307	DEX
FD75:	20 35 FD	308	NXTCHAR JSR RDCHAR
FD78:	C9 95	309	CMP ##95
FD7A:	D0 02	310	BNE CAPTST
FD7C:	B1 28	311	LDA (BASL), Y
FD7E:	C9 E0	312	CAPTST CMP ##E0
FD80:	90 02	313	BCC ADDINP
FD82:	29 DF	314	AND ##DF ; SHIFT TO UPPER CASE
FD84:	9D 00 02	315	ADDINP STA IN, X
FD87:	C9 8D	316	CMP ##8D
FD89:	D0 B2	317	BNE NOTCR
FD8B:	20 9C FC	318	JSR CLREOL
FD8E:	A9 BD	319	CROUT LDA ##8D
FD90:	D0 58	320	BNE COUT
FD92:	A4 3D	321	PRA1 LDY A1H
FD94:	A6 3C	322	LDX A1L
FD96:	20 BE FD	323	PRYX2 JSR CRDUT
FD99:	20 40 F9	324	JSR PRNTYX
FD9C:	A0 00	325	LDY ##00
FD9E:	A9 AD	326	LDA ##AD
FDA0:	4C ED FD	327	JMP COUT
FDA3:		328	PAGE
FDA3:	A5 3C	329	XAMB LDA A1L
FDA5:	09 07	330	DRA ##07
FDA7:	85 3E	331	STA A2L
FDA9:	A5 3D	332	LDA A1H
FDAB:	85 3F	333	STA A2H
FDAD:	A5 3C	334	MODBCHK LDA A1L
FDAF:	29 07	335	AND ##07
FDB1:	D0 03	336	BNE DATAOUT
FDB3:	20 92 FD	337	XAM JSR PRA1
FDB6:	A9 A0	338	DATAOUT LDA ##A0
FDB8:	20 ED FD	339	JSR COUT
FDBB:	B1 3C	340	LDA (A1L), Y
FDBD:	20 DA FD	341	JSR PRBYTE
FDC0:	20 BA FC	342	JSR NXTA1
FDC3:	90 EB	343	BCC MODBCHK
FDC5:	60	344	RTS4C RTS
FDC6:	4A	345	XAMPM LSR A
FDC7:	90 EA	346	BCC XAM
FDC9:	4A	347	LSR A
FDCA:	4A	348	LSR A
FDCB:	A5 3E	349	LDA A2L
FDCD:	90 02	350	BCC ADD
FDCF:	49 FF	351	EOR ##FF
FDD1:	65 3C	352	ADD ADC A1L
FDD3:	48	353	PHA
FDD4:	A9 BD	354	LDA ##8D
FDD6:	20 ED FD	355	JSR COUT
FDD9:	68	356	PLA
FDDA:	48	357	PRBYTE PHA
FDDB:	4A	358	LSR A
FDDC:	4A	359	LSR A
FDDD:	4A	360	LSR A
FDDE:	4A	361	LSR A

*1

1. Kleinbuchstaben werden zu Großbuchstaben.

FDDF:	20 E5 FD	362	JSR PRHEXZ
FDE2:	68	363	PLA
FDE3:	29 OF	364 PRHEX	AND ##0F
FDE5:	09 B0	365 PRHEXZ	ORA ##80
FDE7:	C9 BA	366	CMP ##BA
FDE9:	90 02	367	BCC COUT
FDEB:	69 06	368	ADC ##06
FDED:	6C 36 00	369 COUT	JMP (CSWL)
FDF0:	C9 A0	370 COUT1	CMP ##A0
FDF2:	90 02	371	BCC COUTZ
FDF4:	25 32	372	AND INVFLG
FDF6:	84 35	373 COUTZ	STY YSAV1
FDF8:	48	374	PHA
FDF9:	20 78 FB	375	JSR VIDWAIT ; GO CHECK FOR PAUSE
FDFC:	68	376	PLA
FDFD:	A4 35	377	LDY YSAV1
FDFE:	60	378	RTS
FE00:		379	PAGE
FE00:	C6 34	380 BL1	DEC YSAV
FE02:	F0 9F	381	BEG XAMB
FE04:	CA	382 BLANK	DEX
FE05:	D0 16	383	BNE SETMDZ
FE07:	C9 BA	384	CMP ##BA
FE09:	D0 BB	385	BNE XAMPM
FE0B:	85 31	386 STOR	STA MODE
FE0D:	A5 3E	387	LDA A2L
FE0F:	91 40	388	STA (A3L),Y
FE11:	E6 40	389	INC A3L
FE13:	D0 02	390	BNE RTS5
FE15:	E6 41	391	INC A3H
FE17:	60	392 RTS5	RTS
FE18:	A4 34	393 SETMODE	LDY YSAV
FE1A:	B9 FF 01	394	LDA IN-1,Y
FE1D:	85 31	395 SETMDZ	STA MODE
FE1F:	60	396	RTS
FE20:	A2 01	397 LT	LDX ##01
FE22:	B5 3E	398 LT2	LDA A2L,X
FE24:	93 42	399	STA A4L,X
FE26:	93 44	400	STA A5L,X
FE28:	CA	401	DEX
FE29:	10 F7	402	3PL LT2
FE2B:	60	403	RTS
FE2C:	31 3C	404 MOVE	LDA (A1L),Y
FE2E:	91 42	405	STA (A4L),Y
FE30:	20 B4 FC	406	JSR NXTA4
FE33:	90 F7	407	BCC MOVE
FE35:	60	408	RTS
FE36:	B1 3C	409 VFY	LDA (A1L),Y
FE38:	D1 42	410	CMP (A4L),Y
FE3A:	F0 1C	411	BEG VFYOK
FE3C:	20 92 FD	412	JSR PRA1
FE3F:	B1 3C	413	LDA (A1L),Y
FE41:	20 DA FD	414	JSR PRBYTE
FE44:	A9 A0	415	LDA ##A0
FE46:	20 ED FD	416	JSR COUT
FE49:	A9 A8	417	LDA ##A8
FE4B:	20 ED FD	418	JSR COUT
FE4E:	B1 42	419	LDA (A4L),Y
FE50:	20 DA FD	420	JSR PRBYTE
FE53:	A9 A9	421	LDA ##A9
FE55:	20 ED FD	422	JSR COUT
FE58:	20 B4 FC	423 VFYOK	JSR NXTA4
FE5B:	90 D9	424	BCC VFY
FE5D:	60	425	RTS
FE5E:	20 73 FE	426 LIST	JSR A1PC
FE61:	A9 14	427	LDA ##14
FE63:	48	428 LIST2	PHA
FE64:	20 D0 FB	429	JSR INSTDSP
FE67:	20 53 F9	430	JSR PCADJ
FE6A:	83 3A	431	STA PCL
FE6C:	84 3B	432	STY PCH
FE6E:	68	433	PLA
FE6F:	38	434	SEC

*1

1. Test auf Ausgabestop durch **CTRL S**

FE70:	E9 01	435	SBC	##01
FE72:	D0 EF	436	BNE	LIST2
FE74:	60	437	RTS	
FE75:		438	PAGE	
FE75:	8A	439	A1PC	TXA
FE76:	F0 07	440	BEG	A1PCRTS
FE78:	B5 3C	441	A1PCLP	LDA A1L, X
FE7A:	95 3A	442	STA	PCL, X
FE7C:	CA	443	DEX	
FE7D:	10 F9	444	BPL	A1PCLP
FE7F:	60	445	A1PCRTS	RTS
FE80:	A0 3F	446	SETINV	LDY ##3F
FE82:	D0 02	447	BNE	SETIFLG
FE84:	A0 FF	448	SETNORM	LDY ##FF
FE86:	84 32	449	SETIFLG	STY INVFLG
FE88:	60	450	RTS	
FE89:	A9 00	451	SETKBD	LDA ##00
FE8B:	B5 3E	452	INPORT	STA A2L
FE8D:	A2 3B	453	INPRT	LDX ##KSWL
FE8F:	A0 1B	454	LDY	##KEYIN
FE91:	D0 08	455	BNE	IOPRT
FE93:	A9 00	456	SETVID	LDA ##00
FE95:	B5 3E	457	OUTPORT	STA A2L
FE97:	A2 36	458	OUTPRT	LDX ##CSWL
FE99:	A0 F0	459	LDY	##COUT1
FE9B:	A5 3E	460	IOPRT	LDA A2L
FE9D:	29 0F	461	AND	##0F
FE9F:	F0 06	462	BEG	IOPRT1
FEA1:	09 C0	463	ORA	##IADR/256
FEA3:	A0 00	464	LDY	##00
FEA5:	F0 02	465	BEG	IOPRT2
FEA7:	A9 FD	466	IOPRT1	LDA ##COUT1/256
FEA9:		467	IOPRT2	EGU *
FEA9:	94 00	468	STY	LOCO, X ; ##94, ##00
FEAB:	95 01	469	STA	LOC1, X ; ##95, ##01
FEAD:	60	470	RTS	
FEAE:	EA	471	NOP	
FEAF:	EA	472	NOP	
FEBO:	4C 00 E0	473	XBASIC	JMP BASIC
FEB3:	4C 03 E0	474	BASCONT	JMP BASIC2
FEB6:	20 75 FE	475	GD	JSR A1PC
FEB9:	20 3F FF	476	JSR	RESTORE
FEBC:	6C 3A 00	477	JMP	(PCL)
FEBF:	4C D7 FA	478	REGZ	JMP REGDSP
FEC2:	60	479	TRACE	RTS
FEC3:		480	* TRACE	IS GONE
FEC3:	EA	481	NOP	
FEC4:	60	482	STEPZ	RTS ; STEP IS GONE
FEC5:	EA	483	NOP	
FEC6:	EA	484	NOP	
FEC7:	EA	485	NOP	
FEC8:	EA	486	NOP	
FEC9:	EA	487	NOP	
FECA:	4C FB 03	488	USR	JMP USRADR
FECD:		489	PAGE	
FECD:	A9 40	490	WRITE	LDA ##40
FECF:	20 C9 FC	491	JSR	HEADR
FED2:	A0 27	492	LDY	##27
FED4:	A2 00	493	WR1	LDX ##00
FED6:	41 3C	494	EDR	(A1L, X)
FED8:	48	495	PHA	
FED9:	A1 3C	496	LDA	(A1L, X)
FEDB:	20 ED FE	497	JSR	WRBYTE
FEDE:	20 BA FC	498	JSR	NXTA1
FEE1:	A0 1D	499	LDY	##1D
FEE3:	68	500	PLA	
FEE4:	90 EE	501	BCC	WR1
FEE6:	A0 22	502	LDY	##22
FEEB:	20 ED FE	503	JSR	WRBYTE
FEEB:	F0 4D	504	BEG	BELL
FEED:	A2 10	505	WRBYTE	LDX ##10
FEEF:	0A	506	WRBYT2	ASL A
FEFO:	20 D6 FC	507	JSR	WRBIT

1. Die Fehlerhilfe „STEP“ ist nicht mehr verfügbar.

FEF3:	D0 FA	508	BNE	WRBYT2
FEF5:	60	509	RTS	
FEF6:	20 00 FE	510	JSR	BL1
FEF9:	68	511	PLA	
FEFA:	68	512	PLA	
FEFB:	D0 6C	513	BNE	MONZ
FEFD:	20 FA FC	514	READ JSR	RD2BIT
FF00:	A9 16	515	LDA	##16
FF02:	20 C9 FC	516	JSR	HEADR
FF05:	B5 2E	517	STA	CHKSUM
FF07:	20 FA FC	518	JSR	RD2BIT
FF0A:	A0 24	519	RD2 LDY	##24
FF0C:	20 FD FC	520	JSR	RDBIT
FF0F:	B0 F9	521	BCS	RD2
FF11:	20 FD FC	522	JSR	RDBIT
FF14:	A0 3B	523	LDY	##3B
FF16:	20 EC FC	524	RD3 JSR	RDBYTE
FF19:	B1 3C	525	STA	(A1L, X)
FF1B:	45 2E	526	EOR	CHKSUM
FF1D:	B5 2E	527	STA	CHKSUM
FF1F:	20 BA FC	528	JSR	NXTA1
FF22:	A0 35	529	LDY	##35
FF24:	90 F0	530	BCC	RD3
FF26:	20 EC FC	531	JSR	RDBYTE
FF29:	C5 2E	532	CMP	CHKSUM
FF2B:	F0 0D	533	BEG	BELL
FF2D:	A9 C5	534	PRERR LDA	##C5
FF2F:	20 ED FD	535	JSR	COUT
FF32:	A9 D2	536	LDA	##D2
FF34:	20 ED FD	537	JSR	COUT
FF37:	20 ED FD	538	JSR	COUT
FF3A:	A9 87	539	BELL LDA	##87
FF3C:	4C ED FD	540	JMP	COUT
FF3F:		541	PAGE	
FF3F:	A5 48	542	RESTORE LDA	STATUS
FF41:	48	543	PHA	
FF42:	A5 45	544	LDA	A5H
FF44:	A6 46	545	RESTR1 LDX	XREG
FF46:	A4 47	546	LDY	YREG
FF48:	28	547	PLP	
FF49:	60	548	RTS	
FF4A:	B5 45	549	SAVE STA	A5H
FF4C:	B6 46	550	SAV1 STX	XREG
FF4E:	B4 47	551	STY	YREG
FF50:	08	552	PHP	
FF51:	68	553	PLA	
FF52:	B5 48	554	STA	STATUS
FF54:	BA	555	TSX	
FF55:	B6 49	556	STX	SPNT
FF57:	D8	557	CLD	
FF58:	60	558	RTS	
FF59:	20 84 FE	559	OLDRST JSR	SETNORM
FF5C:	20 2F FB	560	JSR	INIT
FF5F:	20 93 FE	561	JSR	SETVID
FF62:	20 89 FE	562	JSR	SETKBD
FF65:		563	PAGE	
FF65:	D8	564	MON CLD	
FF66:	20 3A FF	565	JSR	BELL
FF69:	A9 AA	566	MONZ LDA	##AA
FF6B:	B5 33	567	STA	PROMPT
FF6D:	20 67 FD	568	JSR	GETLNZ
FF70:	20 C7 FF	569	JSR	ZMODE
FF73:	20 A7 FF	570	NXTITM JSR	GETNUM
FF76:	B4 34	571	STY	YSAV
FF7B:	A0 17	572	LDY	##17
FF7A:	BB	573	CHRSRCH DEY	
FF7B:	30 EB	574	BMI	MON
FF7D:	D9 CC FF	575	CMP	CHRTBL, Y
FF80:	D0 FB	576	BNE	CHRSRCH
FF82:	20 BE FF	577	JSR	TOSUB
FF85:	A4 34	578	LDY	YSAV
FF87:	4C 73 FF	579	JMP	NXTITM
FF8A:	A2 03	580	DIG LDX	##03

FF8C:	0A	581		ASL A	
FF8D:	0A	582		ASL A	
FF8E:	0A	583		ASL A	
FF8F:	0A	584		ASL A	
FF90:	0A	585	NXTBIT	ASL A	
FF91:	26 3E	586		ROL A2L	
FF93:	26 3F	587		ROL A2H	
FF95:	CA	588		DEX	
FF96:	10 FB	589		BPL NXTBIT	
FF98:	A5 31	590	NXTBAS	LDA MODE	
FF9A:	D0 06	591		BNE NXTBS2	
FF9C:		592	*		
FF9C:	B5 3F	593		LDA A2H, X	
FF9E:		594	*		
FF9E:	95 3D	595		STA A1H, X	
FFA0:		596	*		
FFA0:	95 41	597		STA A3H, X	
FFA2:	EB	598	NXTBS2	INX	
FFA3:	F0 F3	599		BEG NXTBAS	
FFA5:	D0 06	600		BNE NXTCHR	
FFA7:	A2 00	601	GETNUM	LDX ##00	
FFA9:	B6 3E	602		STX A2L	
FFAB:	B6 3F	603		STX A2H	
FFAD:	B9 00 02	604	NXTCHR	LDA IN, Y	
FFB0:	CB	605		INY	
FFB1:	49 B0	606		EOR ##B0	
FFB3:	C9 0A	607		CMP ##0A	
FFB5:	70 D3	608		BCC DIQ	
FFB7:	69 8B	609		ADC ##8B	
FFB9:	C9 FA	610		CMP ##FA	
FFBB:	B0 CD	611		BCS DIQ	
FFBD:	60	612		RTS	
FFBE:	A9 FE	613	TOSUB	LDA #GD/256	
FFC0:	48	614		PHA	
FFC1:	B9 E3 FF	615		LDA SUBTBL, Y	
FFC4:	48	616		PHA	
FFC5:	A5 31	617		LDA MODE	
FFC7:	A0 00	618	ZMODE	LDY ##00	
FFC9:	B4 31	619		STY MODE	
FFCB:	60	620		RTS	
FFCC:		621		PAGE	
FFCC:	BC	622	CHRTBL	DFB \$BC	
FFCD:	B2	623		DFB \$B2	
FFCE:	BE	624		DFB \$BE	
FFCF:	B2	625		DFB \$B2	T CMD NOW LIKE USR
FFD0:	EF	626		DFB \$EF	
FFD1:	C4	627		DFB \$C4	
FFD2:	B2	628		DFB \$B2	S CMD NOW LIKE USR
FFD3:	A9	629		DFB \$A9	
FFD4:	BB	630		DFB \$BB	
FFD5:	A6	631		DFB \$A6	
FFD6:	A4	632		DFB \$A4	
FFD7:	06	633		DFB \$06	
FFD8:	95	634		DFB \$95	
FFD9:	07	635		DFB \$07	
FFDA:	02	636		DFB \$02	
FFDB:	05	637		DFB \$05	
FFDC:	F0	638		DFB \$F0	
FFDD:	00	639		DFB \$00	
FFDE:	EB	640		DFB \$EB	
FFDF:	93	641		DFB \$93	
FFE0:	A7	642		DFB \$A7	
FFE1:	C6	643		DFB \$C6	
FFE2:	99	644		DFB \$99	
FFE3:	B2	645	SUBTBL	DFB \$B2	
FFE4:	C9	646		DFB \$C9	
FFE5:	BE	647		DFB \$BE	
FFE6:	C1	648		DFB \$C1	
FFE7:	35	649		DFB \$35	
FFE8:	BC	650		DFB \$BC	
FFE9:	C4	651		DFB \$C4	
FFEA:	96	652		DFB \$96	
FFEB:	AF	653		DFB \$AF	

1. T und S arbeiten jetzt wie USR.

FFEC:	17	654	DFB	*17
FFED:	17	655	DFB	*17
FFEE:	2B	656	DFB	*2B
FFEF:	1F	657	DFB	*1F
FFF0:	83	658	DFB	*83
FFF1:	7F	659	DFB	*7F
FFF2:	5D	660	DFB	*5D
FFF3:	CC	661	DFB	*CC
FFF4:	B5	662	DFB	*B5
FFF5:	FC	663	DFB	*FC
FFF6:	17	664	DFB	*17
FFF7:	17	665	DFB	*17
FFF8:	F5	666	DFB	*F5
FFF9:	03	667	DFB	*03
FFFA:	FB 03	668	DW	NMI
FFFC:	62 FA	669	DW	RESET
FFFE:	40 FA	670	DW	IRG

ENDASM

MONITOR ROM

```
1 *****
2 *
3 *           APPLE II
4 *     SYSTEM MONITOR
5 *
6 *     COPYRIGHT 1977 BY
7 *     APPLE COMPUTER, INC.
8 *
9 *     ALL RIGHTS RESERVED
10 *
11 *           S. WOZNIAK
12 *           A. BAUM
13 *
14 *****
15
16          TITLE
17 LOCJ      EPZ $00
18 LOCL      EPZ $01
19 WNDLFT    EPZ $20
20 WNDWDTH   EPZ $21
21 WNDTOP    EPZ $22
22 WNDBTM    EPZ $23
23 CH        EPZ $24
24 CV        EPZ $25
25 GBASL     EPZ $26
26 GBASH     EPZ $27
27 BASL      EPZ $28
28 BASH      EPZ $29
29 BAS2L     EPZ $2A
30 BAS2H     EPZ $2B
31 H2        EPZ $2C
32 LMNEM     EPZ $2C
33 RTNL      EPZ $2C
34 V2        EPZ $2D
35 RMNEM     EPZ $2D
36 RTNH      EPZ $2D
37 MASK      EPZ $2E
38 CHKSUM    EPZ $2E
39 FORMAT    EPZ $2E
40 LASTIN    EPZ $2F
41 LENGTH    EPZ $2F
42 SIGN      EPZ $2F
43 COLOR     EPZ $30
44 MODE      EPZ $31
45 INVFLG    EPZ $32
46 PROMPT    EPZ $33
47 YSAV      EPZ $34
48 YSAV1     EPZ $35
49 CSWL      EPZ $36
50 CSWH      EPZ $37
51 KSWL      EPZ $38
52 KSWH      EPZ $39
53 PCL       EPZ $3A
54 PCH       EPZ $3B
55 XQT       EPZ $3C
56 A1L       EPZ $3C
57 A1H       EPZ $3D
58 A2L       EPZ $3E
59 A2H       EPZ $3F
60 A3L       EPZ $40
61 A3H       EPZ $41
62 A4L       EPZ $42
63 A4H       EPZ $43
64 A5L       EPZ $44
65 A5H       EPZ $45
66 ACC       EPZ $45
67 XREG      EPZ $46
68 YREG      EPZ $47
69 STATUS    EPZ $48
```

	69	SPNT	EPZ	\$49				
	70	RNDL	EPZ	\$4E				
	71	RNDH	EPZ	\$4F				
	72	ACL	EPZ	\$50				
	73	ACH	EPZ	\$51				
	74	XTNDL	EPZ	\$52				
	75	XTNDH	EPZ	\$53				
	76	AUXL	EPZ	\$54				
	77	AUXH	EPZ	\$55				
	78	PICK	EPZ	\$95				
	79	IN	EQU	\$0200				
	80	USRADR	EQU	\$03F8				
	81	NMI	EQU	\$03FB				
	82	IRQLOC	EQU	\$03FE				
	83	IOADR	EQU	\$C000				
	84	KBD	EQU	\$C000				
	85	KBDS TRB	EQU	\$C010				
	86	TAPEOUT	EQU	\$C020				
	87	SPKR	EQU	\$C030				
	88	TXTCLR	EQU	\$C050				
	89	TXTSET	EQU	\$C051				
	90	MIXCLR	EQU	\$C052				
	91	MIXSET	EQU	\$C053				
	92	LOWSCR	EQU	\$C054				
	93	HISCR	EQU	\$C055				
	94	LORES	EQU	\$C056				
	95	HIRES	EQU	\$C057				
	96	TAPE IN	EQU	\$C060				
	97	PADDLO	EQU	\$C064				
	98	PTRIG	EQU	\$C070				
	99	BASIC	EQU	\$E000				
	100	BASIC2	EQU	\$E003				
	101		ORG	\$F800	ROM START ADDRESS			
F800:	4A	102	PL0T	LSR	A	Y-COORD/2		*1
F801:	08	103		PHP		SAVE LSB IN CARRY		*2
F802:	20 47 F8	104		JSR	GBASCALC	CALC BASE ADR IN GBASL,H		*3
F805:	28	105		PLP		RESTORE LSB FROM CARRY		*4
F806:	A9 0F	106		LDA	#\$0F	MASK \$0F IF EVEN		*5
F808:	90 02	107		BCC	RTMASK			
F80A:	69 E0	108		ADC	#\$E0	MASK \$F0 IF ODD		*6
F80C:	85 2E	109	RTMASK	STA	MASK			
F80E:	B1 26	110	PL0T1	LDA	(GBASL),Y	DATA		
F810:	45 30	111		EOR	COLOR	XOR COLOR		
F812:	25 2E	112		AND	MASK	AND MASK		
F814:	51 26	113		EOR	(GBASL),Y	XOR DATA		
F816:	91 26	114		STA	(GBASL),Y	TO DATA		*7
F818:	60	115		RTS				
F819:	20 00 F8	116	HLINE	JSR	PL0T	PL0T SQUARE		*8
F81C:	C4 2C	117	HLINE1	CPY	H2	DONE?		*9
F81E:	B0 11	118		BCS	RTS1	YES, RETURN		*10
F820:	C8	119		INY		NO, INCR INDEX (X-COORD)		*11
F821:	20 0E F8	120		JSR	PL0T1	PL0T NEXT SQUARE		
F824:	90 F6	121		BCC	HLINE1	ALWAYS TAKEN		*12
F826:	69 01	122	VLINEZ	ADC	#\$01	NEXT Y-COORD		*13
F828:	48	123	VLINE	PHA		SAVE ON STACK		*14
F829:	20 00 F8	124		JSR	PL0T	PL0T SQUARE		
F82C:	68	125		PLA				
F82D:	C5 2D	126		CMP	V2	DONE?		*15
F82F:	90 F5	127		BCC	VLINEZ	NO, LOOP.		*16
F831:	60	128	RTS1	RTS				
F832:	A0 2F	129	CLRSCR	LDY	#\$2F	MAX Y, FULL SCR N CLR		*17
F834:	D0 02	130		BNE	CLRSC2	ALWAYS TAKEN		*18
F836:	A0 27	131	CLRTOP	LDY	#\$27	MAX Y, TOP SCR N CLR		*19
F838:	84 2D	132	CLRSC2	STY	V2	STORE AS BOTTOM COORD		*20
		133	*			FOR VLINE CALLS		
F83A:	A0 27	134		LDY	#\$27	RIGHTMOST X-COORD (COLUMN)		*21
F83C:	A9 00	135	CLRSC3	LDA	#\$0	TOP COORD FOR VLINE CALLS		
F83E:	85 30	136		STA	COLOR	CLEAR COLOR (BLACK)		
F840:	20 28 F8	137		JSR	VLINE	DRAW VLINE		
F843:	88	138		DEY		NEXT LEFTMOST X-COORD		*22
F844:	10 F6	139		BPL	CLRSC3	LOOP UNTIL DONE.		*23
F846:	60	140		RTS				
F847:	48	141	GBASCALC	PHA		FOR INPUT 00DEF GH		*24
F848:	4A	142		LSR	A			

1. Senkrechte Koordinate/2. Das Gerade-/Ungerade-Bit erscheint im Carry und bedeutet den oberen/unteren Block eines auszugebenden Bytes.
2. Carry mit dem gesamten Prozessor-Status retten.
3. Basisadresse ausrechnen und in GBASL, GBASH bereithalten.
4. Gerade/Ungerade erscheint wieder im Carry.
5. Maske für „Gerade“ = oberer Block.
6. Maske für „Ungerade“ = unterer Block : $\$0F + \$E0 + 1$ (weil $C = 1$) = $\$F0$.
7. Durch die Maske besteht die eine Hälfte des Bytes aus Nullen und wird durch das XOR (ausschließendes ODER) wieder zu der alten Farbe gelangen. Die andere Hälfte, die die Maske überstanden hat, erlebt ein zweites XOR mit der alten Farbe und erhält damit die gewünschte Farbe.
8. Farbigen Block plotten.
9. Aktuelle waagrechte Position mit der rechten Grenze vergleichen.
10. Fertig, wenn Grenze erreicht.
11. Sonst ein Feld nach rechts gehen.
12. Erzwungener Sprung : Seit $\$F81C$ ist $C = 0$.
13. Eine Zeile tiefer gehen.
14. Y-Koordinate während des Plottens im Keller zwischenspeichern.
15. Aktuelle Position mit der Untergrenze vergleichen.
16. Sprung zurück, falls die Grenze noch nicht erreicht wurde.
17. $Y = 47$, um den gesamten Bildschirm zu löschen.
18. Erzwungener Sprung.
19. $Y = 39$ zum Löschen des oberen Teils des Bildes.
20. Y wird in die untere Grenze gegeben.
21. Vorbereiten auf VLINE-Aufrufe : $Y = 39$ (rechte Spalte), $A = 0$ (oberste Zeile) und COLOR = Schwarz.
22. Nächste Spalte links.
23. Sprung zurück, bis das Bild gelöscht ist.
24. Eingabeformat : 000DEFGH.

F849:	29	03	143		AND	#S03				
F84B:	09	04	144		ORA	#S04	GENERATE GBASH=000001FG			*1
F84D:	85	27	145		STA	GBASH				
F84F:	68		146		PLA		AND GBASL=HDEDE000			*2
F850:	29	18	147		AND	#S18				
F852:	90	02	148		BCC	GBCALC				
F854:	69	7F	149		ADC	#S7F				
F856:	85	26	150	GBCALC	STA	GBASL				
F858:	0A		151		ASL	A				
F859:	0A		152		ASL	A				
F85A:	05	26	153		ORA	GBASL				
F85C:	85	26	154		STA	GBASL				
F85E:	60		155		RTS					
F85F:	A5	30	156	NXTCOL	LDA	COLOR	INCREMENT COLOR BY 3			*3
F861:	18		157		CLC					
F862:	69	03	158		ADC	#S03				
F864:	29	0F	159	SETCOL	AND	#S0F	SETS COLOR=17*A MOD 16			*4
F866:	85	30	160		STA	COLOR				
F868:	0A		161		ASL	A	BOTH HALF BYTES OF COLOR EQUAL			*5
F869:	0A		162		ASL	A				
F86A:	0A		163		ASL	A				
F86B:	0A		164		ASL	A				
F86C:	05	30	165		ORA	CCOLOR				
F86E:	85	30	166		STA	COLOR				
F870:	60		167		RTS					
F871:	4A		168	SCRN	LSR	A	READ SCREEN Y-COORD/2			*6
F872:	08		169		PHP		SAVE LSB (CARRY)			*7
F873:	20	47	F8	170	JSR	GBASCALC	CALC BASE ADDRESS			*8
F876:	B1	26	171		LDA	(GBASL),Y	GET BYTE			*9
F878:	28		172		PLP		RESTORE LSB FROM CARRY			*10
F879:	90	04	173	SCRN2	BCC	RTMSKZ	IF EVEN, USE LO H			*11
F87B:	4A		174		LSR	A				
F87C:	4A		175		LSR	A				
F87D:	4A		176		LSR	A	SHIFT HIGH HALF BYTE DOWN			*12
F87E:	4A		177		LSR	A				
F87F:	29	0F	178	RTMSKZ	AND	#S0F	MASK 4-BITS			*13
F881:	60		179		RTS					
F882:	A6	3A	180	INSDS1	LDX	PCL	PRINT PCL,H			*14
F884:	A4	3B	181		LDY	PCH				
F886:	20	96	FD	182	JSR	PRYX2				
F889:	20	48	F9	183	JSR	PRBLNK	FOLLOWED BY A BLANK			*15
F88C:	A1	3A	184		LDA	(PCL,X)	GET OP CODE			*16
F88E:	A6		185	INSDS2	TAY					
F88F:	4A		186		LSR	A	EVEN/ODD TEST			*17
F890:	90	09	187		BCC	IEVEN				
F892:	6A		188		ROR	A	BIT 1 TEST			*18
F893:	80	10	189		BCS	ERR	XXXXXX11 INVALID OP			*19
F895:	C9	A2	190		CMP	#SA2				
F897:	F0	0C	191		BEQ	ERR	OPCODE \$89 INVALID			*20
F899:	29	87	192		AND	#S87	MASK BITS			*21
F89B:	4A		193	IEVEN	LSR	A	LSB INTO CARRY FOR L/R TEST			*22
F89C:	AA		194		TAX					
F89D:	BD	62	F9	195	LDA	FMT1,X	GET FORMAT INDEX BYTE			*23
F8A0:	20	79	F8	196	JSR	SCRN2	R/L H-BYTE ON CARRY			*24
F8A3:	D0	04	197		BNE	GETFMT				
F8A5:	A0	80	198	ERR	LDY	#S80	SUBSTITUTE \$80 FOR INVALID OPS			*25
F8A7:	A9	00	199		LDA	#S0	SET PRINT FORMAT INDEX TO 0			*26
F8A9:	AA		200	GETFMT	TAX					
F8AA:	ED	A6	F9	201	LDA	FMT2,X	INDEX INTO PRINT FORMAT TABLE			*27
F8AD:	85	2E	202		STA	FORMAT	SAVE FOR ADR FIELD FORMATTING			
F8AF:	29	03	203		AND	#S03	MASK FOR 2-BIT LENGTH			*28
			204	*		(P=1 BYTE, 1=2 BYTE, 2=3 BYTE)				
F8B1:	85	2F	205		STA	LENGTH				*29
F8B3:	98		206		TYA		OPCODE			*30
F8B4:	29	8F	207		AND	#S8F	MASK FOR 1XXXX1010 TEST			*31
F8B6:	AA		208		TAX		SAVE IT			
F8B7:	98		209		TYA		OPCODE TO A AGAIN			*32
F8B8:	A0	03	210		LDY	#S03				
F8BA:	E0	8A	211		CPX	#S8A				*33
F8BC:	F0	0B	212		BEQ	MNNDX3				
F8BE:	4A		213	MNNDX1	LSR	A				
F8BF:	90	08	214		BCC	MNNDX3	FORM INDEX INTO MNEMONIC TABLE			*34
F8C1:	4A		215		LSR	A				

1. Format für das höherwertige Byte : 000001FG.
2. Format für das niederwertige Byte : HDEE000.
3. Die Nummer der Farbe wird um 3 erhöht.
4. Farbe 16 wird zu Farbe 0, Farbe 17 wird zu Farbe 1 und Farbe 18 zu 2.
5. Beide Byte-Hälften bekommen diesen Farbwert.
6. Y-Koordinate halbieren und das Gerade-/Ungerade-Bit in den Carry bringen.
7. Carry retten.
8. Basisadresse ermitteln.
9. Byte holen.
10. Gerade/Ungerade-Bit aus dem Keller holen.
11. Sprung, falls Gerade.
12. Die linken vier Bit erscheinen rechts : ABCDXXXX → 0001ABCD.
13. Maske : 0001ABCD → 0000ABCD.
14. Befehlszähler PCL, PCH ausgeben.
15. Und dann 3 Blanks : X wird 0.
16. Hole Befehlscode.
17. Gerade/Ungerade-Test.
18. Bit 1 erscheint im Carry.
19. Test auf ungültigen Befehlscode : XXXXXX11.
20. Befehlscode \$89 ist auch ungültig.
21. Maske : 10000111.
22. Bit 0 in den Carry zum Links/Rechts-Test.
23. Format-Index holen.
24. Je nach Carry erscheint rechts das linke oder das rechte Halb-Byte (LLLLRRRR : C = 0 : 0000RRRR, C = 1 : 0000LLLL).
25. Ungültige Befehlscodes durch \$80 ersetzen.
26. Index für das Ausgabeformat auf 0 setzen.
27. Ausgabeformat aus der Tabelle FMT2 holen und speichern.
28. 2-Bit-Maske : 00000011 (0 = 1 Byte, 1 = 2 Byte, 2 = 3 Byte).
29. Befehlslänge (Anzahl der Bytes - 1) speichern.
30. Befehlscode steht seit \$F88E im Y-Register.
31. Maske für 1XX1010-Test anwenden und Ergebnis ins X-Register retten.
32. Nochmals den Befehlscode aus dem Y-Register holen.
33. 1XX1010-Test.
34. Index für die Befehlssymboltabelle bilden.

F8C2:	4A	216	MNNDX2	LSR	A	1) 1XXX1G10=>00101XXX	
F8C3:	09 20	217		ORA	#\$20	2) XXXYYY01=>00111XXX	
F8C5:	88	218		DEY		3) XXXYYY1G=>00110XXX	
F8C6:	DU FA	219		BNE	MNNDX2	4) XXXYYY100=>00100XXX	
F8C8:	C8	220		INY		5) XXXXX000=>000XXXXX	
F8C9:	88	221	MNNDX3	DEY			
F8CA:	DU F2	222		BNE	MNNDX1		
F8CC:	60	223		RTS			
F8CD:	FF FF FF	224		DFB	\$\$\$FF, \$\$\$FF, \$\$\$FF		
F8D0:	20 82 F8	225	INSTDSP	JSR	INSDS1	GEN FMT, LEN BYTES	*1
F8D3:	48	226		PHA		SAVE MNEMONIC TABLE INDEX	*2
F8D4:	B1 3A	227	PRNTOP	LDA	(PCL), Y		
F8D6:	20 DA FD	228		JSR	PRBYTE		
F8D9:	A2 01	229		LDX	#\$01	PRINT 2 BLANKS	*3
F8DB:	20 4A F9	230	PRNTBL	JSR	PRBL2		
F8DE:	C4 2F	231		CPY	LENGTH	PRINT INST (1-3 BYTES) IN A 12 CHR FIELD	*4
F8E0:	C8	232		INY			
F8E1:	90 F1	233		BCC	PRNTOP		
F8E3:	A2 03	234		LDX	#\$03	CHAR COUNT FOR MNEMONIC PRINT	*5
F8E5:	C0 04	235		CPY	#\$04		
F8E7:	90 F2	236		BCC	PRNTBL		
F8E9:	68	237		PLA		RECOVER MNEMONIC INDEX	*6
F8EA:	A8	238		TAY			
F8EB:	B9 C0 F9	239		LDA	MNEM, Y		
F8EE:	85 2C	240		STA	LMNEM	FETCH 3-CHAR MNEMONIC (PACKED IN 2-BYTES)	*7
F8F0:	B9 00 FA	241		LDA	MNEM, Y		
F8F3:	85 2D	242		STA	RMNEM		
F8F5:	A9 00	243	PRMN1	LDA	#\$00		
F8F7:	A0 05	244		LDY	#\$05		
F8F9:	06 2D	245	PRMN2	ASL	RMNEM	SHIFT 5 BITS OF CHARACTER INIC A (CLEARS CARRY)	*8
F8FB:	26 2C	246		ROL	LMNEM		
F8FD:	2A	247		ROL	A		
F8FE:	88	248		DEY			
F8FF:	DU F8	249		BNE	PRMN2		
F901:	69 BF	250		ADC	#\$BF	ADD "?" OFFSET	*9
F903:	20 ED FD	251		JSR	COUT	OUTPUT A CHAR OF MNEM	*10
F906:	CA	252		DBX			
F907:	DU EC	253		BNE	PRMN1		
F909:	20 48 F9	254		JSR	PRBLNK	OUTPUT 3 BLANKS	*11
F90C:	A4 2F	255		LDY	LENGTH		
F90E:	A2 06	256		LDX	#\$06	CNT FOR 6 FORMAT BITS	*12
F910:	EU 03	257	PRADR1	CPX	#\$03		
F912:	F0 1C	258		BEQ	PRADR5	IF X=3 THEN ADDR.	*13
F914:	06 2E	259	PRADR2	ASL	FCRMT		
F916:	90 UE	260		BCC	PRADR3		
F918:	BD B3 F5	261		LDA	CHAR1-1, X		
F91B:	20 ED FD	262		JSR	COUT		
F91E:	BD B9 F9	263		LDA	CHAR2-1, X		
F921:	F0 03	264		BEQ	PRADR3		
F923:	20 ED FD	265		JSR	COUT		
F926:	CA	266	PRADR3	DEX			
F927:	DU E7	267		BNE	PRADR1		
F929:	60	268		RTS			
F92A:	88	269	PRADR4	DEY			
F92B:	30 E7	270		BMI	PRADR2		
F92D:	20 DA FD	271		JSR	PRBYTE		
F930:	A5 2E	272	PRADR5	LDA	FORMAT		
F932:	C9 E8	273		CMP	#\$E8	HANDLE REL ADR MODE SPECIAL (PRINT TARGET, NOT OFFSET)	*14
F934:	B1 3A	274		LDA	(PCL), Y		
F936:	90 F2	275		BCC	PRADR4		
F938:	20 56 F9	276	RELADR	JSR	PCADJ3		
F93B:	AA	277		TAX			
F93C:	E8	278		INX		PCL, PCH+OFFSET+1 TO A, Y	*15
F93D:	DU 01	279		BNE	PRNTYX	+1 TO Y, X	
F93F:	C8	280		INY			
F940:	98	281	PRNTYX	TYA			
F941:	20 DA FD	282	PRNTAX	JSR	PRBYTE	OUTPUT TARGET ADR OF BRANCH AND RETURN	*16
F944:	8A	283	PRNTX	TXA			
F945:	4C DA FD	284		JMP	PRBYTE		
F948:	A2 03	285	PRBLNK	LDX	#\$03	BLANK COUNT	*17
F94A:	A9 A0	286	PRBL2	LDA	#\$A0	LOAD A SPACE	
F94C:	20 ED FD	287	PRBL3	JSR	COUT	OUTPUT A BLANK	*18
F94F:	CA	288		DEX			

1. Befehlszähler ausgeben, Format erzeugen und die Befehlslänge bestimmen.
2. Index für die Befehlssymboltabelle retten.
3. Ein Leerzeichen ausgeben.
4. Befehl in einem Feld von 12 Zeichen ausgeben.
5. Zähler für die Zeichen des Befehlssymbols.
6. Index für die Befehlssymboltabelle aus dem Keller holen.
7. Befehlssymbol holen : 3 Zeichen aus zwei Bytes.
8. 5 Bit aus diesen 2 Bytes in den Akkumulator schieben : löscht Carry.
9. „?“ addieren, damit aus den 5 Bit ein druckreifer Buchstabe wird.
10. Ein Zeichen des Befehlssymbols ausgeben.
11. Drei Leerzeichen ausgeben.
12. Zähler für 6 Format-Bits.
13. Ausgabe der Adresse bei $X = 3$.
14. Besondere Behandlung des relativen Adreßformats : Ausgabe der Zieladresse.
15. Bestimmung der Zieladresse.
16. Ausgabe der Zieladresse und Rücksprung.
17. Zähler für die Leerzeichen.
18. Ausgabe eines Leerzeichens.

F950:	DU F8	289		BNE	PRBL2	LOOP UNTIL COUNT=0	*1
F952:	60	290		RTS			
F953:	38	291	PCADJ	SEC		0=1-BYTE,1=2-BYTE,	
F954:	A5 2F	292	PCADJ2	LDA	LENGTH	2=3-BYTE	
F956:	A4 3B	293	PCADJ3	LDY	PCH		
F958:	AA	294		TAX		TEST DISPLACEMENT SIGN	*2
F959:	10 01	295		BPL	PCADJ4	(FOR REL BRANCH)	
F95B:	08	296		DEY		EXTEND NEG BY DECR PCH	*3
F95C:	65 3A	297	PCADJ4	ADC	PCL		
F95E:	90 01	298		BCC	RTS2	PCL+LENGTH(OR DISPL)+1 TO A	
F960:	C6	299		INY		CARRY INTO Y (PCH)	
F961:	60	300	RTS2	RTS			
		301	*	FMT1	BYTES:	XXXXXXXXY INSTRS	*4
		302	*	IF	Y=0	THEN LEFT HALF BYTE	
		303	*	IF	Y=1	THEN RIGHT HALF BYTE	
		304	*			(X=INDEX)	
F962:	04 20 54						
F965:	30 UD	305	FMT1	DFB	\$04,\$20,\$54,\$		
F967:	00 04 90			DFB	\$80,\$04,\$90,\$		
F96A:	03 22	306		DFB	\$54,\$33,\$0D,\$		
F96C:	54 33 UD			DFB	\$90,\$04,\$20,\$		
F96F:	80 04	307		DFB	\$0D,\$80,\$04,\$		
F971:	90 04 20			DFB	\$20,\$54,\$3B,\$		
F974:	54 33	308		DFB	\$04,\$90,\$00,\$		
F976:	UD 80 04			DFB	\$33,\$0D,\$C8,\$		
F979:	90 04	309		DFB	\$11,\$22,\$44,\$		
F97B:	20 54 3B			DFB	\$C8,\$44,\$A9,\$		
F97E:	0C 80	310		DFB	\$44,\$33,\$0D,\$		
F980:	04 90 00			DFB	\$90,\$01,\$22,\$		
F983:	22 44	311		DFB	\$0D,\$80,\$04,\$		
F985:	33 UD C8			DFB	\$26,\$31,\$87,\$ZZZXXXY01 INSTR'S		
F988:	44 00	312		DFB	\$00	ERR	
F98A:	11 22 44		FMT2	DFB	\$21	INM	
F98D:	33 UD	313		DFB	\$81	Z-PAGE	
F98F:	C8 44 A9			DFB	\$82	ABS	
F992:	01 22	314		DFB	\$00	IMPLIED	
F994:	44 33 UD			DFB	\$00	ACCUMULATOR	
F997:	80 04	315		DFB	\$59	(ZPAG,X)	
F999:	90 01 22			DFB	\$4D	(ZPAG),Y	
F99C:	44 33	316		DFB	\$91	ZPAG,X	
F99E:	0D 80 04			DFB	\$92	ASS,X	
F9A1:	90	317		DFB	\$86	ABS,Y	
F9A2:	26 31 87			DFB	\$4A	(ABS)	
F9A5:	9A	318		DFB	\$85	ZPAG,Y	
F9A6:	00	319		DFB	\$9D	RELATIVE	
F9A7:	21	320		ASC	"," , # (\$"		
F9A8:	81	321					
F9A9:	82	322					
F9AA:	00	323					
F9AB:	00	324					
F9AC:	59	325					
F9AD:	4D	326					
F9AE:	91	327					
F9AF:	92	328					
F9B0:	86	329					
F9B1:	4A	330					
F9B2:	85	331					
F9B3:	9D	332					
F9B4:	AC A9 AC						
F9B7:	A3 Ab A4	333	CHAR1	DFB	\$D9,\$00,\$D8,\$		
F9BA:	D9 00 D8			*CHAR2:	"Y",0,"X\$\$",0		
F9BD:	A4 A4 00	334	CHAR2	DFB	\$09,\$00,\$D8,\$		
		335	*CHAR2:	"Y",0,"X\$\$",0			
		336	*	MNEML	IS OF FORM:		
		337	*	(A)	XXXXX000		
		338	*	(B)	XXXXY100		
		339	*	(C)	1XXX1010		
		340	*	(D)	XXXXYY10		
		341	*	(E)	XXXXYY01		
		342	*		(X=INDEX)		
F9C0:	1C 8A 1C						
F9C3:	23 5D 8B	343	MNEML	DFB	\$1C,\$8A,\$1C,\$		
F9C6:	1B A1 9D						

1. Sprung zurück und weiteres Leerzeichen ausgeben, bis der Zähler auf 0 ist.
2. Vorzeichen der Verlagerung für Relativsprünge ermitteln.
3. Bei negativem Vorzeichen wird PCH (der höherwertige Befehlszähler) um 1 vermindert.
4. Befehle der Form XXXXXY0 : die linken vier Bit, falls $Y = 0$, die rechten vier Bit, falls $Y = 1$.

F9C9:	8A 1D 23 344		DFB	\$1B,\$A1,\$9D,\$	
F9CC:	9D 8B 1D		DFB	\$9D,\$8B,\$1D,\$	
F9CF:	A1 00 29 345		DFB	\$19,\$AE,\$69,\$	
F9D2:	19 AE 69		DFB	\$24,\$53,\$1B,\$	
F9D5:	A8 19 23 346		DFB	\$19,\$A1	(A) FORMAT ABOVE
F9D8:	24 53 1B		DFB	\$00,\$1A,\$5B,\$	
F9DB:	23 24 53 347		DFB	\$24,\$24	(B) FORMAT
F9DE:	19 A1 348		DFB	\$AE,\$AE,\$A8,\$	
F9E0:	00 1A 5B		DFB	\$7C,\$00	(C) FORMAT
F9E3:	5B A5 69 349		DFB	\$15,\$9C,\$6D,\$	
F9E6:	24 24 350		DFB	\$29,\$53	(D) FORMAT
F9E8:	AE AE A8		DFB	\$84,\$13,\$34,\$	
F9EB:	AD 29 00 351		DFB	\$23,\$A0	(E) FORMAT
F9EE:	7C 00 352		DFB	\$D8,\$62,\$5A,\$	
F9F0:	15 9C 6D		DFB	\$94,\$88,\$54,\$	
F9F3:	9C A5 69 353		DFB	\$68,\$44,\$E6,\$	
F9F6:	29 53 354		DFB	\$08,\$84,\$74,\$	
F9F8:	84 13 34		DFB	\$74,\$F4,\$CC,\$	
F9FB:	11 A5 69 355		DFB	\$A4,\$8A	(A) FORMAT
F9FE:	23 A0 356		DFB	\$00,\$AA,\$A2,\$	
FA00:	D8 62 5A		DFB	\$74,\$72	(B) FORMAT
FA03:	48 26 62 357	MNEMR	DFB	\$44,\$68,\$B2,\$	
FA06:	94 88 54		DFB	\$22,\$00	(C) FORMAT
FA09:	44 C8 54 358		DFB	\$1A,\$1A,\$26,\$	
FA0C:	68 44 E8		DFB	\$88,\$C8	(D) FORMAT
FA0F:	94 00 B4 359		DFB	\$C4,\$CA,\$26,\$	
FA12:	08 84 74		DFB	\$A2,\$C8	(E) FORMAT
FA15:	B4 28 6E 360		DFB	\$FF,\$FF,\$FF	
FA18:	74 F4 CC		DFB	\$00,\$F4,\$26,\$	
FA1B:	4A 72 F2 361		DFB	\$88,\$C8	(D) FORMAT
FA1E:	A4 8A 362		DFB	\$C4,\$CA,\$26,\$	
FA20:	00 AA A2		DFB	\$A2,\$C8	(E) FORMAT
FA23:	A2 74 74 363		DFB	\$FF,\$FF,\$FF	
FA26:	74 72 364		DFB	\$00,\$F4,\$26,\$	
FA28:	44 68 B2		DFB	\$88,\$C8	(D) FORMAT
FA2B:	32 B2 00 365		DFB	\$C4,\$CA,\$26,\$	
FA2E:	22 00 366		DFB	\$A2,\$C8	(E) FORMAT
FA30:	1A 1A 26		DFB	\$FF,\$FF,\$FF	
FA33:	26 72 72 367		DFB	\$00,\$F4,\$26,\$	
FA36:	88 C8 368		DFB	\$88,\$C8	(D) FORMAT
FA38:	C4 CA 26		DFB	\$C4,\$CA,\$26,\$	
FA3B:	48 44 44 369		DFB	\$A2,\$C8	(E) FORMAT
FA3E:	A2 C8 370		DFB	\$FF,\$FF,\$FF	
FA40:	FF FF FF 371		DFB	\$00,\$F4,\$26,\$	
FA43:	20 DU F8 372	STEP	JSR	INSTDSP	DISASSEMBLE ONE INST *1
FA46:	68 373		PLA		AT (PCL,H)
FA47:	85 2C 374		STA	RTNL	ADJUST TO USER *2
FA49:	68 375		PLA		STACK. SAVE
FA4A:	85 2D 376		STA	RTNH	RTN ADR.
FA4C:	A2 08 377		LDX	#508	
FA4E:	BD 10 FB 378	XQINIT	LDA	INITBL-1,X	INIT XEQ AREA *3
FA51:	95 3C 379		STA	XQT,X	
FA53:	CA 380		DEX		
FA54:	D0 F8 381		BNE	XQINIT	
FA56:	A1 3A 382		LDA	(PCL,X)	USER OPCODE BYTE *4
FA58:	F0 42 383		BEQ	XBRK	SPECIAL IF BREAK *5
FA5A:	A4 2F 384		LDY	LENGTH	LEN FROM DISASSEMBLY *6
FA5C:	C9 20 385		CMP	#520	
FA5E:	F0 59 386		BEQ	XJSR	HANDLE JSR, RTS, JMP, *7
FA60:	C9 60 387		CMP	#560	JMP (), RTI SPECIAL
FA62:	F0 45 388		BEQ	XRTS	
FA64:	C9 4C 389		CMP	#54C	
FA66:	F0 5C 390		BEQ	XJMP	
FA68:	C9 6C 391		CMP	#56C	
FA6A:	F0 59 392		BEQ	XJMPAT	
FA6C:	C9 40 393		CMP	#540	
FA6E:	F0 35 394		BEQ	XRTI	
FA70:	29 1F 395		AND	#51F	
FA72:	49 14 396		ENR	#514	
FA74:	C9 04 397		CMP	#504	COPY USER INST TO XEQ AREA *8
FA76:	F0 02 398		BEQ	XQ2	WITH TRAILING NOPS
FA78:	B1 3A 399	XQ1	LDA	(PCL),Y	CHANGE REL BRANCH *9
FA7A:	99 3C 00 400	XQ2	STA	XQTNZ,Y	DISP TO 4 FOR

1. Einen Befehl disassembeln (an der Stelle PCL, PCH).
2. Rücksprungadresse retten.
3. „XEQ“-Speicherstellen initialisieren.
4. Hole Befehlscode des Benutzerprogramms.
5. Besondere Behandlung des BRK-Befehls.
6. Die Länge wurde schon beim Disassembeln ermittelt.
7. Sonderbehandlung für JSR, RTS, JMP, JMP (), RTI.
8. Den Befehl in den „XEQ“-Bereich eintragen (nach den Befehl mit NOP's auffüllen).
9. Branch-Befehle im XEQ nach BRANCH oder NBRNCH springen lassen.

FA7D:	8d	401		DEY		JMP TO BRANCH CR	
FA7E:	10 F8	402		BPL	XQ1	NBRANCH FROM REQ.	
FA80:	20 3F FF	403		JSR	RESTORE	RESTORE USER REG CONTENTS.	*1
FA83:	4C 3C 00	404		JMP	XQTNZ	REQ USER CP FROM RAM	*2
FA86:	85 45	405	IRQ	STA	ACC	(RETURN TO NBRANCH)	
FA88:	68	406		PLA			
FA89:	48	407		PHA		**IRQ HANDLER	*3
FA8A:	0A	408		ASL	A		
FA8B:	0A	409		ASL	A		
FA8C:	0A	410		ASL	A		
FA8D:	30 03	411		BMI	BREAK	TEST FOR BREAK	*4
FA8F:	6C FE 03	412		JMP	(IRQLOC)	USER ROUTINE VECTOR IN RAM	*5
FA92:	28	413	BREAK	PLP			
FA93:	20 4C FF	414		JSR	SAV1	SAVE REG'S ON BREAK	*6
FA96:	68	415		PLA		INCLUDING PC	
FA97:	85 3A	416		STA	PCL		
FA99:	68	417		PLA			
FA9A:	85 3B	418		STA	PCH		
FA9C:	20 82 F8	419	XBRK	JSR	INSDS1	PRINT USER PC.	*7
FA9F:	20 DA FA	420		JSR	RGDSP1	AND REG'S	
FAA2:	4C 65 FF	421		JMP	MON	GO TO MONITOR	*8
FAA5:	18	422	XRTI	CLC			
FAA6:	68	423		PLA		SIMULATE RTI BY EXPECTING	*9
FAA7:	85 48	424		STA	STATUS	STATUS FROM STACK, THEN RTS	
FAA9:	68	425	XRTS	PLA		RTS SIMULATION	*10
FAAA:	65 3A	426		STA	PCL	EXTRACT PC FROM STACK	
FAAC:	68	427		PLA		AND UPDATE PC BY 1 (LEN=0)	
FAAD:	65 3B	428	PCINC2	STA	PCH		
FAAF:	A5 2F	429	PCINC3	LDA	LENGTH	UPDATE PC BY LEN	*11
FAB1:	20 56 Fy	430		JSR	PCADJ3		
FAB4:	84 3B	431		STY	PCH		
FAB6:	18	432		CLC			
FAB7:	90 14	433		BCC	NEWPCL		
FAB9:	18	434	XJSR	CLC			
FABA:	20 54 F9	435		JSR	PCADJ2	UPDATE PC AND PUSH	*12
FABD:	AA	436		TAX		ONTO STACK FOR	
FABE:	98	437		TYA		JSR SIMULATE	
FABF:	48	438		PHA			
FAC0:	8A	439		TXA			
FAC1:	48	440		PHA			
FAC2:	A0 02	441		LDY	#\$02		
FAC4:	18	442	XJMP	CLC			
FAC5:	B1 3A	443	XJMPAT	LDA	(PCL),Y		
FAC7:	AA	444		TAX			
FAC8:	88	445		DEY		LOAD PC FOR JMP,	*13
FAC9:	B1 3A	446		LDA	(PCL),Y	(JMP) SIMULATE.	
FACB:	86 3B	447		STX	PCH		
FACD:	85 3A	448	NEWPCL	STA	PCL		
FACF:	80 F3	449		BCS	XJMP		
FAD1:	A5 2D	450	RTNJMP	LDA	RTNH		
FAD3:	48	451		PHA			
FAD4:	A5 2C	452		LDA	RTNL		
FAD6:	48	453		PHA			
FAD7:	20 8E FD	454	REGDSP	JSR	CROUT	DISPLAY USER REG	*14
FADA:	A9 45	455	RGDSP1	LDA	#\$ACC	CONTENTS WITH	
FADC:	85 40	456		STA	A3L	LABELS	
FADE:	A9 00	457		LDA	#\$ACC/256		
FAE0:	85 41	458		STA	A3H		
FAE2:	A2 FB	459		LDX	#\$FB		
FAE4:	A9 A0	460	RDSP1	LDA	#\$A0		
FAE6:	20 ED FD	461		JSR	COUT		
FAE9:	BD 1E FA	462		LDA	RTBL-\$FB,X		
FAEC:	20 ED FD	463		JSR	COUT		
FAEF:	A9 BD	464		LDA	#\$BD		
FAF1:	20 ED FD	465		JSR	COUT		
FAF4:	B5 4A	466		LDA	ACC+S,X		
FAF6:	20 DA FD	467		JSR	PRBYTE		
FAF9:	E8	468		INX			
FAFA:	30 E8	469		BMI	RDSP1		
FAFC:	60	470		RTS			
FAFD:	18	471	BRANCH	CLC		BRANCH TAKEN,	*15
FAFE:	A0 01	472		LDY	#\$01	ADD LEN+2 TO PC	
FBO0:	B1 3A	473		LDA	(PCL),Y		

1. Registerinhalte des Benutzerprogramms wiederherstellen.
2. Befehl im XEQ-Bereich ausführen. (XEQ = execute : ausführen).
3. IRQ-Behandlung.
4. BREAK-Bit im Status-Register testen.
5. IRQ-Adresse des Benutzerprogramms.
6. Registerinhalte und Befehlszähler retten.
7. Befehlszähler und Registerinhalte des Benutzerprogramms ausgeben.
8. Sprung zum Monitor.
9. RTI simulieren : holt den Status aus dem Keller und simuliert RTS.
10. RTS simulieren : Befehlszähler aus dem Keller holen und um 1 korrigieren.
11. PC (Befehlszähler) um LENGTH (Befehlslänge) korrigieren.
12. JSR simulieren : Befehlszähler auf den neuesten Stand bringen und in den Keller bringen.
13. Befehlszähler aus dem Keller holen zur Simulation von JMP und (JMP).
14. Registerinhalte mit den Bezeichnungen ausgeben.
15. Sprung wurde ausgeführt : addiert Länge = 2 zum Befehlszähler.

FB02:	20	56	F9	474		JSR	PCADJ3			
FB05:	85	3A		475		STA	PCL			
FB07:	98			476		TYA				
FB08:	38			477		SEC				
FB09:	80	A2		478		BCS	PCINC2			
FB0B:	20	4A	FF	479	NBRNCH	JSR	SAVE		NORMAL RETURN AFTER	*1
FB0E:	38			480		SEC			REQ USER OF	
FB0F:	80	9E		481		BCS	PCINC3		GO UPDATE FC	*2
FB11:	EA			482	INITBL	NOP				
FB12:	EA			483		NOP			DUMMY FILL FOR	*3
FB13:	4C	0B	FB	484		JMP	NBRNCH		REQ AREA	
FB16:	4C	FD	FA	485		JMP	BRANCH			
FB19:	C1			486	RTBL	DFB	SC1			
FB1A:	D8			487		DFB	SD8			
FB1B:	D9			488		DFB	SD9			
FB1C:	D0			489		DFB	SD0			
FB1D:	D3			490		DFB	SD3			
FB1E:	AD	70	CU	491	PREAD	LDA	PTRIG		TRIGGER PADDLES	*4
FB21:	A0	00		492		LDY	#\$00		INIT COUNT	*5
FB23:	EA			493		NOP			COMPENSATE FOR 1ST COUNT	*6
FB24:	EA			494		NOP				
FB25:	BD	64	CU	495	PREAD2	LDA	PADDL0,X		COUNT Y-REG EVERY	*7
FB28:	10	04		496		BPL	RTS2D		12 USEC	
FB2A:	C8			497		INY				
FB2B:	D0	F8		498		BNE	PREAD2		EXIT AT 255 MAX	*8
FB2D:	88			499		DEY				
FB2E:	60			500	RTS2D	RTS				
FB2F:	A9	00		501	INIT	LDA	#\$00		CLR STATUS FOR DEBUG	*9
FB31:	85	48		502		STA	STATUS		SOFTWARE	
FB33:	AD	56	CU	503		LDA	LORES			
FB36:	AD	54	C0	504		LDA	LOWSCR		INIT VIDEO MODE	*10
FB39:	AD	51	C0	505	SETTXT	LDA	TXTSET		SET FOR TEXT MODE	*11
FB3C:	A9	00		506		LDA	#\$00		FULL SCREEN WINDOW	
FB3E:	F0	0B		507		BEQ	SETWND			
FB40:	AD	50	CU	508	SETGR	LDA	TXTCLR		SET FOR GRAPHICS MODE	*12
FB43:	AD	53	CU	509		LDA	MIXSET		LOWER 4 LINES AS	
FB46:	20	36	F8	510		JSR	CLRTOP		TEXT WINDOW	
FB49:	A9	14		511		LDA	#\$14			
FB4B:	85	22		512	SETWND	STA	WNDTOP		SET FOR 40 COL WINDOW	*13
FB4D:	A9	00		513		LDA	#\$00		TOP IN A-REG,	
FB4F:	85	20		514		STA	WNDLFT		BTM AT LINE 24	
FB51:	A9	28		515		LDA	#\$28			
FB53:	85	21		516		STA	WNDWDTH			
FB55:	A9	18		517		LDA	#\$18			
FB57:	85	23		518		STA	WNBDM		VTAB TO ROW 23	
FB59:	A9	17		519		LDA	#\$17			
FB5B:	85	25		520	TABV	STA	CV		VTABS TO ROW IN A-REG	*14
FB5D:	4C	22	FC	521		JMP	VTAB			
FB60:	20	A4	FB	522	MULPM	JSR	MD1		ABS VAL OF AC AUX	*15
FB63:	A0	10		523	MUL	LDY	#\$10		INDEX FOR 16 BITS	*16
FB65:	A5	50		524	MUL2	LDA	ACL		ACX * AUX + XTND	
FB67:	4A			525		LSR	A		TO AC, XTND	
FB68:	90	0C		526		BCC	MUL4		IF NO CARRY,	*17
FB6A:	18			527		CLC			NO PARTIAL PROD.	
FB6B:	A2	FE		528		LDX	#\$FE			
FB6D:	B5	54		529	MUL3	LDA	XTNDL+2,X		ADD MPLCND (AUX)	*18
FB6F:	75	56		530		ADC	AUXL+2,X		TC PARTIAL PROD	
FB71:	95	54		531		STA	XTNDL+2,X		(XTND).	
FB73:	E8			532		INX				
FB74:	D0	F7		533		BNE	MUL3			
FB76:	A2	03		534	MUL4	LDX	#\$03			
FB78:	76			535	MUL5	DFB	#\$76			*19
FB79:	50			536		DFB	#\$50			
FB7A:	CA			537		DEX				
FB7B:	10	FB		538		BPL	MUL5			
FB7D:	88			539		DEY				
FB7E:	D0	E5		540		BNE	MUL2			
FB80:	60			541		RTS				
FB81:	20	A4	FB	542	DIVPM	JSR	MD1		ABS VAL OF AC, AUX.	*20
FB84:	A0	10		543	DIV	LDY	#\$10		INDEX FOR 16 BITS	*21
FB86:	06	50		544	DIV2	ASL	ACL			
FB88:	26	51		545		ROL	ACH			
FB8A:	26	52		546		ROL	XTNDL		XTND/AUX	

1. Normale Rückkehr nach der XEQ-Ausführung.
2. Erzwungener Sprung zur Erneuerung des Befehlszählers.
3. Befehle für den XEQ-Bereich.
4. Rücksetzen der Analog-Zeitkreise.
5. Zähler zurücksetzen.
6. Ausgleich für die erste Zählung.
7. Das Y-Register wird alle 12 us um 1 erhöht.
8. Ausgang bei dem maximalen Zählerstand von 255.
9. Lösche STATUS für die Fehlerhilfen.
10. Einstellen der ersten Seite Lo-Res Graphik.
11. Setzt den Text-Modus und öffnet das Textfenster auf volle Größe.
12. Stellt den Graphik-Modus ein mit den unteren 4 Zeilen als Textfenster.
13. Öffnen des Textfensters auf 40 Spalten. Die oberste Zeilennummer steht im Akkumulator und die unterste Zeile ist Zeile 24.
14. Setzt den Ausgabezeiger auf die Spalte, deren Nummer im Akkumulator steht.
15. Absolutwert von AC und AUX bilden.
16. Zähler für 16 Bits.
17. Kein Teilprodukt, wenn das Carry-Bit nicht gesetzt ist.
18. Addiert den Multiplikatoren (AUX) zum Teilprodukt (XTND).
19. ROR ACL, X
20. Absolutwert von AC und AUX bilden.
21. Zähler für 16 Bits.

FB8C:	26	53	547	ROL	XTNDH	TO AC.		
FB8E:	38		548	SEC				
FB8F:	A5	52	549	LDA	XTNDL			
FB91:	E5	54	550	SBC	AUXL	MCD TO XTND.		
FB93:	AA		551	TAX				
FB94:	A5	53	552	LDA	XTNDH			
FB96:	E5	55	553	SBC	AUXH			
FB98:	90	06	554	BCC	DIV3			
FB9A:	86	52	555	STX	XTNDL			
FB9C:	85	53	556	STA	XTNDH			
FB9E:	E6	50	557	INC	ACL			
FBA0:	88		558	DEY				
FBA1:	D0	E3	559	BNE	DIV2			
FBA3:	60		560	RTS				
FBA4:	A0	00	561	LDY	#\$00	ABS VAL OF AC, AUX	*1	
FBA6:	84	2F	562	STY	SIGN	WITH RESULT SIGN		
FBA8:	A2	54	563	LDX	#AUXL	IN LSB OF SIGN.		
FBAA:	20	AF	564	JSR	MD2			
FBAD:	A2	50	565	LDX	#ACL			
FBAF:	B5	01	566	LDA	LOC1,X	X SPECIFIES AC OR AUX	*2	
FBB1:	10	0D	567	BPL	MDRTS			
FBB3:	38		568	SEC				
FBB4:	98		569	TYA				
FBB5:	F5	00	570	SBC	LOC0,X	COMPL SPECIFIED REG	*3	
FBB7:	95	00	571	STA	LOC0,X	IF NEG.		
FBB9:	98		572	TYA				
FBBA:	F5	01	573	SBC	LOC1,X			
FBBC:	95	01	574	STA	LOC1,X			
FBBE:	E6	2F	575	INC	SIGN			
FBC0:	60		576	MCRTS	RTS			
FBC1:	48		577	BASCALC	PHA	CALC BASE ADR IN BASL,H	*4	
FBC2:	4A		578	LSR	A	FOR GIVEN LINE NO.		
FBC3:	29	03	579	AND	#\$03	0<=LINE NO.<=\$17		
FBC5:	09	04	580	ORA	#\$04	ARG=000ABCDE, GENERATE	*5	
FBC7:	85	29	581	STA	BASH	BASH=000001CD		
FBC9:	68		582	PLA		AND		
FBCA:	29	18	583	AND	#\$18	BASL=EABAB000		
FBCB:	90	02	584	BCC	BSCLC2			
FBCD:	69	7F	585	ADC	#\$7F			
FBD0:	85	28	586	BSCCLC2	STA	BASL		
FBD2:	0A		587	ASL	A			
FBD3:	0A		588	ASL	A			
FBD4:	05	28	589	ORA	BASL			
FBD6:	85	28	590	STA	BASL			
FBD8:	60		591	RTS				
FBD9:	C9	87	592	BELL1	CMP	#\$87	BELL CHAR? (CNTRL-G)	*6
FBDE:	D0	12	593	BNE	RTS2B	NO, RETURN		
FBDD:	A9	40	594	LDA	#\$40	DELAY .01 SECONDS	*7	
FBDF:	20	A8	595	JSR	WAIT			
FBE2:	A0	0C	596	LDY	#\$C0			
FBE4:	A9	0C	597	BELL2	LDA	#\$0C	TOGGLE SPEAKER AT	*8
FBE6:	20	A8	598	JSR	WAIT	1 KHZ FOR .1 SEC.		
FBE9:	AD	30	599	LDA	SPKR			
FBEC:	88		600	DEY				
FBED:	D0	F5	601	BNE	BELL2			
FBEF:	60		602	RTS2B	RTS			
FBF0:	A4	24	603	STOADV	LDY	CH	CURSER H INDEX TO Y-REG	*9
FBF2:	91	26	604	STA	(BASL),Y		STOR CHAR IN LINE	*10
FBF4:	E6	24	605	ADVANCE	INC	CH	INCREMENT CURSER H INDEX	*11
FBF6:	A5	24	606	LDA	CH		(MOVE RIGHT)	
FBF8:	C5	21	607	CMP	WNDWDTH		BEYOND WINDOW WIDTH?	*12
FBFA:	B0	66	608	BCC	CR		YES CR TO NEXT LINE	
FBFC:	60		609	RTS3	RTS		NO, RETURN	
FBFD:	C9	A0	610	VIDCUT	CMP	#\$A0	CONTROL CHAR?	*13
FBFF:	B0	EF	611	BNE	STOADV		NO, OUTPUT IT.	
FC01:	A6		612	TAY			INVERSE VIDEO?	*14
FC02:	10	EC	613	BPL	STOADV		YES, OUTPUT IT.	
FC04:	C9	8D	614	CMP	#\$8D	CR?		
FC06:	F0	5A	615	BEQ	CR		YES.	
FC08:	C9	8A	616	CMP	#\$8A		LINE FEED?	*15
FC0A:	F0	5A	617	BEQ	LF		IF SO, DO IT.	
FC0C:	C9	88	618	CMP	#\$88		BACK SPACE? (CNTRL-H)	*16
FC0E:	D0	C9	619	BNE	BELL1		NO, CHECK FOR BELL.	

1. Absolutwert von AC und AUX bilden, wobei das resultierende Vorzeichen in Bit 0 von SIGN steht.
2. X enthält die Adresse von AC bzw. AUX.
3. Falls das angegebene Register negativen Inhalt hat, wird das Komplement gebildet.
4. Die Basisadresse wird für eine bestimmte Zeilennummer gebildet. Die Zeilennummer liegt im Bereich von \$00 bis \$17.
5. Aus 000ABCDE wird BASH = 000001CD und BASL = EABAB000 gebildet.
6. Klingelzeichen **CTRL 6** ? Rücksprung, falls es nicht vorliegt.
7. 0,01 Sekunden warten.
8. 0,1 Sekunden-Signal mit 1 kHz am Lautsprecher.
9. Horizontalen Zeiger-Index in das Y-Register bringen.
10. Zeichen ausgeben.
11. Den Zeiger nach rechts bewegen : den horizontalen Index um 1 erhöhen.
12. Ist die Breite des Textfensters überschritten? Nein : Rücksprung. Ja : Ausgabe eines CR (Carriage return : Wagenrücklauf und Zeilenvorschub. Hier : Sprung an den Anfang einer neuen Zeile).
13. Steuerzeichen? Nein : Ausgabe des Zeichens.
14. Inverses Zeichen (schwarz auf weißem Hintergrund)? Ja : Ausgabe des Zeichens.
15. Zeilenvorschub?
16. Rückwärtsschritt (back space)? Nein : Auf Klingelzeichen testen.

FC10:	C6 24	620	BS	DEC	CH	DECREMENT CURSER H INDEX	*1
FC12:	10 E8	621		BPL	RTS3	IF POS, OK. ELSE MOVE UP	*2
FC14:	A5 21	622		LDA	WNDWDTH	SET CH TO WNDWDTH-1	*3
FC16:	85 24	623		STA	CH		
FC18:	C6 24	624		DEC	CH	(RIGHTMOST SCREEN POS)	
FC1A:	A5 22	625	UP	LDA	WNDTOP	CURSER V INDEX	*4
FC1C:	C5 25	626		CMP	CV		
FC1E:	B0 JB	627		BCS	RTS4	IF TOP LINE THEN RETURN	*5
FC20:	C6 25	628		DEC	CV	DECR CURSER V-INDEX	*6
FC22:	A5 25	629	VTAB	LDA	CV	GET CURSER V-INDEX	*7
FC24:	20 C1 FB	630	VTABZ	JSR	BASCALC	GENERATE BASE ADDR	*8
FC27:	65 20	631		ADC	WNDLFT	ADD WINDOW LEFT INDEX	
FC29:	85 28	632		STA	BASL	TO BASL	
FC2B:	60	633		RTS			
FC2C:	49 C0	634	RTS4	RTS			
FC2E:	F0 28	635	ESC1	EOB	#\$CU	ESC?	*9
FC30:	69 FD	636		BEQ	HOME	IF SO, DO HOME AND CLEAR	
FC32:	90 C0	637		ADC	#\$FD	ESC-A OR B CHECK	*10
FC34:	F0 DA	638		BCC	ADVANCE	A, ADVANCE	
FC36:	69 FD	639		BEQ	BS	B, BACKSPACE	
FC38:	90 2C	640		ADC	#\$FD	ESC-C OR D CHECK	*11
FC3A:	F0 DE	641		BCC	LF	C, DOWN	
FC3C:	69 FD	642		BEQ	UP	D, GO UP	
FC3E:	90 5C	643		ADC	#\$FD	ESC-E OR F CHECK	*12
FC40:	D0 59	644		BCC	CLREOL	E, CLEAR TO END OF LINE	
FC42:	A4 24	645	CLRECP	BNE	RTS4	NOT F, RETURN	
FC44:	A5 25	646		LDY	CH	CURSOR H TO Y INDEX	*13
FC46:	48	647	CLEOP1	LDA	CV	CURSOR V TO A-REGISTER	
FC47:	20 24 FC	648		PHA		SAVE CURRENT LINE ON STK	*14
FC4A:	20 9E FC	649		JSR	VTABZ	CALC BASE ADDRESS	*15
FC4D:	A0 00	650		JSR	CLEOLZ	CLEAR TO EOL, SET CARRY	*16
FC4F:	68	651		LDY	#\$00	CLEAR FROM H INDEX=0 FOR REST	*17
FC50:	69 00	652		PLA		INCREMENT CURRENT LINE	*18
FC52:	C5 23	653		ADC	#\$00	(CARRY IS SET)	
FC54:	90 F0	654		CMP	WNBDM	DONE TO BOTTOM OF WINDOW?	*19
FC56:	B0 CA	655		BCC	CLEOP1	NO, KEEP CLEARING LINES	
FC58:	A5 22	656	HOME	BCC	VTAB	YES, TAB TO CURRENT LINE	
FC5A:	85 25	657		LDA	WNDTOP	INIT CURSOR V	*20
FC5C:	A0 00	658		STA	CV	AND H-INDICES	
FC5E:	84 24	659		LDY	#\$00		
FC60:	F0 E4	660		STY	CH	THEN CLEAR TO END OF PAGE	*21
				BEQ	CLEOP1		

1. Zeiger nach links bewegen.
2. Rücksprung, falls positiv. Sonst nach oben bewegen.
3. Setzt den Zeiger auf Fensterbreite-1. (Position ganz rechts im Textfenster).
4. Vergleicht die senkrechte Zeigerposition mit der Obergrenze des Textfensters.
5. Rücksprung, falls die Zeilennummer größer oder gleich der obersten Zeilennummer ist.
6. Sonst geht der Zeiger wieder an die alte Stelle.
7. Holt die Spaltenposition des Zeigers.
8. Bildet die Basisadresse und addiert die linke Grenzposition des Textfensters.
9. ESC-@? Ja : Bildschirm löschen und Zeiger auf die Position links oben im Textfenster.
10. Test auf ESC-A und ESC-B. **A** : Vorwärtsschritt. **B** : Rückwärtsschritt.
11. Test auf ESC-C und ESC-D. **C** : nach unten. **D** : nach oben.
12. Test auf ESC-E und ESC-F. **E** : löscht den Rest der Zeile.
Rücksprung, falls nicht **F** gegeben wurde.
13. Spaltenposition des Zeigers ins Y-Register. Zeilenposition in den Akkumulator.
14. Zeilenposition des Zeigers wird in den Keller gegeben.
15. Basisadresse errechnen.
16. Löscht den Rest der Zeile und setzt das Carry-Bit.
17. Löscht mit dem horizontalen Index 0 den Rest des Textfensters.
18. Nächste Zeile.
19. Weitere Zeilen löschen, bis das untere Ende des Textfensters erreicht ist. Dann geht der Zeiger wieder an die Ausgangsstelle.
20. Zeigerposition auf links oben im Textfenster stellen.
21. Bis zum Ende des Textfensters löschen.

FC62:	A9 00	661	CR	LDA	#\$00	CURSOR TO LEFT OF INDEX	*1
FC64:	85 24	662		STA	CH	(RET CURSOR H=0)	
FC66:	E6 25	663	LF	INC	CV	INCR CURSOR V(DOWN 1 LINE)	*2
FC6d:	A5 25	664		LDA	CV		
FC6A:	C5 23	665		CMP	WNDBTM	OFF SCREEN?	*3
FC6C:	90 B6	666		BCC	VTABZ	NO, SET BASE ADDR	
FC6E:	C6 25	667		DEC	CV	DECR CURSOR V(BACK TO BOTTOM)	*4
FC70:	A5 22	668	SCROLL	LDA	WINDTOP	START AT TOP OF SCRL WNDW	*5
FC72:	48	669		PHA			
FC73:	20 24 FC	670		JSR	VTABZ	GENERATE BASE ADDRESS	*6
FC76:	A5 28	671	SCRL1	LDA	BASL	CCPY BASL,H	*7
FC78:	85 2A	672		STA	BAS2L	TO BAS2L,H	
FC7A:	A5 29	673		LDA	BASH		
FC7C:	85 2B	674		STA	BAS2H		
FC7E:	A4 21	675		LDY	WINDWTH	INIT Y TO RIGHTMOST INDEX	*8
FC80:	68	676		DEY		OF SCROLLING WINDOW	
FC81:	68	677		PLA			
FC82:	69 01	678		ADC	#\$01	INCR LINE NUMBER	*9
FC84:	C5 23	679		CMP	WNDBTM	DONE?	*10
FC86:	B0 0D	680		BCS	SCRL3	YES, FINISH	
FC88:	48	681		PHA			
FC89:	20 24 FC	682		JSR	VTABZ	FORM BASL,H (BASE ADDR)	*11
FC8C:	B1 28	683	SCRL2	LDA	(BASL),Y	MOVE A CHR UP ON LINE	*12
FC8E:	91 2A	684		STA	(BAS2L),Y		
FC90:	88	685		DEY		NEXT CHAR OF LINE	*13
FC91:	10 F9	686		BPL	SCRL2		
FC93:	30 E1	687		BMI	SCRL1	NEXT LINE	*14
FC95:	A0 00	688	SCRL3	LDY	#\$00	CLEAR BOTTOM LINE	*15
FC97:	20 9E FC	689		JSR	CLEOLZ	GET BASE ADDR FOR BOTTOM LINE	*16
FC9A:	B0 86	690		BCS	VTAB	CARRY IS SET	*17
FC9C:	A4 24	691	CLREOL	LDY	CH	CURSOR H INDEX	*18
FC9E:	A9 A0	692	CLEOLZ	LDA	#\$A0		

1. Setzt den Ausgabezeiger links ins Textfenster.
2. Bewegt den Zeiger eine Zeile tiefer.
3. Außerhalb des Textfensters? Nein : Basisadresse bestimmen.
4. Sonst den Zeiger wieder auf die unterste Zeile setzen.
5. Oben im Textfenster anfangen.
6. Basisadresse bestimmen.
7. BASL, BASH nach BAS2L, BAS2H bringen.
8. Y bekommt den Wert der rechten Grenze des Textfensters.
9. Zeilennummer erhöhen.
10. Unterste Zeile erreicht? Wenn ja, dann ist das Schieben beendet.
11. Basisadresse bilden.
12. Ein Zeichen eine Zeile nach oben bewegen.
13. Das nächste Zeichen der Zeile.
14. Die nächste Zeile.
15. Löscht die unterste Zeile.
16. Basisadresse für die unterste Zeile bilden.
17. Erzwungener Sprung nach VTAB.
18. Die horizontale Zeigerposition ins Y-Register bringen.

FCA0:	91 28	693	CLEOL2	STA	(BASL),Y	STORE BLANKS FROM 'HERE'	*1
FCA2:	C8	694		INY		TO END OF LINES (WNDWDTH)	
FCA3:	C4 21	695		CPY	WNDWDTH		
FCA5:	90 F9	696		BCC	CLEOL2		
FCA7:	60	697		RTS			
FCA8:	38	698	WAIT	SEC			
FCA9:	48	699	WAIT2	PHA			
FCAA:	E9 01	700	WAIT3	SBC	#\$01		
FCAC:	D0 FC	701		BNE	WAIT3	1.0204 USEC	*2
FCAE:	68	702		PLA		(13+2712*A+512*A*A)	
FCAF:	E9 01	703		SBC	#\$01		
FCB1:	D0 F6	704		BNE	WAIT2		
FCB3:	60	705		RTS			
FCB4:	E6 42	706	NXTA4	INC	A4L	INCR 2-BYTE A4	*3
FCB6:	D0 02	707		BNE	NXTA1	AND A1	
FCB8:	E6 43	708		INC	A4H		
FCBA:	A5 3C	709	NXTA1	LDA	A1L	INCR 2-BYTE A1.	*4
FCBC:	C5 3E	710		CMF	A2L		
FCBE:	A5 3D	711		LDA	A1H	AND COMPARE TO A2	*5
FCC0:	E5 3F	712		SBC	A2H		
FCC2:	E6 3C	713		INC	A1L	(CARRY SET IF >=)	
FCC4:	D0 02	714		BNE	RTS4B		
FCC6:	E6 3D	715		INC	A1H		
FCC8:	60	716	RTS4B	RTS			
FCC9:	A0 4B	717	HEADR	LDY	#\$4B	WRITE A*256 'LONG 1'	*6
FCCB:	20 DB FC	718		JSR	ZERDLY	HALF CYCLES	
FCCE:	D0 F9	719		BNE	HEADR	(650 USEC EACH)	
FCD0:	69 FE	720		ADC	#\$FE		
FCD2:	B0 F5	721		BCS	HEADR	THEN A 'SHORT 0'	*7
FCD4:	A0 21	722		LDY	#\$21	(400 USEC)	
FCD6:	20 DB FC	723	WRBIT	JSR	ZERDLY	WRITE TWO HALF CYCLES	*8
FCD9:	C8	724		INY		OF 250 USEC ('0')	
FCDA:	C8	725		INY		OR 500 USEC ('0')	
FCEB:	88	726	ZERDLY	DEY			
FDC:	D0 FD	727		BNE	ZERDLY		
FCDE:	90 05	728		BCC	WRTAPE	Y IS COUNT FOR	*9
FCE0:	A0 32	729		LDY	#\$32	TIMING LOOP	
FCE2:	88	730	ONEDLY	DEY			
FCE3:	D0 FD	731		BNE	ONEDLY		
FCE5:	AC 20 C0	732	WRTAPE	LDY	TAPEOUT		
FCE8:	A0 2C	733		LDY	#\$2C		
FCEA:	CA	734		DEX			
FCEB:	60	735		RTS			
FCEC:	A2 08	736	RDBYTE	LDX	#\$08	8 BITS TO READ	*10
FCEE:	48	737	RDBYT2	PHA		READ TWO TRANSITIONS	*11
FCEF:	20 FA FC	738		JSR	RD2BIT	(FIND EDGE)	
FCF2:	68	739		PLA			
FCF3:	2A	740		ROL	A	NEXT BIT	*12
FCF4:	A0 3A	741		LDY	#\$3A	COUNT FOR SAMPLES	
FCF6:	CA	742		DEX			
FCF7:	D0 F5	743		BNE	RDBYT2		
FCF9:	60	744		RTS			
FCFA:	20 FD FC	745	RD2BIT	JSR	RDBIT		
FCFB:	88	746	RDBIT	DEY		DECR Y UNTIL	*13
FCFE:	AD 60 C0	747		LDA	TAPEIN	TAPE TRANSITION	
FD01:	45 2F	748		EOR	LASTIN		
FD03:	10 F8	749		BPL	RDBIT		
FD05:	45 2F	750		EOR	LASTIN		
FD07:	85 2F	751		STA	LASTIN		
FD09:	C0 80	752		CPY	#\$80	SET CARRY ON Y-REG.	
FD0B:	60	753		RTS			
FD0C:	A4 24	754	RDKEY	LDY	CH		
FD0E:	B1 28	755		LDA	(BASL),Y	SET SCREEN TO FLASH	*14
FD10:	48	756		PHA			
FD11:	29 3F	757		AND	#\$3F		
FD13:	09 40	758		ORA	#\$40		
FD15:	91 20	759		STA	(BASL),Y		
FD17:	68	760		PLA			
FD18:	C8 38 00	761		JMP	(KSWL)	GO TO USER KEY-IN	*15
FD1B:	E6 4E	762	KEYIN	INC	RNDL		
FD1D:	D0 02	763		BNE	KEYIN2	INCR RND NUMBER	*16
FD1F:	E6 4F	764		INC	RNDH		
FD21:	2C 00 C0	765	KEYIN2	BIT	KBD	KEY DOWN?	

1. Leerzeichen von der aktuellen Position des Zeigers bis zum Ende der Zeile auf den Bildschirm bringen.
2. Wartezeit : $1,0204$ us.
3. 2-Byte-„Register“ A4 und A1 jeweils um 1 erhöhen.
4. 2-Byte-„Register“ A1 um 1 erhöhen.
5. A1 mit A2 vergleichen und Carry setzen, falls $A1 \geq A2$ ist.
6. Schreibt $A * 256$ mal eine „lange 1“ (jeweils 650 us) auf Band.
7. Dann folgt eine „kurze 0“ (von 400 us).
8. Schreibt zwei halbe Zyklen von 250 us - „0“ oder 500 us - „0“.
9. Y ist Zähler für die Zeitschleife.
10. 8 Bits sind zu lesen.
11. Liest zwei Übergänge.
12. Nächstes Bit.
13. Vermindert Y, bis zum nächsten 0-1 oder 1-0 Übergang auf Band.
14. Läßt das Zeichen an der Ausgabe-Position als blinkenden Eingabe-Zeiger erscheinen.
15. Springt zum aktuellen Eingabe-Unterprogramm.
16. Erhöht die Zufallszahl.
17. Taste gedrückt?

FD24:	10 F5	766		BFL	KEYIN	LOOP		
FD26:	91 20	767		STA	(BASL),Y	REPLACE FLASHING SCREEN		*1
FD28:	AD 00	CU 768		LDA	KBD	GET KEYCODE		*2
FD2B:	2C 1U	CU 769		BIT	KBDSTRB	CLR KEY STROBE		*3
FD2E:	60	770		RTS				
FD2F:	20 0C	FD 771	ESC	JSR	RDKEY	GET KEYCODE		*4
FD32:	20 2C	FC 772		JSR	ESC1	HANDLE ESC FUNC.		
FD35:	20 0C	FD 773	RDCHAR	JSR	RDKEY	READ KEY		*5
FD38:	C9 9B	774		CMP	#\$9B	ESC?		*6
FD3A:	F0 F3	775		BEQ	ESC	YES, DON'T RETURN		
FD3C:	60	776		RTS				
FD3D:	A5 32	777	NOTCR	LDA	INVFLG			
FD3F:	48	778		PHA				
FD40:	A9 FF	779		LDA	#\$FF			
FD42:	85 32	780		STA	INVFLG	ECHO USER LINE		*7
FD44:	BD 00	02 781		LDA	IN,X	NON INVERSE		
FD47:	20 ED	FD 782		JSR	COUT			
FD4A:	68	783		PLA				
FD4B:	85 32	784		STA	INVFLG			
FD4D:	BD 00	02 785		LDA	IN,X			
FD4E:	C9 88	786		CMP	#\$88	CHECK FOR EDIT KEYS		*8
FD52:	F0 1D	787		BEQ	BCKSPC	BS, CTRL-X.		
FD54:	C9 98	788		CMP	#\$98			
FD56:	F0 0A	789		BEQ	CANCEL			
FD58:	E0 F8	790		CFX	#\$F8	MARGIN?		*9
FD5A:	90 03	791		BCC	NOTCR1			
FD5C:	20 3A	FF 792		JSR	BELL	YES, SOUND BELL		
FD5F:	E8	793	NOTCR1	INX		ADVANCE INPUT INDEX		*10
FD60:	D0 13	794		BNE	NXTCHAR			
FD62:	A9 DC	795	CANCEL	LDA	#\$DC	BACKSLASH AFTER CANCELLED LTN		*11
FD64:	20 ED	FD 796		JSR	COUT			
FD67:	20 8E	FD 797	GETLNZ	JSR	CROUT	OUTPUT CR		*12
FD6A:	A5 33	798	GETLN	LDA	PROMPT			
FD6C:	20 ED	FD 799		JSR	COUT	OUTPUT PROMPT CHAR		*13
FD6F:	A2 01	800		LDX	#\$01	INIT INPUT INDEX		*14
FD71:	8A	801	BCKSPC	TXA		WILL BACKSPACE TO U		
FD72:	F0 F3	802		BEQ	GETLNZ			
FD74:	CA	803		DEX				
FD75:	20 35	FD 804	NXTCHAR	JSR	RDCHAR			
FD78:	C9 95	805		CMP	#\$PICK	USE SCREEN CHAR		*15
FD7A:	D0 02	806		BNE	CAPTST	FOR CTRL-U		
FD7C:	B1 28	807		LDA	(BASL),Y			
FD7E:	C9 E0	808	CAPTST	CMP	#\$E0			
FD80:	90 02	809		BCC	ADDINP	CONVERT TO CAPS		*16
FD82:	29 0F	810		AND	#\$0F			
FD84:	9D 00	02 811	ADDINP	STA	IN,X	ADD TO INPUT BUF		*17
FD87:	C9 8D	812		CMP	#\$8D			
FD89:	D0 82	813		BNE	NOTCR			
FD8B:	20 9C	FC 814		JSR	CLREOL	CLR TO EOL IF CR		*18
FD8E:	A9 8D	815	CROUT	LDA	#\$8D			
FD90:	D0 5B	816		BNE	COUT			
FD92:	A4 3D	817	PRA1	LDY	A1H	PRINT CR,A1 IN HEX		*19
FD94:	A6 3C	818		LDX	A1L			
FD96:	20 8E	FD 819	PRYX2	JSR	CROUT			
FD99:	20 40	F9 820		JSR	PRNTYX			
FD9C:	A0 00	821		LDY	#\$00			
FD9E:	A9 AD	822		LDA	#\$AD	PRINT '-'		*20
FDA0:	4C ED	FD 823		JMP	COUT			
FDA3:	A5 3C	824	XAN8	LDA	A1L			
FDA5:	09 07	825		ORA	#\$07	SET TO FINISH AT		*21
FDA7:	85 3E	826		STA	A2L	MOD 8=7		
FDA9:	A5 3D	827		LDA	A1H			
FDA8:	85 3F	828		STA	A2H			
FDA0:	A5 3C	829	MOD3CHK	LDA	A1L			
FDAF:	29 07	830		AND	#\$07			
FDB1:	DJ 03	831		BNE	DATAOUT			
FDB3:	20 92	FD 832	XAM	JSR	PRA1			
FDB6:	A9 A0	833	DATAOUT	LDA	#\$A0			
FDB8:	20 ED	FD 834		JSR	COUT	OUTPUT BLANK		*22
FDBB:	B1 3C	835		LDA	(A1L),Y			
FDBD:	20 DA	FD 836		JSR	PRBYTE	OUTPUT BYTE IN HEX		*23
FDC0:	20 BA	FC 837		JSR	NXTA1			

1. Den blinkenden Zeiger durch das ursprüngliche Zeichen ersetzen.
2. Holt den Code der gedrückten Taste.
3. Tastendruck löschen.
4. Holt ein Zeichen und behandelt es als ESC-Anweisung.
5. Holt ein Zeichen.
6. Ist es ESC? Ja : Sprung zur ESC-Behandlung. Nein : Rücksprung.
7. Gibt die Eingaben des Benutzers im Normal-Modus aus.
8. Test auf ←, → und **CTRL X**.
9. Ist das 249. Eingabezeichen erreicht oder überschritten? Ja : BELL.
10. Zähler des Eingabepuffers um 1 erhöhen.
11. BACKSLASH (\) nach der gelöschten Eingabezeile.
12. Ausgabezeiger auf den Anfang der nächsten Zeile setzen.
13. Bereitschaftszeichen ausgeben.
14. Den Zähler des Eingabepuffers auf den Anfangswert setzen.
15. **CTRL U** : Zeichen auf der Bildschirmposition eingeben.
16. Formt Kleinbuchstaben (\$E0-\$FF) in Großbuchstaben (\$A0-\$BF) um.
17. Gibt das Zeichen in den Eingabepuffer.
18. Bei der Eingabe von **RETURN** wird bis zum Ende der Bildschirmzeile gelöscht.
19. Gibt den Inhalt von A 1 hexadezimal am Anfang der nächsten Zeile aus.
20. Drückt ein Minuszeichen (-).
21. Ende bei Mod 8 = 7 festlegen.
22. Ausgabe eines Leerzeichens.
23. Das Byte hexadezimal ausgeben.

FDC5:	90 E6	838		BCC	MOD8CHK	CHECK IF TIME TO,	
FDC5:	60	839	RTS4C	RTS		PRINT ADDR	
FDC6:	4A	840	XAMPM	LSR	A	DETERMINE IF MCN	*1
FDC7:	90 EA	841		BCC	XAM	MODE IS XAM	
FDC9:	4A	842		LSR	A	ADD, OR SUB	
FCCA:	4A	843		LSR	A		
FDCB:	A5 3E	844		LDA	A2L		
FDCD:	90 U2	845		BCC	ADD		
FDCF:	49 FF	846		EOR	#SFF	SUB: FORM 2'S COMPLEMENT	*2
FDD1:	65 3C	847	ADD	ADC	ALL		
FDD3:	48	848		PHA			
FDD4:	A9 BD	849		LDA	#SBD		
FDD6:	20 ED FD	850		JSR	COUT	PRINT '=' , THEN RESULT	*3
FDD9:	08	851		PLA			
FDDA:	48	852	PRBYTE	PHA		PRINT BYTE AS 2 HEX	*4
FDDB:	4A	853		LSR	A	DIGITS, DESTROYS A-REG	
FDDC:	4A	854		LSR	A		
FDDD:	4A	855		LSR	A		
FDEE:	4A	856		LSR	A		
FDDF:	20 E5 FD	857		JSR	PRHEXZ		
FDE2:	68	858		PLA			
FDE3:	29 JF	859	PRHEX	AND	#SUF	PRINT HEX DIG IN A-REG	*5
FDE5:	09 30	860	PRHEXZ	CRA	#SBO	LSB'S	
FDE7:	C9 BA	861		CMP	#SBA		
FDE9:	90 U2	862		BCC	COUT		
FDEB:	69 U6	863		ADC	#S06		
FDED:	6C 58 UJ	864	CCUT	JMP	(CSWL)	VECTOR TO USER OUTPUT ROUTINE	*6
FDF0:	C9 A0	865	COUT1	CMP	#SA0		
FDF2:	90 J2	866		BCC	COUTZ	DON'T OUTPUT CIRL'S INVERSE	*7
FDF4:	25 32	867		AND	INVFLG	MASK WITH INVERSE FLAG	*8
FDF6:	84 35	868	COUTZ	STY	YSAV1	SAV Y-REG	*9
FDF8:	48	869		PHA		SAV A-REG	
FDF9:	20 FD FB	870		JSR	VIDOUT	CUTPUT A-REG AS ASCII	*10
FDFC:	68	871		PLA		RESTORE A-REG	*11
FDFD:	A4 55	872		LDY	YSAV1	AND Y-REG	
FDFE:	60	873		RTS		THEN RETURN	
FE00:	C6 34	874	BL1	DEC	YSAV		
FE02:	F0 9F	875		BEQ	XAMB		
FE04:	CA	876	BLANK	DEX		BLANK TO MON	
FE05:	D0 16	877		BNE	SETMDZ	AFTER BLANK	
FE07:	C9 BA	878		CMP	#SBA	DATA STORE MODE?	*12
FE09:	D0 BB	879		BNE	XAMPM	NO, XAM, ADD OR SUB	
FE0B:	85 31	880	STOR	STA	MODE	KEEP IN STORE MCDE	*13
FE0D:	A5 3E	881		LDA	A2L		
FE0F:	91 40	882		STA	(A3L),Y	STORE AS LOW BYTE AS (A3)	*14
FE11:	E6 40	883		INC	A3L		
FE13:	D0 U2	884		BNE	RTS5	INCR A3, RETURN	*15
FE15:	E6 41	885		INC	A3H		
FE17:	60	886	RTS5	RTS			
FE18:	A4 34	887	SETMODE	LDY	YSAV	SAVE CONVERTED ':', '+',	*16
FE1A:	B9 FF 01	888		LDA	IN-1,Y	'-', '.' AS MODE.	
FE1D:	85 31	889	SETMDZ	STA	MODE		
FE1F:	60	890		RTS			
FE20:	A2 01	891	LT	LDX	#S01		
FE22:	B5 3E	892	LT2	LDA	A2L,X	COPY A2 (2 BYTES) TO	*17
FE24:	95 42	893		STA	A4L,X	A4 AND A5	
FE26:	95 44	894		STA	A5L,X		
FE28:	CA	895		DEX			
FE29:	10 F7	896		BPL	LT2		
FE2B:	60	897		RTS			
FE2C:	B1 3C	898	MOVE	LDA	(A1L),Y	MOVE (A1 TO A2) TO	*18
FE2E:	91 42	899		STA	(A4L),Y	(A4)	
FE30:	20 B4 FC	900		JSR	NXTA4		
FE33:	90 F7	901		BCC	MOVE		
FE35:	60	902		RTS			
FE36:	B1 3C	903	VFY	LDA	(A1L),Y	VERIFY (A1 TO A2) WITH	*19
FE38:	D1 42	904		CMP	(A4L),Y	(A4)	
FE3A:	F0 1C	905		BEQ	VFYOK		
FE3C:	20 92 FD	906		JSR	PRA1		
FE3F:	B1 3C	907		LDA	(A1L),Y		
FE41:	20 DA FD	908		JSR	PRBYTE		
FE44:	A9 A0	909		LDA	#SAU		
FE46:	20 ED FD	910		JSR	COUT		

1. Feststellen, ob der Monitor-Zustand XAM (prüfen), ADD oder SUB ist.
2. SUB: Zweier-Komplement bilden.
3. Drucke „=“, dann das Ergebnis.
4. Gibt das Byte als zwei Hexadezimalziffern aus. Der Akkumulator wird verändert.
5. Gibt die rechten vier Bits des Akkumulators als Hexadezimalziffer aus.
6. Springt zum aktuellen Zeichenausgabe-Unterprogramm.
7. Gibt die Steuerzeichen nicht invers oder blinkend aus.
8. Maskierung für inverse und blinkende Zeichen.
9. Die Inhalte des Akkumulators und des Y-Registers retten.
10. Den Inhalt des Akkumulators als ASCII-Zeichen ausgeben.
11. Akkumulator und Y-Register wiederherstellen.
12. Speicherzustand? Nein: XAM, ADD oder SUB.
13. Im Speicherzustand bleiben.
14. A2L nach (A3) speichern.
15. A3 um 1 erhöhen.
16. „:“, „=“, „-“ oder „.“, als Zustandskennzeichen in MODE eintragen.
17. A2 nach A4 und A5 bringen.
18. Den Bereich A1 bis A2 nach dem Bereich ab A4 bringen.
19. Den Bereich von A1 bis A2 mit dem Bereich ab A4 vergleichen.

FE49:	A9 A8	911		LDA	#SA8		
FE4B:	20 ED	FD 912		JSR	CCUT		
FE4E:	B1 42	913		LDA	(A4L),Y		
FE50:	20 DA	FD 914		JSR	PRBYTE		
FE53:	A9 A9	915		LDA	#SA9		
FE55:	20 ED	FD 916		JSR	COUT		
FE58:	20 B4	FC 917	VFYOK	JSR	NXTA4		
FE5B:	90 E9	918		BCC	VFY		
FE5D:	60	919		RTS			
FE5E:	20 75	FE 920	LIST	JSR	ALPC	MOVE A1 (2 BYTES) TO	*1
FE61:	A9 14	921		LDA	#S14	PC IF SPEC'D AND	
FE63:	48	922	LIST2	PHA		DISSEMBLE 20 INSTRS	
FE64:	20 D0	F8 923		JSR	INSTD5P		
FE67:	20 53	F9 924		JSR	PCADJ	ADJUST PC EACH INSTR	*2
FE6A:	65 3A	925		STA	PCL		
FE6C:	84 3B	926		STY	PCH		
FE6E:	68	927		PLA			
FE6F:	38	928		SEC			
FE70:	E9 01	929		SBC	#S01	NEXT OF 20 INSTRS	*3
FE72:	D0 EF	930		BNE	LIST2		
FE74:	60	931		RTS			
FE75:	8A	932	ALPC	TXA		IF USER SPEC'D ADR	*4
FE76:	F0 07	933		BEQ	ALPCRTS	COPY FROM A1 TO PC	
FE78:	B5 3C	934	ALPCLP	LDA	A1L,X		
FE7A:	95 3A	935		STA	PCL,X		
FE7C:	CA	936		DEX			
FE7D:	10 F9	937		BPL	ALPCLP		
FE7F:	60	938	ALPCRTS	RTS			
FE80:	A0 3F	939	SETINV	LDY	#S3F	SET FOR INVERSE VID	*5
FE82:	D0 02	940		BNE	SETIFLG	VIA COUT1	
FE84:	A0 FF	941	SETNORM	LDY	#SFF	SET FOR NORMAL VID	*6
FE86:	84 32	942	SETIFLG	STY	INVFLG		
FE88:	60	943		RTS			
FE89:	A9 00	944	SETKBD	LDA	#S00	SIMULATE PORT #0 INPUT	*7
FE8B:	85 3E	945	INPORT	STA	A2L	SPECIFIED (KEYIN ROUTINE)	
FE8D:	A2 38	946	INPRT	LDX	#KSWL		
FE8F:	A0 1B	947		LDY	#KEYIN		
FE91:	D0 08	948		BNE	IOPRT		
FE93:	A9 00	949	SETVID	LDA	#S00	SIMULATE PORT #0 OUTPUT	*8
FE95:	85 3E	950	OUTPORT	STA	A2L	SPECIFIED (COUT1 ROUTINE)	
FE97:	A2 36	951	OUTPRT	LDX	#CSWL		
FE99:	A0 F0	952		LDY	#COUT1		
FE9B:	A5 3E	953	IOPRT	LDA	A2L	SET RAM IN/OUT VECTORS	*9
FE9D:	29 0F	954		AND	#S0F		
FE9F:	F0 06	955		BEQ	IOPRT1		
FEA1:	09 C0	956		ORA	#IOADR/256		
FEA3:	A0 00	957		LDY	#S00		
FEA5:	F0 02	958		BEQ	IOPRT2		
FEA7:	A9 FD	959	IOPRT1	LDA	#COUT1/256		
FEA9:	94 00	960	IOPRT2	STY	LOC0,X		
FEAB:	95 01	961		STA	LOC1,X		
FEAD:	60	962		RTS			
FEAE:	EA	963		NOP			
FEAF:	EA	964		NOP			
FEB0:	4C 00	E0 965	XBASIC	JMP	BASIC	TO BASIC WITH SCRATCH	*10
FEB3:	4C 03	E0 966	BASCOMT	JMP	BASIC2	CONTINUE BASIC	*11
FEB6:	20 75	FE 967	GO	JSR	ALPC	ADR TO PC IF SPEC'D	*12
FEB9:	20 3F	FF 968		JSR	RESTORE	RESTORE META REGS	*13
FEBC:	6C 3A	00 969		JMP	(PCL)	GO TO USER SUBR	*14
FEBF:	4C D7	FA 970	REGZ	JMP	REGDSP	TO REG DISPLAY	*15
FEC2:	C6 34	971	TRACE	DEC	YSAV		
FEC4:	20 75	FE 972	STEPZ	JSR	ALPC	ADR TO PC IF SPEC'D	*16
FEC7:	4C 43	FA 973		JMP	STEP	TAKE ONE STEP	*17
FECA:	4C F8	03 974	USR	JMP	USRADR	TO USR SUBR AT USRADR	*18
FECD:	A9 40	975	WRITE	LDA	#S40		
FECF:	20 C9	FC 976		JSR	HEADR	WRITE 10-SEC HEADER	*19
FED2:	A0 27	977		LDY	#S27		
FED4:	A2 00	978	wr1	LDX	#S00		
FED6:	41 3C	979		EOR	(A1L,X)		
FED8:	48	980		PHA			
FED9:	A1 3C	981		LDA	(A1L,X)		

1. A1 nach PC bringen, falls eine Adresse angegeben wurde und 20 Befehle in Assemblerform ausgeben.
2. Nach jedem Befehl wird der Befehlszähler (PC) korrigiert.
3. Nächster Befehl.
4. Wenn eine Adresse angegeben wurde, wird A1 nach PC gebracht.
5. Setzt den INVERSE-Zustand für COUT1.
6. Setzt den NORMAL-Zustand.
7. Stellt die Standard-Eingabe auf die Tastatur ein.
8. Stellt die Standard-Ausgabe auf den Bildschirm ein.
9. Setzt die Ein- / Ausgabe-Verzweigungsadressen.
10. Sprung zu BASIC, wobei Programm und Variablen gelöscht werden.
11. In BASIC fortfahren. Programm und Variablen bleiben unverändert.
12. Adresse in den Befehlszähler (PC), falls sie angegeben wurde.
13. Die internen Register wiederherstellen.
14. Das Maschinenprogramm des Benutzers ausführen.
15. Die Register anzeigen.
16. Adresse in den Befehlszähler, falls eine angegeben wurde.
17. Einen Schritt ausführen.
18. Sprung zum Benutzerprogramm, dessen Adresse in USRADR liegt.
19. Eine 10-Sekunden-Kopfinformation schreiben.

FEDB:	20	ED	FE	932		JSR	WRBYTE			
FEDE:	20	BA	FC	983		JSR	NXTA1			
FEE1:	A0	ID		944		LDY	#\$1D			
FEE3:	68			985		PLA				
FEE4:	90	EE		986		BCC	WR1			
FEE6:	A0	22		987		LDY	#\$22			
FEE8:	20	ED	FE	988		JSR	WRBYTE			
FEEB:	F0	4D		989		BEQ	BELL			
FEED:	A2	10		990	WRBYTE	LDX	#\$10			
FEEF:	0A			991	WRBYT2	ASL	A			
FEF0:	20	D6	FC	992		JSR	WRBIT			
FEF3:	D0	FA		993		BNE	WRBYT2			
FEF5:	60			994		RTS				
FEF6:	20	00	FE	995	CRMON	JSR	BL1	HANDLE CR AS BLANK		*1
FEF9:	68			996		PLA		THEN POP STACK		
FEFA:	68			997		PLA		AND RTN TO MCN		
FEFB:	D0	6C		998		BNE	MONZ			
FEFD:	20	FA	FC	999	READ	JSR	RD2BIT	FIND TAPEIN EDGE		*2
FF00:	A9	16		1000		LDA	#\$16			
FF02:	20	C9	FC	1001		JSR	HEADR	DELAY 3.5 SECONDES		*3
FF05:	85	2E		1002		STA	CHKSUM	INIT CHKSUM=\$FF		*4
FF07:	20	FA	FC	1003		JSR	RD2BIT	FIND TAPEIN EDGE		*5
FF0A:	A0	24		1004	RD2	LDY	#\$24	LOOK FOR SYNC BIT		*6
FF0C:	20	FD	FC	1005		JSR	RDBIT	(SHORT 0)		
FF0F:	B0	F9		1006		BCS	RD2	LOOP UNTIL FOUND		
FF11:	20	FD	FC	1007		JSR	RDBIT	SKIP SECOND SYNC H-CYCLE		*7
FF14:	A0	3B		1008		LDY	#\$3B	INDEX FOR 0/1 TEST		*8
FF16:	20	EC	FC	1009	RD3	JSR	RDBYTE	READ A BYTE		*9
FF19:	81	3C		1010		STA	(A1L,X)	STORE AT (A1)		*10
FF1B:	45	2E		1011		EOR	CHKSUM			
FF1D:	85	2E		1012		STA	CHKSUM	UPDATE RUNNING CHKSUM		*11
FF1F:	20	BA	FC	1013		JSR	NXTA1	INCR A1, COMPARE TO A2		*12
FF22:	A0	35		1014		LDY	#\$35	COMPENSATE 0/1 INDEX		*13
FF24:	90	F0		1015		BCC	RD3	LOCP UNTIL DONE		
FF26:	20	EC	FC	1016		JSR	RDBYTE	READ CHKSUM BYTE		*14
FF29:	F5	2E		1017		CMP	CHKSUM			
FF2B:	C0	0D		1018		BEQ	BELL	GOOD, SOUND BELL AND RETURN		*15
FF2D:	A9	C5		1019	PRERR	LDA	#\$C5			
FF2F:	20	ED	FD	1020		JSR	COUT	PRINT "ERR", THEN BELL		*16
FF32:	A9	D2		1021		LDA	#\$D2			
FF34:	20	ED	FD	1022		JSR	COUT			
FF37:	20	ED	FD	1023		JSR	COUT			
FF3A:	A9	D7		1024	BELL	LDA	#\$87	OUTPUT BELL AND RETURN		*17
FF3C:	4C	ED	FD	1025		JMP	COUT			
FF3F:	A5	48		1026	RESTORE	LCA	STATUS	RESTORE 6502 REG CCNTEMTS		*18
FF41:	48			1027		PHA		USED BY DEBUG SOFTWARE		
FF42:	A5	45		1028		LDA	ACC			
FF44:	A6	46		1029	RESTR1	LDX	XREG			
FF46:	A4	47		1030		LDY	YREG			
FF48:	20			1031		PLP				
FF49:	60			1032		RTS				
FF4A:	85	45		1033	SAVE	STA	ACC	SAVE 6502 REG CONTENTS		*19
FF4C:	86	46		1034	SAV1	STX	XREG			
FF4E:	84	47		1035		STY	YREG			
FF50:	08			1036		PHP				
FF51:	68			1037		PLA				
FF52:	85	48		1038		STA	STATUS			
FF54:	8A			1039		TSX				
FF55:	86	49		1040		STX	SPNT			
FF57:	D8			1041		CLD				
FF58:	60			1042		RTS				
FF59:	20	64	FE	1043	RESET	JSR	SETNORM	SET SCREEN MODE		*20
FF5C:	20	2F	FB	1044		JSR	INIT	AND INIT KBD/SCREEN		
FF5F:	20	93	FE	1045		JSR	SETVID	AS I/O DEV'S		
FF62:	20	09	FE	1046		JSR	SETKBD			
FF65:	D8			1047	MON	CLD		MUST SET HEX MODE!		*21
FF66:	20	3A	FF	1048		JSR	BELL			
FF69:	A9	AA		1049	MONZ	LDA	#\$AA	'*' PROMPT FOR MCN		*22
FF6B:	85	33		1050		STA	PROMPT			
FF6D:	20	67	FD	1051		JSR	GETLNZ	READ A LINE		*23
FF70:	20	C7	FF	1052		JSR	ZMODE	CLEAR MON MODE, SCAN IDX		*24
FF73:	20	A7	FF	1053	NXTITM	JSR	GETNUM	GET ITEM, NON-HEX		*25
FF76:	84	34		1054		STY	YSAV	CHAR IN A-REG		

1. Das **RETURN**-Zeichen (CR) als Leerzeichen behandeln, eine Rücksprungadresse im Keller überspringen und zum Monitor zurückkehren.
2. Den Bandanfang suchen.
3. 3,5 Sekunden warten.
4. Prüfsumme auf den Anfangswert \$FF setzen.
5. Auf weitere Informationen vom Band warten.
6. Ein Synchronisationsbit suchen (kurze „0“).
7. Die zweite Hälfte des Synchronisationsbits überspringen.
8. Index für den 0/1-Test.
9. Ein Byte lesen.
10. In (A1) speichern.
11. Die Prüfsumme auf den neuesten Stand bringen.
12. A1 um 1 erhöhen und dann mit A2 vergleichen.
13. 0/1-Index ausgleichen.
14. Prüfsumme vom Band lesen.
15. Ein Ton vom Lautsprecher und Rückkehr, falls kein Fehler gefunden wurde.
16. Drückt „ERR“ und läßt den Lautsprecher einen Ton abgeben.
17. Gibt ein Klingelzeichen auf die Standard-Ausgabe.
18. Wiederherstellung der Werte der internen Register für die Fehlerhilfen.
19. Retten der internen Register in die Speicherstellen \$45 bis \$49.
20. Setzt den Normal-Zustand für die Zeichenausgabe und initialisiert das Tastenfeld, den Bildschirm und die Standard-Ein-/Ausgabegeräte.
21. Der Mikroprozessor soll Additionen und Subtraktionen mit 8 Bit binär ausführen und nicht jeweils vier Bit als Dezimalzahl auffassen.
22. Das Bereitschaftszeichen des Monitors.
23. Eine Eingabezeile anfordern.
24. Löscht den Monitorzustand.
25. Holt eine Information aus der Eingabezeile, wobei nichthexadezimale Zeichen im Akkumulator stehen.

FF78:	A0 17	1055		LDY	#S17	X-REG=0 IF NO HEX INPUT	*1
FF7A:	88	1056	CHRSRCH	DEY			
FF7B:	30 E8	1057		BMI	MON	NOT FOUND, GO TO MON	*2
FF7D:	D9 CC FF	1058		CMP	CHRTBL,Y	FIND CMND CHAR IN TEL	*3
FF80:	D0 F8	1059		BNE	CHRSRCH		
FF82:	20 BE FF	1060		JSR	TCSUB	FOUND, CALL CORRESPONDING	*4
FF85:	A4 34	1061		LDY	YSAV	SUBROUTINE	
FF87:	4C 73 FF	1062		JMP	NXTITM		
FF8A:	A2 03	1063	DIG	LDX	#S03		
FF8C:	0A	1064		ASL	A		
FF8d:	0A	1065		ASL	A	GOT HEX DIG,	*5
FF8E:	0A	1066		ASL	A	SHIFT INTO A2	
FF8F:	0A	1067		ASL	A		
FF90:	0A	1068	NXTBIT	ASL	A		
FF91:	26 3E	1069		ROL	A2L		
FF93:	26 3F	1070		ROL	A2H		
FF95:	CA	1071		DEX		LEAVE X=SFF IF DIG	*6
FF96:	10 F0	1072		3PL	NXTBIT		
FF98:	A5 31	1073	NXTBAS	LDA	MODE		
FF9A:	D0 06	1074		BNE	NXTBS2	IF MODE IS ZERO	*7
FF9C:	B5 3F	1075		LDA	A2H,X	THEN COPY A2 TO	
FF9E:	95 3D	1076		STA	A1H,X	A1 AND A3	
FFA0:	95 41	1077		STA	A3H,X		
FFA2:	E8	1078	NXTBS2	INX			
FFA3:	F0 F3	1079		BEQ	NXIBAS		
FFA5:	D0 06	1080		BNE	NXTCHR		
FFA7:	A2 00	1081	GETNUM	LDX	#S00	CLEAR A2	*8
FFA9:	86 3E	1082		STX	A2L		
FFAB:	86 3F	1083		STX	A2H		
FFAD:	39 00 02	1084	NXTCHR	LDA	IN,Y	GET CHAR	*9
FFB0:	C8	1085		INY			
FFB1:	49 B0	1086		EOR	#S00		
FFB3:	C9 0A	1087		CMP	#S0A		
FFB5:	90 D3	1088		BCC	DIG	IF HEX DIG, THEN	*10
FFB7:	69 48	1089		ADC	#S88		
FFB9:	C9 FA	1090		CMP	#SFA		
FFBB:	B0 CD	1091		BCS	DIG		
FFBD:	60	1092		RTS			
FFBE:	A9 FE	1093	TOSUB	LDA	#GO/256	PUSH HIGH-ORDER	*11
FFC0:	46	1094		PHA		SUBR ADR ON STK	
FFC1:	B9 E3 FF	1095		LDA	SUBTBL,Y	PUSH LOW ORDER	*12
FFC4:	48	1096		PHA		SUBR ADR ON STK	
FFC5:	A5 31	1097		LDA	MODE		
FFC7:	A0 00	1098	ZMODE	LDY	#S00	CLR MODE, OLD MODE	*13
FFC9:	84 31	1099		STY	MODE	TO A-REG	
FFCB:	60	1100		RTS		GO TO SUBR VIA RTS	*14
FFCC:	BC	1101	CHRTBL	DFB	SBC	F("CTRL-C")	
FFCD:	B2	1102		DFB	SB2	F("CTRL-Y")	
FFCE:	BE	1103		DFB	SBE	F("CTRL-E")	
FFCF:	ED	1104		DFB	SED	F("T")	
FFD0:	EF	1105		DFB	SEF	F("V")	
FFD1:	C4	1106		DFB	SC4	F("CTRL-K")	
FFD2:	EC	1107		DFB	SEC	F("S")	
FFD3:	A9	1108		DFB	SA9	F("CTRL-P")	
FFD4:	BB	1109		DFB	SBB	F("CTRL-B")	
FFD5:	A6	1110		DFB	SA6	F("-")	
FFD6:	A4	1111		DFB	SA4	F("+")	
FFD7:	06	1112		DFB	S06	F("M") (F=EX-OR S00+S89)	
FFD8:	95	1113		DFB	S95	F("<")	
FFD9:	07	1114		DFB	S07	F("N")	
FFDA:	02	1115		DFB	S02	F("I")	
FFDB:	05	1116		DFB	S05	F("L")	
FFDC:	F0	1117		DFB	SF0	F("W")	
FFDD:	00	1118		DFB	S00	F("G")	
FFDE:	EB	1119		DFB	SEB	F("R")	
FFDF:	93	1120		DFB	S93	F(":")	
FFE0:	A7	1121		DFB	SA7	F(".")	
FFE1:	C6	1122		DFB	SC6	F("CR")	
FFE2:	99	1123		DFB	S99	F(BLANK)	
FFE3:	B2	1124	SUBTBL	DFB	#BASCONT-1		
FFE4:	C9	1125		DFB	#USR-1		
FFE5:	BE	1126		DFB	#REGZ-1		

1. $X = 0$, falls keine Hexadezimalzahl eingegeben wurde.
2. Sprung zum Monitor, falls nichts gefunden wurde.
3. Kommandozeichen mit der Kommandotabelle vergleichen.
4. Aufruf des zugehörigen Unterprogramms, falls das Kommando in der Tabelle gefunden wurde.
5. Schiebt die erhaltene Hexadezimalziffer nach A2.
6. Beim Aufruf von DIG wird X den Wert \$FF erhalten.
7. Wenn der Monitor-Zustand 0 ist, wird A2 nach A1 und A3 gebracht.
8. Löscht A2.
9. Holt ein Zeichen.
10. Sprung, falls eine Hexadezimalziffer vorliegt.
11. Der höherwertige Adreßteil der Unterprogramme wird in den Keller gebracht.
12. Der niederwertige Adreßteil aus der Tabelle wird auch in den Keller gebracht.
13. Löscht den Monitor-Modus.
14. Sprung zum Unterprogramm über RTS.

Adreßbuch (numerisch geordnet)

0000	LOC0	004E	RNDL	F83C	CLRSC3	FAA6	PWRUP
0001	LOC1	004F	RNDH	F847	GBASCALC	FAA9	SETPG3
0020	WNDLFT	0095	PICK	F856	GBCALC	FAAB	SETPLP
0021	WNDWDTH	0200	IN	F864	SETCOL	FABA	SLOOP
0022	WNDTOP	03F0	BRKV	F871	SCRN	FAC7	NXTBYT
0023	WNBOTM	03F2	SOFTVEV	F879	SCRN2	FAD7	REGDSP
0024	CH	03F4	PWREDUP	F87F	RTMSKZ	FADA	RGDSP1
0025	CV	03F5	AMPERV	F882	INSDS1	FAE4	RDSP1
0026	GBASL	03F8	USRADR	F88C	INSDS2	FAFD	PWRCON
0027	GBASH	03FB	NMI	F89B	IEVEN	FB02	DISKID
0028	BASL	03FE	IRQLOC	F8A5	ERR	FB09	TITLE
0029	BASH	0400	LINE1	F8A9	GETPMT	FB11	XLTBL
002A	BAS2L	07F8	MSLOT	F8BE	MNNDX1	FB19	RTBL
002B	BAS2H	C000	IOADR	F8C2	MNNDX2	FB1E	PREAD
002C	H2	C000	KBD	F8C9	MNNDX3	FB25	PREAD2
002C	LMNEM	C010	KBDSTRB	F8D0	INSTDSF	FB2E	RTS2D
002D	V2	C020	TAPEOUT	F8D4	PRNTOP	FB2F	INIT
002E	RMNEM	C030	SPKR	F8DB	PRNTBL	FB39	SETTXT
002E	FORMAT	C050	TXTCLE	F8F5	NXTCOL	FB40	SETGR
002E	MASK	C051	TXTSET	F8F9	PRMN2	FB4B	SETWND
002E	CHKSUM	C052	MXCLER	F910	PRADR1	FB5B	TABV
002F	LASTIN	C053	MXSET	F914	PRADR2	FB60	APPLEII
002F	LENGTH	C054	LOWSCR	F926	PRADR3	FB65	STITLE
002F	SIGN	C055	HISCR	F92A	PRADR4	FB6F	SETPWRC
0030	COLOR	C056	LORES	F930	PRADR5	FB78	VIDWAIT
0031	MODE	C057	HIRES	F938	RELADR	FB88	KBDWAIT
0032	INVFLG	C058	SETAN0	F940	PRNTXY	FB94	NOWAIT
0033	PROMPT	C059	CLRAN0	F941	PRNTAX	FB97	ESCOLD
0034	YSAV	C05A	SETAN1	F944	PRNTX	FB9B	ESCNOW
0035	YSAV1	C05B	CLRAN1	F948	PRBLNK	FBA5	ESCNEW
0036	CSWL	C05C	SETAN2	F94A	PRBL2	FBC1	BASCALC
0037	CSWE	C05D	CLRAN2	F94C	PRBL3	FBD0	BASCLC2
0038	KSWL	C05E	SETAN3	F953	PCADJ	FBD9	BELL1
0039	KSWH	C05F	CLRAN3	F954	PCADJ2	FBE4	BELL2
003A	PCL	C060	TAPEIN	F956	PCADJ3	FBF0	RTS2B
003B	PCH	C064	PADDL0	F95C	PCADJ4	FBF0	STORADV
003C	A1L	C070	PTRIG	F961	RTS2	FBF4	ADVANCD
003D	A1H	CFFF	CLRR0M	F962	FMT1	FBFC	RTS3
003E	A2L	E000	BASIC	F9A6	FMT2	FBFD	VIDOUT
003F	A2H	E003	BASIC2	F9B4	CHAR1	FC10	BS
0040	A3L	F800	PLOT	F9BA	CHAR2	FC1A	UP
0041	A3H	F80C	RTMASK	F9C0	MNEML	FC22	VTAB
0042	A4L	F80E	PLOT1	FA00	MNEMR	FC24	VTABZ
0043	A4H	F819	HLINE	FA40	IRQ	FC2B	RTS4
0044	A5L	F81C	HLINE1	FA4C	BREAK	FC2C	ESC1
0045	A5H	F826	VLINEZ	FA59	OLDBRK	FC42	CLREOP
0045	ACC	F828	VLINE	FA62	RESET	FC46	CLEOP1
0046	XREG	F831	RTS1	FA6F	INITAN	FC58	HOME
0047	YREG	F832	CLRSCR	FA81	NEWMON	FC62	CR
0048	STAATUS	F836	CLRTOP	FA9B	FIXSEV	FC66	LF
0049	SPNT	F838	CLRSC2	FAA3	NOFIX	FC70	SCROLL

FC76	SCRL1	FD67	GETLNZ	FE22	LT2	FEDA	WR1
FC8C	SCRL2	FD6A	GETLN	FE2C	MOVE	FEED	WRBYTE
FC95	SCRL3	FD71	BCKSPC	FE36	VFY	FEFF	WRBYT2
FC9C	CLREOL	FD75	NXTCHAR	FE58	VFYOK	FEF6	CRMON
FC9E	CLEOLZ	FD7E	CAPTST	FE5E	LIST	FEFD	READ
FCA0	CLEOL2	FD84	ADDINP	FE63	LIST2	FF0A	RD2
FCA8	WAIT	FD8E	CROUT	FE75	A1PC	FF16	RD3
FCA9	WAIT2	FD92	PRA1	FE78	A1PCLP	FF2D	PRERR
FCAA	WAIT3	FD96	PRYX2	FE7F	A1PCRTS	FF3A	BELL
FCB4	NXTA4	FDA3	XAM8	FE80	SETINV	FF3F	RESTORE
FCBA	NXTA1	FDAJ	MOD8CHK	FE84	SETNORM	FF44	RESTR1
FCC8	RTS4B	FDB3	XAM	FE86	SETIFLG	FF4A	SAVE
FCC9	HEADR	FDB6	DATAOUT	FE89	SETKBD	FF4C	SAV1
FCD6	WRBIT	FDC5	RTS4C	FE8B	INPORT	FF59	OLDRST
FCD8	ZERDLY	FDC6	XAMPM	FE8D	INPRT	FF65	MON
FCE2	ONEDLY	FDD1	ADD	FE93	SETVID	FF69	MONZ
FCE5	WRTAPE	FDDA	PRBYTE	FE95	OUTPORT	FF73	NXTITM
FCEC	RDBYTE	FDE3	PRHEX	FE97	OUTPRT	FF7A	CHRSRCH
FCEE	RDBYT2	FDE5	PRHEXZ	FE9B	IOPRT	FF8A	DIG
FCFA	RD2BIT	FDED	COUT	FEA7	OIPRT1	FF90	NXTBIT
FCFD	RDBIT	PDF0	COUT1	FEA9	IQPRT2	FF9B	NXTBAS
FD0C	RDKEY	PDF6	COUTZ	FEB0	XBASIC	FFA2	NXTBS2
FD1B	KEYIN	FE00	BL1	FEB3	BASCONT	FFA7	GETNUM
FD21	KEYIN2	FE04	BLANK	FEB6	GO	FFAD	NXTCHR
FD2F	ESC	FE0B	STOR	FEBF	REGZ	FFBE	TOSUB
FD35	RDCHAR	FE17	RTS5	FEC2	TRACE	FFC7	ZMODE
FD3D	NOTCR	FE18	SETMODE	FEC4	STPZ	FFCC	CHRTBL
FD5F	NOTCR1	FE1D	SETMDZ	FECA	USR	FFB3	SUBTBL
FD62	CANCEL	FE20	LT	FECD	WRITE		

Adreßbuch (alphabetisch geordnet)

003D	A1H	E003	BASIC2	FC9C	CLREOL	FA9B	FIXSEV
003C	A1L	0028	BASL	FC42	CLREOP	F962	FMT1
FE75	A1PC	FD71	BCKSPC	CFF4	CLRR0M	F9A6	FMT2
FE78	A1PCLP	FF3A	BELL	F838	CLRSC2	002E	FORMAT
FE7F	A1PCRTS	FBD9	BELL1	F83C	CLRSC3	F847	GBASCALC
003F	A2H	FBE4	BELL2	F832	CLRSCR	0027	GBASH
003E	A2L	FE00	BL1	F836	CLRTOP	0026	GBASL
0041	A3H	FE04	BLANK	0030	COLOR	F856	GBCALC
0040	A3L	FA4C	BREAK	FDED	COUT	F8A9	GETMFT
0043	A4H	03F0	BRKV	PDF0	COUT1	FD6A	GETLN
0042	A4L	FC10	BS	PDF6	COUTZ	FD67	GETLNZ
0045	A5H	FD62	CANCEL	FC62	CR	FFA7	GETNUM
0044	A5L	FD7E	CAPTST	FEF6	CRMON	FEB6	GO
0045	ACC	F9B4	CHAR1	FD8E	CROUT	002C	H2
FDD1	ADD	F9BA	CHAR2	0037	CSWH	FCC9	HEADR
FD84	ADDINP	002E	CHKSUM	0036	CSWL	C057	HRES
FBF4	ADVANCE	FF7A	CHRSRCH	0025	CV	0055	HISCR
03F5	AMPERV	FFCC	CHRTBL	FDB6	DATAOUT	F819	HLINE
FB00	APPLEII	0024	CH	FF8A	DIG	F81C	HLINE1
002B	BAS2H	FCA0	CLEOL2	FB02	DISKID	FC58	HOME
002A	BAS2L	FC9E	CLEOLZ	F8A5	ERR	F89B	EVEN
FBC1	BASCALC	FC46	CLEOP1	FD2F	ESC	0200	IN
FBD0	BASCLC2	C059	CLRAN0	FC2F	ESC1	FB2F	INIT
FEB3	BASCONT	C05B	CLRAN1	FBA5	ESCNEW	FA6F	INITAN
0029	BASH	C05D	CLRAN2	FB9B	ESCNOW	FE8B	INPORT
E000	BASIC	C05F	CLRAN3	FB97	ESCOLD	FE8D	INPRT

F882	INSDS1	FFA2	NXTBS2	FD35	RDCHAR	03F2	SOFT2V
F88C	INSDS2	FAC7	NXTBYT	FD0C	RDKEY	C030	SPKR
F8D0	INSTDSP	FD75	NXTCHAR	FAE4	RDSP1	0049	SPNT
0032	INVLFG	FFAD	NXTCHR	FEFD	READ	004B	STATUS
C000	IOADR	F8F5	NXTCOL	FAD7	REGDSP	FEC4	STEPZ
FE9B	IOPRT	FF73	NXTITM	FEFB	REGZ	FB65	STITLE
FEA7	IOPRT1	FA59	OLDBRK	F938	RELADR	FE0B	STOR
FEA9	IOPRT2	FF59	OLDRST	FA62	RESET	FBF0	STORADV
FA40	IRQ	FCE2	ONEDLY	FF3F	RESTORE	FFE3	SUBTL
03FE	IRQLOC	FE95	OUTPORT	FF44	RESTR1	FB5B	TABV
C000	KBD	FE97	OUTPRT	FADA	RGDSP1	C060	TAP2IN
C010	KBDSTRB	C064	PADDL0	002D	RMNEM	C020	TAPBOUT
FB88	KBDWAIT	F953	PCADJ	004F	RNDH	FB09	TITLE
FD1B	KEYIN	F954	PCADJ2	004E	RNDL	FBFE	TOSUB
FD21	KEYIN2	F956	PCADJ3	FB19	RTBL	FEC2	TRACE
0039	KSWH	F95C	PCADJ4	F80C	RTMASK	C050	TXTCCLR
0038	KSWL	003B	PCH	F87F	RTMSKZ	C051	TXTSET
002F	LASTIN	003A	PCL	F831	RTS1	FC1A	UP
002F	LENGTH	0095	PICK	F961	RTS2	FEC4	USR
FC66	LF	F800	PLOT	FBEF	RTS2B	03F8	USRADR
0400	LINE1	F80E	PLOT1	FB2E	RTS2D	002D	V2
FE5E	LIST	FD92	PRA1	FBFC	RTS3	FBFD	VIDOUT
FE63	LIST2	F910	PRADR1	FC2B	RTS4	FE36	VFY
002C	LMNEM	F914	PRADR2	FCC8	RTS4B	FE58	VFYOK
0000	LOC0	F926	PRADR3	FDC5	RTS4C	FB78	VIDWAIT
0001	LOC1	F92A	PRADR4	FE17	RTS5	F828	VLINE
C056	LORES	F930	PRADR5	FF4C	SAV1	FB26	VLINEZ
C054	LOWSCR	F94A	PRLB2	FF4A	SAVE	FC22	VTAB
FE20	LT	F94C	PRBL3	FC76	SCRL1	FC24	VTABZ
FE22	LT2	F948	PRBLNK	FC8C	SCRL2	FC8A	WAIT
002E	MASK	FDDA	PRBYTE	FC95	SCRL3	FC99	WAIT2
C052	MI XCLR	FB1E	PREAD	FB71	SCRN	FC9A	WAIT3
C053	MI XSET	FB25	PREAD2	F879	SCRN2	0023	WNBDM
F9C0	MNEML	FF2D	PRERR	FC70	SCROLL	0020	WNDLFT
FA00	MNEMR	FDE3	PRHEX	C058	SETAN0	0022	WNDTOP
FB8E	MNNDX1	FDE5	PRHEXZ	C05A	SETAN1	0021	WNDWDTH
F8C2	MNNDX2	F8F9	PRMN2	C05C	SETAN2	FED4	WR1
F8C9	MNNDX3	F941	PRNTAX	C05E	SETAN3	FGD6	WRBIT
FDAD	MOD8CHK	F8DB	PRNTBL	F864	SETCOL	FE0F	WRBYT2
0031	MODE	F8D4	PRNTP0	FB40	SETGR	FE0D	WRBYTE
FF65	MON	F944	PRNTP	FE86	SETIFL	FECD	WRITE
FF69	MONZ	F940	PRNTYX	FE80	SETINV	FCE5	WRTAPE
FE2C	MOVE	0033	PROMPT	FE89	SETKBD	FDB3	XAM
07F8	MSLOT	FD96	PRYX2	FE18	SETMODE	FDA3	XAMB
FA81	NEWMON	C070	PTRIG	FE1D	SETMDZ	FDC6	XAMPM
03FB	NMI	FAFD	PWRCON	FE84	SETNORM	FE00	XBASIC
FAA3	NOFIX	03F4	PWREDUP	FAA9	SETPG3	FB11	XLTLBL
FD3D	NOTCR	FAA6	PWRUP	FAAB	SETPLP	0046	XREG
FD5D	NOTCR1	FF0A	RD2	FB6F	SETPWRC	0047	YREG
FB94	NOWAIT	FCFA	RD2BIT	FB39	SETTXT	0034	YSAV
FCBA	NXTA1	FF16	RD3	FE93	SETVID	0035	YSAV1
FCB4	NXTA4	FCFD	RDBIT	FB4B	SETWND	FCDB	ZERDLY
FF98	NXTBAS	FCEE	RDBYT2	002F	SIGN	FFC7	ZMODE
FF90	NXTBIT	FCEC	RDBYTE	FABA	SLOOP		

6502 : Herstellerbezeichnung für den Mikroprozessor in Ihrem Apple.

Abtastzelle : (Scan line) : Ein einmaliges Überstreichen des Kathodenstrahls über die Bildfläche einer Kathodenstrahlröhre.

Adreßbus : Ein Satz Kabel (oder das Signal auf diesen Leitungen), um binär entschlüsselte Adressen vom Mikroprozessor zum Rest des Computers zu übertragen.

Adresse : Jedem Speicherplatz wird eine bestimmte Nummer zugeordnet. Beim Apple ist jede Adresse eine Zahl zwischen 0 und 65.535 (Hexadezimal : \$0000 bis \$FFFF).

Adressierbereich : Die Gesamtheit aller Speicherstellen, auf die der Mikroprozessor direkt zugreifen kann.

Adressieren : Auf eine Speicherstelle zugreifen.

Adressierungsart : Im 6502-Mikroprozessor gibt es 13 verschiedene Weisen, auf die meisten Speicherstellen zuzugreifen. Diese 13 Möglichkeiten, Adressen zu bilden, nennt man Adressierungsarten.

Analog : Analoge Messungen verwenden im Gegensatz zu digitalen Messungen kontinuierlich veränderbare physikalische Größen (wie Länge, Spannung oder Widerstand), um Werte darzustellen. Digitale Messungen gebrauchen präzise, begrenzte Größen (wie Vorhandensein oder Fehlen von Spannungen oder magnetischen Feldern), um Werte darzustellen.

AND : Logische UND-Funktion : Der Ausgang ist genau dann „an“, wenn alle Eingänge „an“ sind.

Anschlußbelegung : Eine Beschreibung der Funktionen aller Anschlüsse eines IC's.

APPLE : Eine wohlschmeckende Frucht. 2. Der Markenname Ihres „ persönlichen Computers“ (personal computer). 3. Apple Computer, Inc., Hersteller von persönlichen Computern.

ASCII : Abkürzung für „American Standard Code for Information Interchange“. Dieser Standard-Code (auch USASCII oder fälschlich ASC-II) weist jedem der 128 Zeichen aus Ziffern, Buchstaben, Sonderzeichen und Steuerzeichen einen einheitlichen Wert von 0 bis 127 zu.

Assembler : Ein Programm, das Befehlssymbole und Operanden oder symbolische Adressen einer Assemblersprache in Operationscode und Operand einer Maschinensprache umwandelt.

Assemblersprache : Eine Sprache mit ähnlicher Struktur wie Maschinensprache, aber aufgebaut auf Befehlssymbolen und symbolischen Adressen. Assemblerprogramme sind sehr viel leichter zu schreiben und zu begreifen als Programme in Maschinensprache.

Ausgabe : Eine Ausgabe besteht aus Daten, die vom Computer erzeugt wurden und die für die Außenwelt bestimmt sind.

BASIC : Abkürzung für „Beginner's All-Purpose Symbolic Instruction Code“. BASIC ist eine höhere Programmiersprache mit einer ähnlichen Struktur wie FORTRAN, aber etwas leichter zu lernen.

Befehl : Der kleinste Teil eines Programms, den ein Computer ausführen kann. In der 6502-Maschinensprache enthält ein Befehl ein, zwei oder drei Byte; in höheren Programmiersprachen können einzelne Befehle (die dann auch oft „Anweisungen“ genannt werden) aus vielen Zeichen bestehen.

Befehlscode : Ein numerischer (meist binärer) Maschinenbefehl.

Binär : Ein Zahlensystem mit zwei Ziffern (0 und 1), wobei jede Ziffer einer Binärzahl eine Potenz von 2 darstellt. Die meisten digitalen Computer arbeiten (tief in ihrem Inneren) binär. Ein binäres Signal kann leicht ausgedrückt werden durch die Anwesenheit oder das Fehlen einer physikalischen Größe, wie elektrisches Potential oder magnetische Feld.

Binäre Funktion : Eine Operation durch einen elektrischen Schaltkreis dargestellt, der einen oder mehrere Eingänge und nur einen Ausgang hat. Alle Ein- und Ausgänge sind binäre Signale (siehe AND, OR und Exclusive OR).

Bit : Abkürzung für „Binary digit“. Die kleinste Information, die ein Computer halten kann. Ein einzelnes Bit gibt nur „0“ oder „1“ als Wert an. Mehrere Bits können zu größeren Werten gruppiert werden (siehe Byte).

Bus : Ein Satz Kabel oder Leiterbahnen in einem Computer, die einen zusammenhängenden Satz von Daten transportieren oder die Daten, die sich gerade auf dem Bus befinden.

Byte : Basis-Einheit für das Maß eines Computer-Speichers. Ein Byte umfaßt gewöhnlich acht Bit. Somit kann es einen Wert zwischen 0 und 255 haben. Jedes ASCII-Zeichen kann in einem Byte dargestellt werden. Die Speicherstellen des Apple sind alle ein Byte groß und die Adresse jeder dieser Speicherstellen besteht aus zwei Bytes. In einem Byte werden die Bits von rechts nach links numeriert : Bit 0 bis Bit 7. Dabei hat Bit 0 die Wertigkeit 1 ($= 2^0$) und Bit 7 den Stellenwert 128 ($= 2^7$).

Call : Aufruf. Ein Befehl, der ein Unterprogramm aufruft. Dazu wird das gerade ausgeführte Programm oder Unterprogramm verlassen und ein neues begonnen, meist mit der Absicht, zu dem ursprünglichen Programm oder Unterprogramm zurückzukehren.

Code : Eine Methode, etwas in anderer Form darzustellen. Der ASCII-Code stellt Zeichen als Binärzahlen dar; die BASIC-Programmiersprache wandelt Lösungsverfahren in Befehle um. Von Code spricht man auch im Zusammenhang mit niederen Programmiersprachen.

Computer : Eink Gerät, das eine Anzahl Befehle erhalten und speichern kann und dann aufgrund dieser Befehle in einer vorher bestimmten und vorhersagbaren Weise arbeitet. Diese Definition schließt ein, daß sowohl die Befehle als auch die Daten, nach denen sich die Befehle richten, geändert werden können. Ein Gerät, dessen Befehle nicht verändert werden können, ist kein Computer.

Daten : Informationen jeglicher Art.

Disassembler : Ein Programm, das die Operationscodes der Maschinensprache in die Befehlssymbole und Operanden der Assemblersprache umwandelt. Ein Disassembler ist das Gegenteil eines Assemblers.

Eingabe : Daten, die von der Außenwelt in den Computer fließen.

Ein- / Ausgabe : Software oder Hardware, um Daten mit der Außenwelt auszutauschen.

Einsprungsadresse : Die Stelle eines Maschinensprache-Unterprogramms, die die erste ausführbare Anweisung des Unterprogramms enthält. Deshalb steht sie gewöhnlich am Anfang des Unterprogramms.

Exclusive OR : Ausschließendes ODER. Eine binäre Funktion, deren Wert „aus“ ist, falls alle Eingänge „an“ sind oder alle Eingänge „aus“ sind.

Farbsignal : Ein Signal, das Farbfernsehgeräte erkennen und in die farbigen Punkte umsetzen, die man auf dem Bildschirm sieht. Ohne das Farbsignal erscheinen alle Bilder schwarz-weiß.

Fehler : Engl. „bug“ = Wanze, Käfer. Ein Fehler bzw. Defekt. Ein Hardwarefehler ist eine physikalische oder elektrische Fehlfunktion oder ein Aufbaufehler. Ein Software-Fehler ist ein Fehler im Programm, entweder logischer (Denkfehler) oder typographischer (Tippfehler) Natur.

Fenster : Ein reserviertes Feld auf einer Anzeige, das für bestimmte Zwecke vorgesehen ist.

Format : Die physikalische Form, in der etwas erscheint. **Formatieren** : solch ein Format festlegen.

Graphik : Ein System, um graphische Objekte darzustellen.

Hardware : Die physikalischen Teile des Computers.

Hexadezimalzahl : Ein Zahlensystem, das Werte der Basis 16 durch die Ziffern 0 bis 9 und die Buchstaben A bis F darstellt. Jede hexadezimale Ziffer einer Hexadezimalzahl stellt eine Potenz von 16 dar. In diesem Handbuch wird allen Hexadezimalzahlen ein \$-Zeichen vorangestellt.

Höchstwertig : Engl. high-order. Das wichtigste Teil oder das höchstwertige Teil einer Menge von ähnlichen Teilen. Das „führende Bit“ (high-order bit) in einem Byte ist das Bit mit der höchsten Stellenwertigkeit.

Höhere Programmiersprache : Eine Sprache, die für Menschen verständlicher ist als für Maschinen.

Hz (Hertz) : Zyklen bzw. Schwingungen pro Sekunde. Ein Rad eines Fahrrades, das zwei Umdrehungen pro Sekunde schafft, läuft mit 2 Hertz. Der Mikroprozessor des Apple läuft mit 1.023.000 Hertz.

IC : siehe integrierter Schaltkreis.

Integrierter Schaltkreis : Eine kleine Oblade (kleiner als ein Fingernagel und ungefähr so dick) aus einem glasartigen Material (gewöhnlich Silikon), in die ein elektronischer Schaltkreis eigeätzt wurde. Ein IC kann zehn bis zehntausend einzelne elektronische Bauteile ersetzen. IC's werden gewöhnlich in DIL-Gehäusen untergebracht. Der Name IC wird manchmal sowohl für den Schaltkreis als auch für das Gehäuse gebraucht.

Interface (Schnittstelle) : Der Informationsaustausch zwischen zwei Geräten oder der Mechanismus, der diesen Austausch ermöglicht.

Interpreter : Ein Programm (normalerweise in Maschinensprache geschrieben), das eine höhere Programmiersprache versteht und ausführt.

Interrupt : Programmunterbrechung. Ein physikalischer Effekt läßt den Computer in ein spezielles Interrupt-Unterprogramm springen. Nach der Unterbrechungsbehandlung nimmt der Computer die Programmausführung ohne erkennbare Veränderungen in dem Unterbrochenen Programm wieder auf. Interrupts werden gebraucht, um dem Computer zu melden, daß ein bestimmtes Gerät Aufmerksamkeit verlangt.

K : Steht für die griechische Vorsilbe „Kilo“ und bedeutet „tausend“. Im Zusammenhang mit Computern steht „K“ gewöhnlich für 2^{10} (= 1.024 oder hexadezimal \$400).

Kalstart : Die startphase eines Computers, der gerade angestellt wurde.

Keller : Eng. stack. Ein reservierter Speicherbereich, im zeitweise Informationen zu speichern. Der Zugriff auf Daten im Keller geschieht nicht über Adressen, sondern in der Reihenfolge, in der diese Daten in den Keller gelangt sind. Die Information, die zuletzt in den Keller gegeben wurde, wird die erste sein, die aus dem Keller geholt wird.

Kilobyte : 1 K Bytes = 1.024 Bytes.

Kommando : Eine bestimmte Information zur Steuerung eines Computers (oder eines Programms). Die Gesamtheit aller Kommandos, die dem Benutzer das Bedienen eines Computers und seiner Peripherie ermöglicht, heißt Kommandosprache.

Leiterbahn : Ein elektrisch leitender Pfad auf einer gedruckten Schaltung, der Bauteile elektrisch miteinander verbindet.

Maschinensprache : Die Sprache auf dem niedrigsten Niveau, die ein Computer versteht. Maschinensprachen sind gewöhnlich binär. Befehle in Maschinensprache sind ein-Byte-Befehls-codes, denen manchmal verschiedene Operanden folgen.

Mikrocomputer : Ein Computer auf der Basis eines Mikroprozessors.

Mikroprozessor : Ein integrierter Schaltkreis, der Maschinenprogramme versteht und ausführt.

Modulo : Eine arithmetische Operation mit zwei Operanden. Modulo teilt den ersten Operanden durch den zweiten und ergibt sich dann als Rest der Division.

Modus : Eine Bedingung oder mehrere Bedingungen, unter denen bestimmte Regeln angewandt werden.

Monitor : 1. Ein Fernsehempfänger mit direktem Video-Eingang. 2. Ein Programm, bei dem Sie auf sehr niedrigem Niveau mit Ihrem Computer arbeiten (meistens mit den Adressen und Inhalten einzelner Speicherstellen).

Multiplexer : Ein elektronischer Schaltkreis mit vielen Dateneingängen, ein paar Auswahleingängen und einem Ausgang, der genau einen der vielen Dateneingänge mit dem einzigen Ausgang verbindet. Die Auswahl des Dateneingangs, der an den Ausgang geschaltet wird, wird durch die Auswahleingänge bestimmt.

Niedere Programmiersprache : Eine Sprache, die für Maschinen verständlicher ist, als für Menschen.

Niederwertig : Engl. low-order. Das unbedeutendste Teil oder das Teil mit dem geringsten Wert einer Menge von ähnlichen Teilen. Das niederwertigste Bit (low-order bit) eines Bytes ist das Bit mit der geringsten Stellenwertigkeit.

Operand : Objekt eines Befehls oder einer Anweisung. Im Befehl wird der Operand meist durch seine Adresse gekennzeichnet.

Or (ODER) : Eine binäre Funktion, deren Ausgang „an“ ist, wenn wenigstens einer der Eingänge „an“ ist.

Pascal : 1. Ein bekannter französischer Wissenschaftler. 2. Eine höhere Programmiersprache.

Peripherie : Mit dem Computer verbundene Geräte, die nicht Teil des Computers sind. Die meisten peripheren Geräte sind Ein-/Ausgabegeräte.

Personal Computer („Persönlicher“ Computer) : Ein Computer mit Speicher, Sprachen und Peripherie, der sich gut für den Gebrauch zuhause, im Büro, in Wissenschaft und Technik oder in der Schule eignet.

Platine : Gedruckte Schaltung. Eine Platte aus Fiberglas oder Epoxyd mit einem dünnen Metallbelag, der teilweise so weggeätzt wurde, daß Leiterbahnen zurückblieben. Elektronische Bauteile können mit Lötzinn befestigt werden und können dann elektronische Signale über die geätzten Leiterbahnen austauschen. Kleine Platinen werden oft „Karten“ genannt, besonders, wenn sie durch Steckerleisten angeschlossen werden.

Potentiometer : Ein elektronisches Bauteil, dessen elektrischer Widerstand (Widerstand gegenüber dem Elektronenfluß) proportional zur Einstellung einer Skala oder eines Knopfes ist. Auch bekannt als „Poti“ oder „Trimmer“.

Programm : Eine Folge von Befehlen, die einen Prozess beschreiben.

Programmschalter : Engl. soft switch. Ein Schalter mit zwei Stellungen, der von der Software eines Computers bedient werden kann.

PROM : Engl. Programmable Read-Only Memory. Ein PROM ist ein Festwertspeicher (ROM), dessen Inhalte elektrisch geändert werden können. Die Informationen in einem PROM verschwinden nicht, wenn der Strom abgeschaltet wird. Einige PROM's (sog. EPROM's) können mit UV-Licht gelöscht werden und sind dann wieder neu programmierbar.

Puffer : Ein Gerät oder ein Speicherbereich, um etwas zeitweilig zu speichern. Der „Bildpuffer“ enthält graphische Informationen, die auf dem Bildschirm angezeigt werden. Der „Eingabepuffer“ enthält eine schon teilweise geformte Eingabezeile.

RAM : Engl. Random-Access Memory. Der Hauptspeicher des Computers. Die Abkürzung RAM bezieht sich auf die integrierten Schaltkreise dieses Speichertyps und auf den Hauptspeicher selbst.

ROM : Eng. Read-Only Memory. Festwertspeicher. Dieser Speichertyp wird meist gebraucht, um wichtige Programme oder Daten zu halten, die beim Einschalten des Computers verfügbar sein müssen. Die Informationen in ROM's sind schon im Produktionsprozess festgelegt und nicht veränderbar. Bei ausgeschaltetem Gerät verlieren ROM's ihre gespeicherten Informationen nicht.

Rücksprung RETURN : Ein Unterprogramm verlassen und zu dem Programm zurückkehren, das es aufgerufen hatte.

Schaltplan : Ein Diagramm, das die elektrischen Verbindungen und die Schaltung eines elektronischen Gerätes darstellt.

Schnittstelle : Siehe Interface.

Seite : Engl. page. 1. Ein Speicherbereich, der mit einem Byte adressierbar ist. Auf dem Apple enthält eine „Seite“ im Speicher 256 Speicherstellen. 2. Ein Bildschirm voll Informationen.

Software : Die Programme, die der Hardware etwas zu tun geben.

Speicheradresse : Eine Speicheradresse ist ein Zwei-Byte-Wert, der eine einzelne Speicherstelle aus dem gesamten Adressierbereich auswählt. Speicheradressen werden im Apple mit ihrem niederwertigen Byte (low-order byte) zuerst und dann mit ihrem höherwertigen Byte (high-order byte) abgespeichert.

Speicherbereich : Siehe Adressierbereich.

Speicherstelle : Die kleinste Unterteilung des gesamten Speicherbereichs, auf die der Computer zugreifen kann. Jede Speicherstelle wird mit einer anderen Adresse und einem bestimmten Wert verknüpft. Die Speicherstellen des Apple enthalten jeweils ein Byte.

Sprache : Eine Computersprache ist ein Code, den (hoffentlich!) sowohl der Programmierer als auch der Computer versteht. Der Programmierer drückt in diesem Code das aus, was er tun möchte und der Computer versteht den Code und führt die gewünschten Aktionen aus.

Steuerzeichen : Im Apple meist in Verbindung mit **CTRL**. Zeichen im ASCII-Zeichensatz, die gewöhnlich keine graphische Darstellung haben, aber gebraucht werden, um verschiedene Funktionen zu kontrollieren. Das **RETURN** -Steuerzeichen ist ein Signal für den Apple, daß Sie eine Eingabezeile fertig getippt haben und wünschen, daß sich der Computer damit beschäftigt.

Symbolische Adresse : In Assemblersprache werden gerne für die Operanden Namen statt bloßer Hexadenzimalzahlen verwendet. Jeder Name steht dann für eine ganz bestimmte Hexadenzimalzahl. Allerdings erfordert diese verständlichere Schreibweise (siehe Anhang C) einen sehr aufwendigen Assembler.

Syntax : Die Befehlsstruktur in einer bestimmten Sprache. Falls Sie bei der Eingabe eines Befehls die Syntax verletzen, nennt der Computer dies manchmal einen „SYNTAX ERROR“ (Syntax-Fehler).

Text : Zeichen, gewöhnlich Buchstaben und Zahlen. „Text“ bezieht sich meist auf beliebige Texte in Deutsch oder auch in Computersprache.

Unterprogramm : Engl. subroutine. Ein Programmteil, der für einen einzigen Aufruf ausgeführt wird. Unterprogramme werden gebraucht, um dieselbe Befehlsfolge an verschiedenen Stellen eines Programms auszuführen.

Warmstart : Den Computer neu starten, nachdem Sie die Kontrolle über sein Sprach- oder Operationssystem verloren haben.

Wechselschalter : Engl. toggle switch. Ein Schalter mit zwei Positionen, der zwar zwischen den beiden Stellungen hin und her wechselt, aber nicht direkt auf eine bestimmte Position gesetzt werden kann.

Zeichen : Jedes graphische Symbol, das für uns eine bestimmte Bedeutung hat. Dazu gehören Groß- und Kleinbuchstaben, Zahlen und verschiedene Symbole (wie Satzzeichen und Sonderzeichen).

Zeiger : Engl. cursor. Ein Spezialsymbol, das auf eine bestimmte Position hinweist. Der blinkende Zeiger auf dem Bildschirm des Apple erinnert Sie daran, wo Sie gerade beim Schreiben sind.

Zeile : Auf dem Bildschirm ist eine „Zeile“ eine waagrechte Aufeinanderfolge von graphischen Zeichen, die sich von dem einen Ende des Bildschirms zu dem anderen ausdehnt.

Zugriff : Die Inhalte einer Speicherstelle prüfen oder ändern.

Synertek/MOS Technology 6500 Programming Manual

Dieses Handbuch ist eine Einführung für die Programmierung des MC6502 Mikroprozessors in Maschinensprache. Es beschreibt die Arbeitsweise dieses Mikroprozessors in aller Ausführlichkeit, bezieht sich jedoch nicht auf die Besonderheiten des Apple. Es ist bei Apple, Inc. unter Bestellnummer A2L0003 zu beziehen.

Synertek/MOS Technology 6500 Hardware Manual

Dieses Handbuch enthält eine detaillierte Beschreibung der internen Arbeitsweise des 6502 Mikroprozessors. Sie finden darin viele nützliche Informationen, wie Sie periphere Geräte an den Mikroprozessor des Apple anschließen können. Das Buch ist bei Apple, Inc. unter Bestellnummer A2L0002 zu beziehen.

The Apple II Monitor Peeled

Dieses Buch enthält eine ausgezeichnete, erschöpfende Darstellung der Monitor-Unterprogramme des Apple Monitor ROM. Zu beziehen beim Autor: William E. Dougherty, 14349 San Jose Street, Los Angeles, CA 91345.

Programming the 6502

Dieses Buch von Rodnay Zaks ist ein ausgezeichnetes Handbuch für Maschinen- und Assembler-Programmierung des Apple 6502 Mikroprozessors. Es ist bei Sybex Inc., 2020 Milvia, Berkeley, CA 94704, erhältlich. Außerdem ist es bei Ihrem Apple-Händler unter Bestellnummer C202 erhältlich.

6502 Applications

Dieses Buch — ebenfalls von Rodnay Zaks — beschreibt viele Anwendungen des Apple 6502 Mikroprozessors. Es ist unter Bestellnummer D302 bei Sybex Inc. erhältlich.

System Description : The Apple II

Der Konstrukteur des Apple, Steve Wozniak, beschreibt in diesem Artikel die grundlegende Konstruktion und Arbeitsweise des Apple II. Der in der Mai-Ausgabe 1977 des BYTE-Magazins erschienene Artikel kann bei BYTE Publications, Inc., Peterborough, NH 30458, bezogen werden.

SWEET 16 : The 6502 Dream Machine

Steve Wozniak beschreibt in diesem Artikel die Maschinensprache des SWEET16-Interpreters im Apple Integer BASIC ROM. Der Artikel erschien in der Oktober-Ausgabe 1977 des BYTE-Magazins und ist ebenfalls bei BYTE Publications, Inc., Peterborough, NH 30458, erhältlich.

More Colors for your Apple

Dieser Artikel von Allen Watson III beschreibt detailliert die Apple Hi-Res-Graphik. Er enthält auch eine Erläuterung von Steve Wozniak, dem Konstrukteur des Apple, in der er eine Modifikation beschreibt, mit der Sie Ihren Apple der Version 0 mit den auf der Apple-Platine der Version 1 verfügbaren Farben erweitern können. Dieser Artikel erschien in der Juni-Ausgabe des BYTE-Magazins 1979 und kann bei BYTE Publications, Inc., Peterborough, NH 30458, bezogen werden.

Call APPLE (Apple Puget Sound Program Library Exchange)

Dies ist die Zeitschrift einer der größten Apple-Benutzer-Gruppen. Wenden Sie sich an: Apple Puget Sound Program Library Exchange, 6708 39th Ave. Southwest, Seattle, Wash., 98136.

The Cider Press

Dies ist die Zeitschrift einer weiteren großen Apple-Benutzer-Gruppe: The Cider Press, c/o The Apple Core of San Francisco, Box 4816, San Francisco, CA 94101.

APPLE-COM-POST

Dies ist die deutschsprachige Zeitschrift der APPLE USER GROUP EUROPE, Postfach 4068, D-4320 Hattingen 13.

Info

Die Club-Zeitschrift des Apple-Clubs in 4986 Rödinghausen, Wehmerhorststr. 110.

ABACUS

Herausgeber dieser Zeitschrift ist : Commandus, Umlandstr. 158, D-1000 Berlin 15.

INDEX

Stichwortverzeichnis	220
Photos	224
Tabellen	224
Tasten und Zeichen	225
Schaubilder	225

Stichwortverzeichnis

A

Adreßformate	68-69
Analog-Eingänge	25-26-102
Anschluß :	
Betriebsspannungen	107
Drucktasten	26
Kassettenrekorder	107
Lautsprecher	108
Netz	98
Peripherie	109
Potentiometer	26
Spielsteuerungen	25-103
Tastatur	106
Video	9-101
Apple II Plus	29
Assembler	53-68
Auffrischung	99
Aufteilungsstecker	74
Ausgabe-Stop	28
Ausgabe-Zeiger	32-33
Autostart-ROM	28-39-74

B

Befehlscodes	123
Befehlssatz	120
Bereitschaftszeichen	35-41
Bild löschen	37-38
Bildmodus-Schalter	12-13-102
Bildpuffer	10
Bildschirm	9
Bildspeicherbereiche	12
Blinkend	34
Block-Graphik	10-17

C

COU1	32-64-87-88
CSW	87
CTRL	6

D

Dateneingänge	82
Disassembeln	53
DOS	39-87

Drucktasten	25-26-102
Duale Eingänge	82

E

Ein-/Ausgabe	82-102
Ein-Bit-Ausgänge : Signal-Ausgänge	25
Ein-Bit-Eingänge	25-26-102
Eingabe-Zeiger	35
Einlesen von Daten	51
Einschalt-Byte	40
Erweiterungs-ROM	88
Escape-Zustand	37
EXAMINE	57

F

Farben	10-17-20-29
Farbsignalunterdrückung	13-28
Fehlerhilfen	28-55

G

Gemischte Anzeige	10-39
GETLN	35-65-66-74
GO	52

H

Hauptplatine	3
Hochauflösende Graphik	10-20

I

Impulsausgang	25-26-73-82-102
Interface	24
Invers	34-58

K

Kalt-Start	39
Kassetten-Ein-/Ausgang	24-102-107
KEYBOARD	59-88
KSW	87
Kurzschluß	98

L

Language Card	73
Lautsprecher	22-29-102-108
Lauwarmer Start	39
Linkspfeiltaste	36
LIST	53

M

Mikroprozessor	92
Mini-Assembler	53-68
Monitor	44
Monitor-ROM	28-41-74
MOVE	48

N

Netzteil	2-29-96
Normal	34-58

P

Peripherie-Anschlüsse	108
Peripherie-Ein-/Ausgabe	84
Peripherie-Zwischenspeicher	87
Potentiometer	26
PRINTER	58-87
Programmschalter	12-22-26-83
Punktmatrix	12

R

RAM	72-99
RDKEY	35-66
READ	51
Rechtspfeiltaste	36
Register	57-68-85-122
REPT	6-104-106
RESET	6-28-39-41
RETURN	32-35-44
ROM	72-76-84-98

S

Schreiben von Daten	50
Schreibvorgang	22
Seite	10-74
Seite Eins... Seite Sieben	74

Seite Null	78
Signal-Ausgänge	26
Speicher-Größe	75
Spiele-Anschluß	25
Spielsteuer-Rücksetzimpuls	82-102
Spielsteuerungen	25
Standard-Ausgabe	32
Standard-Eingabe	35
Starten	52
STEP	28-55

T

Taktsteuerung	92
Tastatur	4-102-104
Tastatur-Anschluß	106
Tastendruckspeicher	5
Tastenimpuls	5
Tastenlöschimpuls	6-82-102
Text	10-13
Text-Fenster	33
Ton	22-29-36-41-69
TRACE	28-55

U

USER 1	103
USER	61

V

Vergleichen von Bereichen	49
VERIFY	49
Version 0 / 1	28
Verzweigungsadressen	87
Video-Anschluß	9-101
Video-Generator	100

W

Warm-Start	39-40
Wechselschalter	22-83
Wire-Wrap	10
WRITE	50

Z

Zeichensatz	14
Zeitkreis	26
Zufallszahlen	35

Photos

Photo 1 : Der Apple II	2
Photo 2 : Die Rückseite des Apple-Netzteils	2
Photo 3 : Das Apple-Tastenfeld	4
Photo 4 : Video-Anschlüsse und Potentiometer	10
Photo 5 : Der Zeichensatz des Apple	14
Photo 6 : Ein-/Ausgabeanschluß für Spiele	25
Photo 7 : Die USER 1 Lötbrücke	103

Tabellen

Tabelle 1 : Adressen des Tastenfeldes	6
Tabelle 2 : Tasten und die zugehörigen ASCII-Codes	7
Tabelle 3 : Der ASCII Zeichensatz	8
Tabelle 4 : Bildschirm-Speicherbereiche	12
Tabelle 5 : Bildmodus-Schalter	12
Tabelle 6 : Bildmodus-Kombinationen	13
Tabelle 7 : ASCII Bildschirmzeichen	15
Tabelle 8 : Farben der Block-Graphik	17
Tabelle 9 : Adressen für die Signal-Ausgänge	26
Tabelle 10 : Ein-/Ausgabe-Adressen	27
Tabelle 11 : Speicherstellen für das Text-Fenster	34
Tabelle 12 : Ausgabekontrolle durch Speicherstelle 50	34
Tabelle 13 : Spezielle Adressen für den Autostart-ROM	40
Tabelle 14 : Monitor-Adressen auf Seite 3	68
Tabelle 15 : Mini-Assembler Adreßformate	69
Tabelle 16 : RAM-Aufteilung	73
Tabelle 17 : ROM-Aufteilung und Programme	76
Tabelle 18 : Belegung der Seite Null durch den Monitor	78
Tabelle 19 : Belegung der Seite Null durch Applesoft II BASIC	78
Tabelle 20 : Belegung der Seite Null durch Apple DOS 3.2	79
Tabelle 21 : Belegung der Seite Null durch Integer BASIC	79
Tabelle 22 : Eingebaute Ein-/Ausgabe-Adressen	83
Tabelle 23 : Ein-/Ausgabebereiche für Geräte-Anschlußkarten	84
Tabelle 24 : PROM-Bereiche für Ein-/Ausgabe-Gerätekarten	85
Tabelle 25 : Ein-/Ausgabe-Basisadressen	86
Tabelle 26 : Ein-/Ausgabe-Zwischenspeicher-Adressen	87

Anschlußbezeichnungen

Tabelle 27 : Beschreibung der Taktsignale	94
Tabelle 28 : Anschlüsse der Video-Vierstiftbuchse	101
Tabelle 29 : Beschreibung der Spielanschlußsignale	104
Tabelle 30 : Beschreibung des Tastaturanschlusses	106
Tabelle 31 : Beschreibung des Betriebsspannungsanschlusses	107

Tabelle 32 : Beschreibung des Lautsprecheranschlusses	108
Tabelle 33 : Signale an den Peripherie-Anschlüssen	110

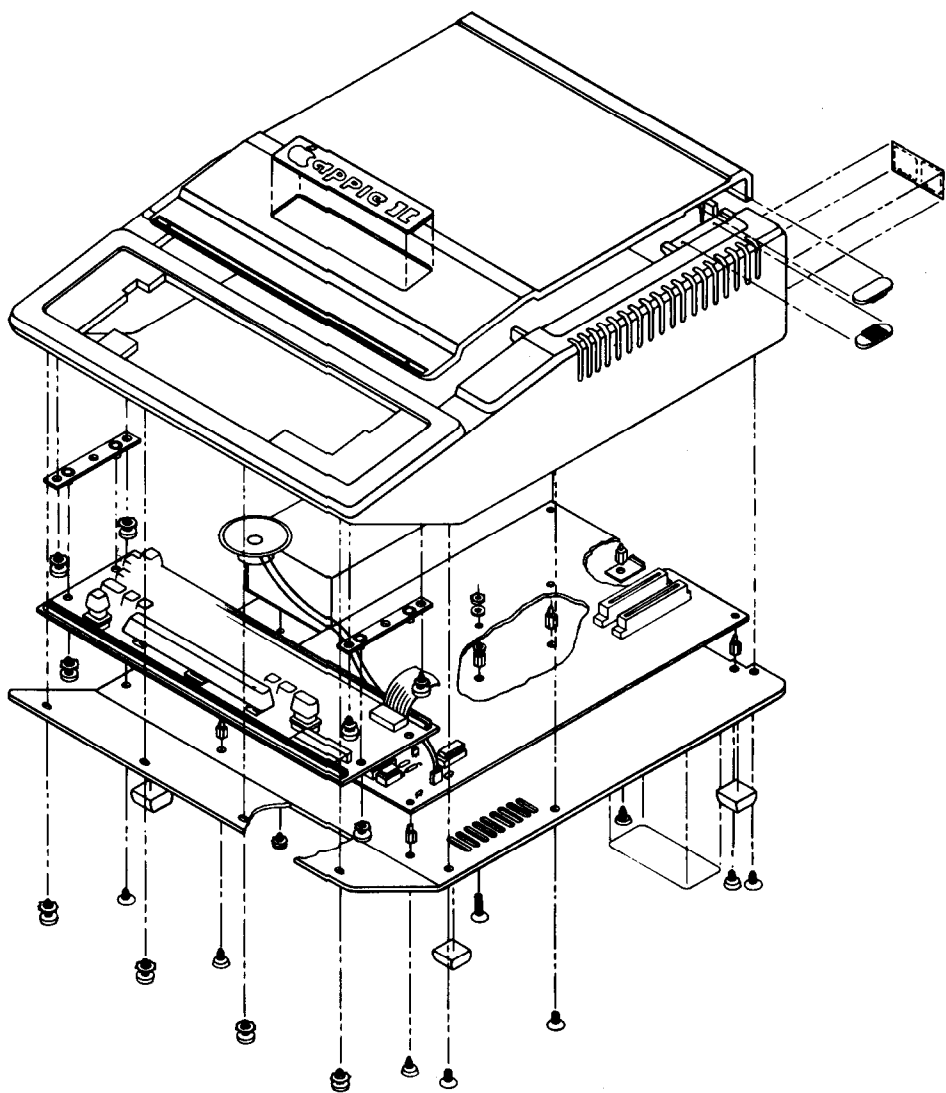
Tasten und Zeichen

!	35
#	69
\$	53-68-69
&	68
*	35-41-44
+	59
-	59
.	52
:	47-55-59
<	49
>	35
?	35
←	36-32
→	36
@	38
A	37
B	37
C	38
CTRL B	58
CTRL C	28-33-58
CTRL E	57
CTRL G (bell)	28-32-33
CTRL H (←)	36-32
CTRL J (Zeilenvorschub)	32
CTRL K	59-88
CTRL P	58-87
CTRL S	28-33
CTRL U (→)	36
CTRL X	28-36
CTRL Y	61
D	37-38
E	37-38
ESC	28-37
F	37-38
G	52
I	28-38-58-60
J	28-38
K	28-38
L	53
M	28-38-49-60
N	58-60
R	51
REPT	6-104-106
RESET	6-28-39-41
RETURN	32-35-44
S	55-60
SHIFT	6

T	56
V	49-61
W	50
[.....	8
\	8-36
]	35
^	54-69
—	8

Schaubilder

Schaubild 1 : Speicherplan des Text-Bildschirms	16
Schaubild 2 : Speicherplan der Block-Graphik	19
Schaubild 3 : Speicherplan für die hochauflösende Graphik	23
Schaubild 4 : Escape-Anweisungen für Zeigerbewegungen	38
Schaubild 5 : Aufteilung des Speichers in Funktionsbereiche	72
Schaubild 6 : Mögliche Speicheraufteilungen	75
Schaubild 7 : Anschlußdiagramm der Speicheraufteilungsstecker	75
Schaubild 8 : Freigabe-Schaltkreis für den Erweiterungs-ROM	89
Schaubild 9 : \$CFXX Dekodierung	89
Schaubild 10 : Die Hauptplatine des Apple	93
Schaubild 11 : Die Taktsignale und ihre Beziehung zueinander	95
Schaubild 12 : Schaltbild des Netzgerätes	97
Schaubild 13 : Anschlüsse des 9316B ROM	98
Schaubild 14 : Anschlüsse der RAM-IC's	99
Schaubild 15 : Video-Hilfsanschlüsse	101
Schaubild 16 : Spieleanschlußbelegung	103
Schaubild 17 : Schaltbild der Apple-Tastatur	105
Schaubild 18 : Tastenfeld-Anschlußbelegung	106
Schaubild 19 : Stecker zum Anschluß der Betriebsspannungen	107
Schaubild 20 : Anschluß des Lautsprechers	108
Schaubild 21 : Anschlußbelegung der Peripherie-Steckleisten	109
Schaubild 22 : Schaltplan des Apple II	113-118





PRINTED IN BELGIUM N° de produit Apple A2L0001A (030-0004-01) - D