

# TINY COMPILER

Auftruf: SYS \$6000, Startadresse

Autor: N. Heuer

06.09.1984 Einstieg

,6000 4C C8 63	JMP \$6300	um Hauptprogramm	,606F 60	RTS
,6003 A6 FD	LDX #\$FD		,6070 A5 5B	LDA \$FB
,6005 A4 A7	LDY #\$A7		,6072 36	SEC
,6007 A9 00	LDA #\$00		,6073 E5 FD	SBC \$FD
,6009 85 FE	STA \$FE		,6075 85 FE	STA \$FE
,600B 85 FF	STA \$FF		,6077 A5 FC	Subtraktion
,600D E0 00	CPX #\$00		,6079 E5 A7	LDA \$FC
,600F D0 04	BNE \$6015		,607B 85 FF	SBC \$A7
,6011 C0 00	CPY #\$00		,607D 60	STA \$FF
,6013 F0 16	BEQ \$602B	Multiplication	,607E 00	RTS
,6015 A5 FE	LDA \$FE		,607F 00	BRK
,6017 10	CLC		,6080 00	BRK
,6018 45 FB	ADC \$FB			
,601A 85 FE	STA \$FE		,6080 空白 Zeile ENDE LE	
,601C A5 FF	LDA \$FF		,60A0 TZTES BYTE BEI UNFOUND ILLEGAL BY	
,601E 45 FC	ADC \$FC		,60C0 TE PROTOKOLLIEREN ? [J/N] PA	
,6020 85 FF	STA \$FF		,60E0 SS 2M SUCHE ZEILE	
,6022 CA	DEX		,6100 TINYCOMPILING PASS 1	
,6023 E0 FF	CPX #\$FF		,6120 DRUN STOP ERLAUBEN ? [J/N] MI	
,6025 D0 E6	BNE \$600D		,6140 SSING LET/INVALID STATEMENT/ILLEGAL	
,6027 88	DEY		,6160 L PARAMETER WRONG CHAR/MISSING	
,6029 18	CLC		,6180 CHAR LINE NOT FOUND/PROGRAM/ILLEGAL	
,602A 90 E2	BCC \$600D		,61A0 PSEUDO	
,602B 60	RTS			
,602C A2 FF	LDX #\$FF			
,602E 84 FF	STX #\$FF			
,6030 84 FE	STX #\$FE		,61AB 61 49	fehlermeldungs- releasen
,6032 E4 FE	INC #\$FE		,61AA 61 5A	
,6034 D0 02	BNE \$6038		,61AO 61 6B	
,6036 E6 FC	INC #\$FF		,61AE 61 75	
,6038 A5 FB	LDA \$FB		,61B0 61 78	
,603A 38	SEC	Division	,61B2 61 84	
,603B E5 FD	SBC \$FD		,61B4 61 92	
,603D 85 FB	STA \$FB		,61B6 61 97	
,603F A5 FC	LDA \$FC		,61B8 61 CA	
,6041 E5 A7	SBC \$A7		,61BA 9A	
,6043 85 FC	STA \$FC		,61BB 0A	
,6045 B0 EB	BCS \$6032		,61BC AA	
,6047 60	RTS		,61BD BD A7 61	
,6048 A5 FD	LDA \$FD		,61C0 85 22	
,604A D0 E0	BNE \$602C		,61C2 BD A8 61	
,6052 A5 A7	LDA \$A7		,61C5 4C 45 A4	
,604E D0 DC	BNE \$602C		,61C8 8D 10 00	
,6050 A9 FF	LDA #\$FF		,61CB EE C9 61	
,6052 85 FE	STA \$FE		,61CE DO 03	
,6054 85 FF	STA \$FF		,61DO EE CA 61	
,6056 60	RTS		,61D3 60	
,6057 09 00	LDA #\$00		,61D4 20 FD AE	Parameter komma
,6059 A2 34	LDX #\$34		,61D7 20 8A AD	JSR \$AD8A
,605B 9D FF FF	STA \$FFFF, X	CLR	,61DA 20 F7 JSR \$B7F7	16-Bit Werte holen
,605E CA	DEX		,61DD A5 14	LDA \$14
,605F D0 FA	BNE \$605B		,61DF A4 15	LDY \$15
,6061 60	RTS		,61E1 BD C9 61	STA \$61C9
,6062 A5 FB	LDA \$FB		,61E4 BC CA 61	STY \$61CA
,6064 18	CLC		,61E7 60	RTS
,6065 45 FD	ADC \$FD		,61E8 A2 00	LDX #\$00
,6067 85 FE	STA \$FE	Addition	,61EA BD 03 60	LDA \$6003, X Werte holen
,6069 A5 FC	LDA \$FC		,61ED 20 C8 61	JSR \$61CB und ablegen
,606B 65 A7	ADC \$A7		,61FO EB	INX
,606D 85 FF	STA \$FF		,61F1 EO 7B	CPX #\$7B schon alle Bytes?

,61F3 D0 F5 BNE \$61EA *nach davor* 6277 0A ASL  $\rightarrow A=0$   
 ,61F5 60 RTS 6278 18 CLC  $B=2$   
 ,61F6 A5 15 LDA \$15 High-Byte Startadresse 6279 6D 89 60 ADC #\$0B9  
 ,61FB 8D 80 60 STA \$6080 627C AA TAX  
 ,61FB 8D 82 60 STA \$6082 627D A9 00 LDA #\$00  
 ,61FE 8D 84 60 STA \$6084 627F 6D BA 60 ADC \$608A  
 ,6201 8D 86 60 STA \$6086 6282 AB TAY  
 ,6204 8D 88 60 STA \$6088 6283 18 CLC Flag: Variable  
 ,6207 A5 14 LDA \$14 Low-Byte Startadresse 6284 60 RTS  
 ,6209 18 CLC 6285 A5 Zahl LDA \$14  
 ,620A 69 03 ADC #\$03 +3 6287 48 PHA Programmstart  
 ,620C 8D 83 60 STA \$6083 ablegen 6288 A5 15 und Stapel  
 ,620F 18 CLC 628A 48 PHA  
 ,6210 69 45 ADC #\$45 + \$45 628B A5 7A LDA \$7A Chipt-Zeiger  
 ,6212 8D 85 60 STA \$6085 ablegen 628D 48 628E A5 7B auf Stapel  
 ,6215 18 CLC 6290 48 PHA  
 ,6216 69 0F ADC #\$0F + 15 6291 AD 34 62 LDA \$6234 Text-Zeiger low  
 ,6218 8D 87 60 STA \$6087 ablegen 6294 38 SEC -1  
 ,621B 18 CLC 6295 E9 01 SBC #\$01  
 ,621C 69 0B ADC #\$0B + 11 ablegen 6297 85 7A STA \$7A Chipt-low  
 ,621E 8D 7F 60 STA \$607F 6299 AD 35 62 LDA \$6235 Text-Zeiger high  
 ,6221 18 CLC 629C E9 00 SBC #\$00 übertrag  
 ,6222 69 0E ADC #\$0E + 14 629E 85 7B STA \$7B Chipt-High  
 ,6224 8D 81 60 STA \$6081 ablegen 62A0 20 79 JSR \$0079 Chipt  
 ,6227 60 RTS 62A3 20 F3 BC JSR \$BCF3 ASCI->Fließkomm  
 ,6228 A2 34 LDX #\$34 JSR von 647A 62A6 20 1B BC JSR \$BC1B FAC runden  
 ,622A A9 00 LDA #\$00 52 62A9 20 F7 B7 JSR \$B7F7 FAC > 16 Bit  
 ,622C 20 08 61 JSR \$61C8 Nullen im 62AC A5 7B LDA \$7B  
 ,622F CA DEX 62AD 8D 35 62 STA \$6235 neuen Zeiger  
 ,6230 D0 FA BNE \$622E 62AE 8D 35 62 STA \$7A zurückspeichern  
 ,6232 40 RTS 62B1 A5 7A LDA \$6234  
 ,6233 AD 09 00 LDA \$0009 643F Byte Werte 62B3 8D 34 62 STA \$6234  
 ,6236 EE 34 62 INC \$6234 62B6 68 PLA  
 ,6239 D0 03 BNE \$623E 62B7 85 7B STA \$7B Chipt-Zeiger  
 ,623B EE 35 62 INC \$6235 62B9 68 PLAs wiederherstellen  
 ,623E 24 AB BIT \$AB 62BA 85 7A STA \$7A Ergebnis  
 ,6240 10 03 BPL \$6245 62BC A6 14 LDY \$14 zu markieren  
 ,6242 4C 5C 62 JMP \$625C besser: BMI 62BE A4 15 PLA  
 ,6245 C9 22 CMP #\$22 = "2" 62C0 68 STA \$15 Startadresse  
 ,6247 D0 0A BNE \$6253 nach 62C1 85 15 PLA zu markieren  
 ,6249 AD 7E 60 LDA \$607E 62C3 68 STA \$14 Flag: Konstante  
 ,624C 49 FF EOR #\$FF AND 62C4 85 14 RTS Zeropage  
 ,624E 8D 7E 60 STA \$607E 62C6 38 LDA #\$00 6B08 initialisierung  
 ,6251 A9 22 LDA #\$22 "wiederholen" 62C7 60 STA \$04 low-Byte auf 0  
 ,6253 2C 7E 60 BIT \$607E 62C8 A9 00 LDA #\$B5  
 ,6256 30 04 BMI \$625C 62CA 85 04 STA \$B5  
 ,6258 C9 20 CMP #\$20 = SPACEZ 62CC 85 B5 STA \$BB  
 ,625A F0 D7 BEQ \$6233 JA 62CE 85 BB STA \$BD  
 ,625C C9 00 CMP #\$00 geholtes Byte = 0? 62D0 85 8D LDA #\$EO 7 benötigte Zeilenum.  
 ,625E 60 RTS 62D2 A9 E0 STA \$05 3 ab \$2000  
 ,625F C9 30 CMP #\$30 62D4 85 05 LDA #\$F0 7 Adressen zu 4/5  
 ,6261 B0 05 BCS \$6268  $\geq 0$  62D6 A9 F0 STA \$B6 3 ab \$3000  
 ,6263 A2 03 LDX #\$03 "ILLEGAL PARAMETER" 62DB 85 B6 LDA #\$A0 7 Zeilenummern  
 ,6265 4C B9 61 JMP \$61B9 Fehlerende 62DA A9 A0 STA \$BC 3 ab \$A000 ablegen  
 ,6268 C9 3A CMP #\$3A 62DC 85 8C LDA #\$B0 7 Speicherpositionen  
 ,626A 90 19 BCC \$6285  $\leq 9$  62DE A9 B0 STA \$BE 3 ab \$B000  
 ,626C C9 41 CMP #\$41 Rechenzeichen? 62E0 85 BE RTS  
 ,626E 90 F3 BCC \$6263 a. Error 62E2 60 LDA \$04 JSR von 62E5  
 ,6270 C9 5B CMP #\$5B Buchstabe? 62E3 A5 04 CLC 6C4E  
 ,6272 D0 22 BCS \$6263 nein, Error 62E5 18 ADC #\$02 4/5 um 2  
 ,6274 3B SEC 62E6 69 02 erlichen  
 ,6275 E9 41 SBC #\$41  $\rightarrow A=0$   
 B=1 etc.

TINYCOMPILER

..62E8 85 04	STD \$04	76/5 um 2	..6362 20 1E AB JSR \$AB1E ausgeben
..62EA A5 05	ADC #\$00	erlöhen	..6365 20 E4 FF JSR \$FFE4 Taste holen
..62EC 69 00	STA \$05		CMP ##4E "N"?
..62EE 85 05	LDA \$B5		BEQ \$6375 Ja
..62F0 25 05	CLC		CMP ##4A "J"
..62F2 18			BNE \$6365 nein
..62F3 69 02	ADC #\$02	BS/BG um 2	LDA #\$01 Protokoll ein
..62F5 85 B5	STA \$B5	erlöhen	(STA \$A6) 2-BYTE \$2C
..62F7 A5 B6	LDA \$B6		RTS
..62F9 69 00	ADC #\$00		LDA #\$00
..62FB 85 B6	STA \$B6	Vektoren	STA \$A6 Protokoll aus
..62FD 60	RTS	SP von class erh.	Zahl
..62FE A5 BB	LDA \$BB	Zeilenum-Vektor	JSR vom überlieferten holen
..6300 18	CLC		CMP ##23 = "#"?
..6301 69 02	ADC #\$02	+ 2	BEQ \$6388 Ja
..6303 85 BB	STA \$BB		CMP ##26 = "&"?
..6305 A5 BC	LDA \$BC		BEQ \$638E Jo
..6307 69 00	ADC #\$00		JMP \$625F
..6309 85 BC	STA \$BC		LDX \$14 ? Anfangsadresse
..630B A5 BD	LDA \$BD		LDY \$15 ? Lade
..630D 18	CLC	Speicherpos-Vektor	SEC Flag: Konstante
..630E 69 02	ADC #\$02		RTS & Ende
..6310 85 BD	STA \$BD	+ 2	LDX ##A1
..6311 A5 BE	LDA \$BE		LDY #\$00
..6314 69 00	ADC #\$00		CLC Flag: Variablen Sond. befehle
..6316 85 BE	STA \$BE		RTS mit ende
..6318 60	RTS		JSR von 6445, 6635 deci
..6319 AD 34 62	LDA \$6234	JSR von 6445, 6635 akt. Adr. conv. 68D1	JSR \$6233 Byte holen
..631C 38	SEC	-1	CMP ##45 = "E"?
..631D E9 01	SBC #\$01	um -6	BNE \$639E nein
..631F BD 34 62	STA \$6234	ablegen	JMP \$6462 RTS ablegen & PASS2
..6322 AD 35 62	LDA \$6235	ständ. ständ.	CMP ##43 = "C"?
..6325 E9 00	SBC #\$00	übertrag	BNE \$63B4 nein
..6327 BD 35 62	STA \$6235		LDY ##D7
..632A 60	RTS		JSR \$AAD7(-C2)
..632B A9 20	LDA #\$20	sempre GCF entarbeiten	ablegen
..632D A0 61	LDY #\$61	RUN STOP ERLAUBEN?	JSR \$61C8
..632F 20 1E AB	JSR \$AB1E	ausgeben	JSR \$61C8 nächste Zeile
..6332 20 E4 FF	JSR \$FFE4	Taste holen	
..6335 C9 4E	CMP ##4E "N"?		
..6337 F0 09	BEQ \$6342	Jo	
..6339 C9 4A	CMP ##4A "J"?		
..633A DO F5	BNE \$6332	nein, dann nächste Zeile	
..633D A9 01	LDA #\$01	Flag für STOP	
..633F 85 06	STA \$06	(setzen)	
..6341 60 .8424c	RTS		
..6342 A9 00	LDA #\$00	Flag für kein STOP	
..6344 85 06	STA \$06	setzen	
..6346 60	RTS	Illegal Byte	
..6347 48	PHA	Akkuraten	
..6348 A9 B0	LDA ##B0	Byte nach X	
..634A A0 60	LDY ##60	FOUND ILLEGAL BYTE	
..634C 20 1E AB	JSR \$AB1E	ausgeben	
..634F 68	PLA	Akkur zu rückholen	
..6350 AA	TAX		
..6351 A9 00	LDA #\$00	Byte nach X	
..6353 20 CD BD	JSR \$BDCD	High-Byte := 0	
..6356 20 D7 AA	JSR \$AAD7	CR ausgeben	
..6359 A2 08	LDX ##08	PROGRAM ERROR	
..635B 4C B9 61	JMP \$61B9	ausgabe	
..635E A9 C4	LDA ##C4	JSR von 6507 Protokoll?	
..6360 A0 60	LDY ##60	PROTOKOLIEREN?	

,63EF 85 FB STA \$FB Füllen-Routine-  
 ,63F1 AD BB 60 LDA \$608B Vektor  
 ,63F4 85 FC STA \$FC ~~hinkopieren~~  
 ,63F6 AD C9 61 LDA \$61C9 echter Programm-  
 ,63F9 AC CA 61 LDY \$61CA start  
 ,63FC 8D 89 60 STA \$6089 } hinkopieren  
 ,63FF BC BA 60 STY \$608A  
 ,6402 A0 05 LDY #\$05  
 ,6404 AD 89 60 LDA \$6089 Programmat low  
 ,6407 38 SEC  
 ,6408 E9 01 SBC #\$01 -1  
 ,640A 91 FB STA (\$FB), Y in Füllroutine  
 ,640C C8 INY  
 ,640D AD BA 60 LDA \$608A High  
 ,6410 E9 00 SBC #\$00 übertrage  
 ,6412 91 FB STA (\$FB), Y in Füllroutine  
 ,6414 20 28 62 JSR \$6228 S2 Nullen ablegen  
 ,6417 A0 01 LDY #\$01  
 ,6419 AD C9 61 LDA \$61C9 echter Programmstart  
 ,641C 91 14 STA (\$14), Y  
 ,641E 8D 8B 60 STA \$608B  
 ,6421 C8 INY in den  
 ,6422 AD CA 61 LDA \$61CA  
 ,6425 91 14 STA (\$14), Y  
 ,6427 8D 8C 60 STA \$608C  
 ,642A A9 20 LDA #\$20 JSR  
 ,642C 20 C8 61 JSR \$61C8 ablegen  
 ,642F AD 87 60 LDA \$6087 Füllen-Routine  
 ,6432 20 C8 61 JSR \$61C8 low+high  
 ,6435 AD 88 60 LDA \$6088  
 ,6438 20 C8 61 JSR \$61C8 ablegen  
 ,643B A5 2B LDA \$2B } BASIC-Aufang  
 ,643D A4 2C LDY \$2C  
 ,643F 8D 34 62 STA \$62347 Selbstmodifikation  
 ,6442 8C 35 62 STY \$6235 }  
 ,6445 20 19 63 JSR \$6319 und dekrementiere  
 ,6448 A9 FF LDA #\$FF  
 ,644A 85 A8 STA \$A8  
 ,644C 20 33 62 JSR \$6233 Byte aus BASIC-Tabelle  
 ,644F F0 03 BEQ \$6454 Null? ja  
 ,6451 4C 47 63 JMP \$6347 Found illegal byte END  
 ,6454 20 33 62 JSR \$6233 Link-Byte low  
 ,6457 85 02 STA \$02 ablegen  
 ,6459 20 33 62 JSR \$6233 Link-Byte High  
 ,645C D0 0C BNE \$646A nicht O  
 ,645E A5 02 LDA \$02 low  
 ,6460 D0 08 BNE \$646A nicht Null  
 ,6462 A9 60 LDA #\$60 sonst RTS ablegen  
 ,6464 20 C8 61 JSR \$61C8 (Programmende)  
 ,6467 4C AC 6B JMP \$6BAD zum PASS 2  
 ,646A 20 33 62 JSR \$6233 Zeilennummer low  
 ,646D 85 02 STA \$02 ablegen  
 ,646F 20 33 62 JSR \$6233 ablegen  
 ,6472 85 03 STA \$03 ablegen  
 ,6474 A5 A6 LDA \$A6 Protokoll?  
 ,6476 F0 11 BEQ \$6489 nein  
 ,6478 A9 8D LDA #\$8D  
 ,647A A0 60 LDY #\$60 ZEILE  
 ,647C 20 1E AB JSR \$AB1E ausgeben  
 ,647F A6 02 LDX #02  
 ,6481 A5 03 LDA \$03 Zeilennummer  
 ,6483 20 CD BD JSR \$BDCC der ausgeben

,6486 20 D7 AA JSR \$A000 ausgeben  
 ,6489 A0 00 LDY \$00 Hochkomma-Flag  
 ,648B BC 7E 60 STY \$6000  
 ,648E 84 A8 STY \$A8 C0  
 ,6490 A5 02 LDA \$02  
 ,6492 91 BB STA (\$BB), Y  
 ,6494 C8 INY  
 ,6495 A5 03 LDA \$03  
 ,6497 91 BB STA (\$BB), Y  
 ,6499 88 DEY  
 ,649A AD C9 61 LDA \$61C9 Speicherpos.  
 ,649D 91 8D STA (\$8D), Y in Tabelle  
 ,649F C8 INY ablegen  
 ,64A0 AD CA 61 LDA \$61CA  
 ,64A3 91 8D STA (\$8D), Y  
 ,64A5 20 FE 62 JSR \$62FE 88/C & PDIE eöhren  
 ,64A8 A5 06 LDA \$06 RUN STOP erlaubt?  
 ,64AA F0 0F BEQ \$64BB nein  
 ,64AC A9 20 LDA #\$20 JSR \$A82C ablegen  
 ,64AE 20 C8 61 JSR \$61C8  
 ,64B1 A9 2C LDA #\$2C  
 ,64B3 20 C8 61 JSR \$61C8  
 ,64B6 A9 A8 LDA #\$AB  
 ,64B8 20 C8 61 JSR \$61C8  
 ,64BB 20 33 62 JSR \$6233 Byte holen  
 ,64BE F0 85 BEQ \$6445 Null, dann nächste Zeile  
 ,64C0 C9 3B CMP #\$3B = ":"  
 ,64C2 D0 0A BNE \$64C7 nein  
 ,64C4 4C 0E 5 JMP \$6500 nächste Zeile  
 ,64C7 C9 9E CMP #\$9E = '\$S' ?  
 ,64C9 D0 67 BNE \$6532 nein  
 ,64CB 20 7A 63 JSR \$637A 16-Bit-Zahl holen  
 ,64CE B0 3B BCS \$6508 konstante  
 ,64D0 A9 AD LDA #\$AD LD A \$xxxx  
 ,64D2 20 C8 61 JSR \$61C8 ablegen  
 ,64D4 4C 0E 5 TXA low-Byte  
 ,64D7 C9 9E JSR \$61C8 ablegen  
 ,64D9 98 TYA High-Byte  
 ,64DA 20 C8 61 JSR \$61C8 ablegen  
 ,64DD E9 INX low+1  
 ,64DE D0 01 BNE \$64E1 kein Übergang  
 ,64E0 C8 INY High erh.  
 ,64E1 A9 AC LDA #\$AC LDY \$xxxx  
 ,64E3 20 C8 61 JSR \$61C8 ablegen  
 ,64E6 8A TXA low-Byte  
 ,64E7 20 C8 61 JSR \$61C8 } low-High  
 ,64EA 98 TYA ablegen  
 ,64EB 20 C8 61 JSR \$61C8 ablegen  
 ,64EE 20 1D 65 JSR \$651D STA / \$14/5 ablegen  
 ,64F1 A9 20 LDA #\$20 JSR \$E130  
 ,64F3 20 C8 61 JSR \$61C8 ablegen  
 ,64F6 A9 30 LDA #\$30  
 ,64F8 20 C8 61 JSR \$61C8 ablegen  
 ,64FB A9 E1 LDA #\$E1 (BASIC-SY5)  
 ,64FD 20 FF 61 JSR \$61C8  
 ,6500 20 33 62 JSR \$6233 Byte holen  
 ,6503 B9 FF BNE \$6500 eigendefiniert  
 ,6505 4C 45 64 JMP \$6445 nächste Zeile erreicht  
 ,6508 A9 A9 LDA #\$A9  
 ,650A 20 C8 61 JSR \$61C8 LD A \$xxxx ablegen  
 ,650D BA TXA low-Byte

# TINY COMPILER

Fortsetzung PRINT Text

.650E 20 C8 61 JSR \$61CB ablegen	.65AB 4C A6 66 JMP \$66A6
.6511 A9 A0 LDA \$61CB LDY #\$xx	.65AB AD C9 61 LDA \$61CB Speicherpos low
.6513 20 C8 61 LDA \$61CB ablegen	.65AE 18 CLD {+10}
.6516 90 LDA \$61CB high byte	.65AF 69 0A ADC #\$0A
.6517 20 C8 61 JSR \$61CB ablegen	.65B1 85 FB STA \$FB nach FB
.651A 4C EE 64 JMP \$64EE JSR \$61CB ablegen	.65B3 AD CA 61 LDA \$61CA ? übertrag
.651D A9 85 LDA #\$85 STA \$14	.65B6 69 00 ADC #\$00
.651F 20 C8 61 JSR \$61CB ablegen	.65BB 85 FC STA \$FC nach FC
.6522 A9 14 LDA #\$14 ablegen	.65BA A9 A9 LDA #\$A9 LDY #\$xx
.6524 20 C8 61 JSR \$61CB ablegen	.65BC 20 C8 61 JSR \$61CB ablegen
.6527 A9 84 LDA #\$84	.65BF A5 FB LDA \$FB Textpos low
.6529 20 C8 61 JSR \$61CB STX \$15	.65C1 20 C8 61 JSR \$61CB ablegen
.652C A9 15 LDA #\$15 ablegen	.65C4 A9 A0 LDA #\$A0 LDY #\$xx
.652E 20 C8 61 JSR \$61CB ablegen	.65C6 20 C8 61 JSR \$61CB ablegen
.6531 60 RTS	.65C9 A5 FC LDA \$FC Textpos high ablegen
.6532 C9 80 CMP #\$80 = "END"?	.65CB 20 C8 61 JSR \$61CB ablegen
.6534 D0 12 BNE \$654B nein	.65CE A9 20 LDA #\$20
.6536 A9 4C LDA #\$4C	.65D0 20 C8 61 JSR \$61CB
.6538 20 C8 61 JSR \$61CB JMP JE385	.65D3 A9 1E LDA #\$1E
.653B A9 85 LDA #\$85 ablegen	.65D5 20 C8 61 JSR \$61CB
.653D 20 C8 61 JSR \$61CB ablegen	.65D8 A9 AB LDA #\$AB
.6540 A9 E3 LDA #\$E3	.65DA 20 C8 61 JSR \$61CB
.6542 20 C8 61 JSR \$61CB ablegen	.65DD A9 4C LDA #\$4C JMP
.6544 4C 00 65 JMP \$6500 nächste Zeile	.65DF 20 C8 61 JSR \$61CB ablegen
.6548 C9 90 CMP #\$90 = "STOP"?	.65E2 AD C9 61 LDA \$61CA Speicherposition
.654A D0 17 BNE \$6563 nein	.65E5 85 FB STA \$FB
.654C A9 38 LDA #\$38 SEC	.65E7 AD CA 61 LDA \$61CA nach FB/FC
.654E 20 C8 61 JSR \$61CB ablegen	.65EA 85 FC STA \$FC
.6551 A9 4C LDA #\$4C besser	.65EC 20 C8 61 JSR \$61CB 2 Byte überspr.
.6553 20 C8 61 JSR \$61CB JMP \$A82P \$1634	.65EF 20 C8 61 JSR \$61CB
.6556 A9 2F LDA #\$2F ablegen	.65F2 20 33 62 JSR \$6233 Byte holen
.6558 20 C8 61 JSR \$61CB ablegen	.65F5 F0 2C BEQ \$6623 Zeileende?
.655B A9 A8 LDA #\$A8	.65F7 C9 22 CMP #\$22 Hochkomma?
.655D 20 C8 61 JSR \$61CB ablegen	.65F9 F0 1A BEQ \$6615 ja
.6560 4C 00 65 JMP \$6500 nächste Zeile	.65FB 20 C8 61 JSR \$61CB sonst ablegen
.6563 C9 9C CMP #\$9C = "CR"?	.65FE 4C F2 65 JMP \$65F nächste Zeile von 6628, 662E
.6565 D0 14 BNE \$657B nein	.6601 A9 00 LDA #\$00
.6567 A9 20 LDA #\$20 JSR	.6603 AB TAY
.6569 20 C8 61 JSR \$61CB	.6604 20 C8 61 JSR \$61CB Null ablegen
.656C AD B7 60 LDA \$6087	.6607 AD C9 61 LDA \$61C9 Zeiger in
.656F 20 C8 61 JSR \$61CB CLR-Routine	.660A 91 FB STA (\$FB), Y
.6572 AD BB 60 LDA \$608B ablegen	.660C AD CA 61 LDA \$61CA JMP
.657 20 C8 61 JSR \$61CB schreiben	.660F CB INY
.6578 4C 00 65 JMP \$6500 nächste Zeile	.6610 91 FB STA (\$FB), Y
.657B C9 BA CMP #\$BA = "RUN"?	.6612 4C 45 64 JMP \$6445 nächste Zeile
.657D D0 14 BNE \$6593 nein	.6615 20 33 62 JSR \$6233 Byte holen
.657F A9 4C LDA #\$4C	.6618 F0 09 BEQ \$6623 Zeileende, dann +CP
.581 20 C8 61 JSR \$61CB JMP	.661A C9 3B CMP #\$3B =
.584 AD BB 60 LDA \$608B ablegen	.661C F0 0D BEQ \$662B ja
.587 20 C8 61 JSR \$61CB ablegen	.661E A2 04 LDX #\$04 WRONG CHAR
.58A AD BC 60 LDA \$608C	.6620 4C B9 61 JMP \$61B9 ERROR
.58D 20 C8 61 JSR \$61CB ablegen	.6623 A9 0D LDA #\$0D Schraubengen
.590 4C 00 65 JMP \$6500 nächste Zeile	.6625 20 C8 61 JSR \$61CB
.593 C9 8F CMP #\$8F = "REM"	.6628 4C 01 66 JMP \$6601 System holen PRINT
.595 D0 03 BNE \$659A nein	.662B 20 33 62 JSR \$6233 System holen OK Zahl
.597 4C 00 65 JMP \$6500 nächste Zeile	.662E F0 D1 BEQ \$6601 Null, OK Zahl
.59A C9 99 CMP #\$99 = "PRINT"?	.6630 4C 1E 66 JMP \$661E System holen 65A1
.59C D0 0A BNE \$65AB nein	.6633 20 19 63 JSR \$6319 add. adr. dec.
.59E 20 33 62 JSR \$6233 Byte holen	.6636 20 7A 63 JSR \$637A Zahl holen
.5A1 C9 22 CMP #\$22 Hochkomma?	.6639 90 24 BCD \$665F Variable
.5A3 F0 06 BEQ \$65AB ja	.663B A9 A2 LDA #\$A2 LDX #\$xx
.5A5 4C 33 66 JMP \$6633 sonst PRINT Zahl	

,663D 20 C8 61 JSR \$61CB ablegen  
 ,6640 8A TXA low  
 ,6641 20 C8 61 JSR \$61CB ablegen  
 ,6644 A9 A9 LDA ##A9 LDA #\$xx  
 ,6646 20 C8 61 JSR \$61CB ablegen  
 ,6649 98 TYA High  
 ,664A 20 C8 61 JSR \$61CB ablegen  
 ,664D A9 20 LDA ##20 } JSR \$8DCD  
 ,664F 20 C8 61 JSR \$61CB ablegen  
 ,6652 A9 CD LDA ##CD } JSR \$8DCD  
 ,6654 20 C8 61 JSR \$61CB ablegen  
 ,6657 A9 BD LDA ##BD } JSR \$8DCD  
 ,6659 20 C8 61 JSR \$61CB call CR ablegen  
 ,665C 4C 80 66 JMP \$6680  
 ,665F A9 AE LDA ##AE LDX \$xxxx  
 ,6661 20 C8 61 JSR \$61CB ablegen  
 ,6664 8A TXA } Variableoffset  
 ,6665 20 C8 61 JSR \$61CB ablegen  
 ,6668 98 TYA } Variableoffset  
 ,6669 20 C8 61 JSR \$61CB offset  
 ,666C EB INX } offset  
 ,666D DO 01 BNE \$6670 } +1  
 ,666F C8 INY }  
 ,6670 A9 AD LDA ##AD LDA \$xxxx  
 ,6672 20 C8 61 JSR \$61CB ablegen  
 ,6675 BA TXA }  
 ,6676 20 C8 61 JSR \$61CB offset  
 ,6679 98 TYA } +1 ablegen  
 ,667A 20 C8 61 JSR \$61CB } JSR \$8DCD ablegen +1  
 ,667D 4C 4D 66 JMP \$664D } JUMP 668C  
 ,6680 20 33 62 JSR \$6233 Byte holen  
 ,6683 F0 0A BEQ \$668F Zeileende  
 ,6685 C9 3B CMP ##3B = ".."  
 ,6687 DO 18 BNE \$66A1 nein, Error  
 ,6689 20 33 62 JSR \$6233 Byte holen (Null)  
 ,668C 4C 45 64 JMP \$6445 nächste Zeile  
 ,668F A9 20 LDA ##20 } JSR \$AAD7  
 ,6691 20 C8 61 JSR \$61CB ablegen  
 ,6694 A9 D7 LDA ##D7 } JSR \$AAD7  
 ,6696 20 C8 61 JSR \$61CB ablegen  
 ,6699 A9 AA LDA ##AA } JSR \$AAD7  
 ,669B 20 C8 61 JSR \$61CB ablegen  
 ,669E 4C 45 64 JMP \$6445 nächste Zeile  
 ,66A1 A2 04 LDX ##04 } WRONG CHAR  
 ,66A3 4C B9 61 JMP \$61B9 } ERROR  
 ,66A6 C9 88 CMP ##88 } JUMP 65A8  
 ,66AB DO 0A BNE \$66B4 nein  
 ,66AA 20 7A 63 JSR \$637A Parameter holen  
 ,66AD 90 08 BCC \$66B7 Variable  
 ,66AF A2 05 LDX ##05 } LET ERROR  
 ,66B1 4C B9 61 JMP \$61B9 }  
 ,66B4 4C 02 68 JMP \$6802 Fortsetzung...  
 ,66B7 86 FE STX \$FE }  
 ,66B9 84 FF STY \$FF } Variableoffset merken  
 ,66BB 20 33 62 JSR \$6233 Byte holen  
 ,66BE C9 B2 CMP ##B2 = "?"  
 ,66C0 F0 05 BEQ \$66C7 }  
 ,66C2 A2 06 LDX ##06 MISSING CHAR  
 ,66C4 4C B9 61 JMP \$61B9 } ERROR  
 ,66C7 20 33 62 JSR \$6233 Zeichen holen  
 ,66CA C9 C2 CMP ##C2 = "PEEK" ?  
 ,66CC DO 03 BNE \$66D1 nein

,66CE 4C F1 68 JMP } SP1  
 ,66D1 20 19 63 JSR \$637A Adr. deca  
 ,66D4 20 7A 63 JSR \$637A Variablen  
 ,66D7 90 3C BCC \$6715 Variablen  
 ,66D9 A9 A9 LDA ##A9 LDA #\$xx  
 ,66DB 20 C8 61 JSR \$61CB ablegen  
 ,66DE 8A TXA low-Byte  
 ,66DF 20 C8 61 JSR \$61CB ablegen  
 ,66E2 A9 A0 LDA ##A0 LDY \$Sxx  
 ,66E4 20 C8 61 JSR \$61CB ablegen  
 ,66E7 98 TYA High-BYTE  
 ,66EB 20 C8 61 JSR \$61CB ablegen  
 ,66EB 4C 33 7 JMP \$6733 } JUMP von 668C 6862,6749  
 ,66EE A9 BD LDA ##BD STA \$Sxx  
 ,66FO 20 C8 61 JSR \$61CB ablegen  
 ,66F3 A5 FE LDA \$FE } Variableoffset  
 ,66F5 20 C8 61 JSR \$61CB ablegen  
 ,66FB A5 FF LDA \$FF }  
 ,66FA 20 C8 61 JSR \$61CB ablegen  
 ,66FD A9 8C LDA ##8C ST \$xxxx  
 ,66FF 20 C8 61 JSR \$61CB ablegen } inrement  
 ,6702 E6 FE INC \$FE }  
 ,6704 DO 02 BNE \$670B } increment  
 ,6706 E6 FF INC \$FF }  
 ,6708 A5 FE LDA \$FE } Offset +1  
 ,670A 20 C8 61 JSR \$61CB ablegen } Offset +1  
 ,670D A5 FF LDA \$FF } ablegen  
 ,670F 20 C8 61 JSR \$61CB } nächste Zeile  
 ,6712 4C 00 65 JMP \$6500 }  
 ,6715 A9 AD LDA ##AD LDA \$xxxx  
 ,6717 20 C8 61 JSR \$61CB ablegen } Variableoffset  
 ,671A BA TXA } ablegen  
 ,671B 20 C8 61 JSR \$61CB ablegen } Variableoffset  
 ,671E 98 TYA } ablegen  
 ,671F 20 C8 61 JSR \$61CB } ablegen  
 ,6722 A9 AC LDA ##AC LDY \$xxxx  
 ,6724 20 C8 61 JSR \$61CB ablegen } ablegen  
 ,6727 E8 INX }  
 ,6728 DO 01 BNE \$672B } +1  
 ,672A C8 INY }  
 ,672B 8A TXA }  
 ,672C 20 C8 61 JSR \$61CB } Offset +1  
 ,672F 98 TYA } ablegen  
 ,6730 20 C8 61 JSR \$61CB } ablegen  
 ,6733 20 33 62 JSR \$6233 }  
 ,6736 C9 AA CMP ##AA "+" ?  
 ,6738 F0 12 BEQ \$674C ja  
 ,673A C9 AB CMP ##AB "-" ?  
 ,673C F0 0E BEQ \$674C ja  
 ,673E C9 AC CMP ##AC "\*"?  
 ,6740 F0 0A BEQ \$674C ja  
 ,6742 C9 AD CMP ##AD "/" ?  
 ,6744 F0 06 BEQ \$674C ja } Zeile schreibe die  
 ,6746 20 19 63 JSR \$6319 Speicher" ablegen  
 ,6749 4C EE 66 JMP \$66EE }  
 ,674C 85 FD STA \$FD Rechnerzeichen setzen  
 ,674E A9 85 LDA ##85 STA \$FB  
 ,6750 20 C8 61 JSR \$61CB } ablegen  
 ,6753 A9 FB LDA ##FB }  
 ,6755 20 C8 61 JSR \$61CB } ablegen  
 ,6758 A9 84 LDA ##84 STY \$FC } ablegen

VAMPIER

675A 20 CB 61 JSR \$61CB *low ablegen*

675D A9 FC JSR \$61CB *high ablegen*

675F 20 D0 03 JSR \$61CB *low ablegen*

6762 20 A9 03 BC 167C *low ablegen*

6765 90 15 JSR \$61CB *ablegen*

6767 A9 17 JSR \$61CB *low ablegen*

6769 20 CB 61 TXA *low ablegen*

676E BA 61 JSR \$61CB *low ablegen*

676D 20 CB 61 LDA #\$A0 LDY \$xx *ablegen*

6770 A9 A0 JSR \$61CB *ablegen*

6772 20 CB 61 TYA *ablegen*

6775 98 JSR \$61CB *ablegen*

6776 20 CB 61 JMP \$67 *ablegen*

6779 4C 9A 67 LDA #\$xx *ablegen*

677C A9 AD JSR \$xx *ablegen*

677E 20 CB 61 TXA *ablegen*

6781 BA 61 JSB *ablegen*

6782 20 CB 61 TV *ablegen*

6785 98 BD *ablegen*

6786 20 CB 61 BD *ablegen*

6789 EB AC *ablegen*

678A D0 01 61CB *ablegen*

678C CB *ablegen*

678E A9 AC \$61CB *ablegen*

678F 20 CB 61 *ablegen*

6792 BA \$61CB *ablegen*

6793 2A #\$85 STA \$FD *ablegen*

6796 GR \$61CB *ablegen*

6797 DA #\$FD *ablegen*

6799 JSR \$61CB *ablegen*

679L D0 04 *ablegen*

679J JSR \$61CB *ablegen*

679K LDA #\$A7 *ablegen*

679L JSR \$61CB *ablegen*

679M LDA #\$FD *ablegen*

679N CMP #\$AA = "+ "?" *Rechenzeichen*

679O BNE \$67BD *nein*

F 60 LDA \$607F *Additionsvektor*

30 60 LDY \$6080 *Additionsvektor*

DD 67 JMP \$67DD *Subtraktionsvektor*

AB CMP #\$AB == "?" *Subtraktionsvektor*

09 BNE \$67CA *nein*

ID B1 60 LDA \$6081 *Multiplicationsvektor*

AC B2 60 LDY \$6082 *Multiplicationsvektor*

4C DD 67 JMP \$67DD *Divisionsvektor*

C9 AC CMP #\$AC = "x"?" *Multiplicationsvektor*

DO 09 BNE \$67D7 *nein*

AD B3 60 LDA \$6083 *Multiplicationsvektor*

AC B4 60 LDY \$6084 *Multiplicationsvektor*

4C DD 67 JMP \$67DD *Divisionsvektor*

AD B5 60 LDA \$6085 *Divisionsvektor*

AC B6 60 LDY \$6086 *Divisionsvektor*

4B PHA *Divisionsvektor*

A9 20 LDA #\$20 JSR *ablegen*

20 CB 61 JSR \$61CB *ablegen*

6B PLA *ablegen*

20 CB 61 JSR \$61CB *low ablegen*

98 TYA *high ablegen*

20 CB 61 JSR \$61CB *ablegen*

67EB A9 05 LDA #\$A5 LDA \$FE *ablegen*

67ED 20 CB 61 JSR \$61CB LDA #\$FE *ablegen*

67F0 A9 FE LDA #\$FE *ablegen*

67F2 20 CB 61 JSR \$61CB LDA #\$A4 *ablegen*

67F5 A9 A4 LDA #\$A4 *ablegen*

67F7 20 CB 61 JSR \$61CB LDA #\$FF *ablegen*

67FA A9 FF LDA #\$FF *ablegen*

67FC 20 CB 61 JSR \$61CB *ablegen*

67FF 4C EE 66 JMP \$66EE *variable* *ablegen*

6802 D9 97 CMP #\$97 = "POKE?" *nein*

6804 D0 2E BNE \$6834 *nein*

6806 20 7A 63 JSR \$637A *zählt notieren*

6809 08 PHP *Carry retten*

680A B6 FE STX \$FE *zahl bzw. Offset*

680C B4 FF STY \$FF *retten* *notieren*

680E 20 33 62 JSR \$6233 *Zeichen notieren*

6811 C9 2C CMP #\$2C = ","? *?*

6813 F0 05 BEQ \$6B1A *ja*

6815 A2 06 LDX #\$06 *MISSING CHAR*

6817 4C B9 61 JMP \$61B9 *ERROR*

681A 20 7A 63 JSR \$637A *zählt notieren*

681D 90 18 BCC \$6837 *variable notieren*

681F D0 00 CPY #\$00 *zahlgröße 205*

6821 F0 05 BEQ \$6828 *nein*

6823 42 0E LDX #\$0E *Illegal Quantity*

6825 *JMP \$6824* JMP \$A437 *ERROR*

6828 A9 A9 LDA #\$A9 *LDA \$xx*

682A 20 CB 61 JSR \$61CB *ablegen*

682D BA TXA *wert*

682E 20 CB 61 JSR \$61CB *ablegen*

6831 4C 44 68 JMP \$6844 *Fortsetzung*

6834 4C B2 68 JMP \$68B2 *Fortsetzung*

6837 A9 AD LDA #\$AD *LDA \$xxxx*

6839 20 CB 61 JSR \$61CB *ablegen*

683C BA TXA *variable*

683D 20 CB 61 JSR \$61CB *offset ablegen*

6840 98 TYA *Carry holen*

6841 20 CB 61 JSR \$61CB *variable*

6844 28 PLP *Carry holen*

6845 90 12 BCC \$6859 *in Variable Poli*

6847 A9 8D LDA #\$8D STA \$xxxx *ablegen*

6849 20 CB 61 JSR \$61CB *ablegen*

684C A5 FE LDA \$FE *Adresse*

684E 20 CB 61 JSR \$61CB *Adresse*

6851 A5 FF LDA \$FF *ablegen*

6853 20 CB 61 JSR \$61CB *ablegen*

6856 4C 00 65 JMP \$6500 *nächste Zeile*

6859 A9 48 LDA #\$48 PHA *PHA*

685B 20 CB 61 JSR \$61CB *ablegen*

685E A9 AD LDA #\$AD LDA \$xxxx *ablegen*

6860 20 CB 61 JSR \$61CB *ablegen*

6863 A5 FE LDA \$FE *offset*

6865 20 CB 61 JSR \$61CB *ablegen*

6868 A5 FF LDA \$FF *ablegen*

686A 20 CB 61 JSR \$61CB *ablegen*

686D E6 FE INC \$FE *Offset*

686F D0 02 BNE \$6873 *erhöhen*

6871 E6 FF INC \$FF *erhöhen*

6873 A9 AC LDA #\$AC LDY \$xxxx *ablegen*

6875 20 CB 61 JSR \$61CB *ablegen*

6878 A5 FE LDA \$FE *?*

687A 20 CB 61 JSR \$61CB } offset  
 687D A5 FF LDA \$FF  
 687F 20 CB 61 JSR \$61CB  
 6882 A9 85 LDA #\$85 STA \$FB  
 6884 20 CB 61 JSR \$61CB  
 6887 A9 FB LDA #\$FB ablegen  
 6889 20 CB 61 JSR \$61CB  
 688C A9 84 LDA #\$84 STY \$FC  
 688E 20 CB 61 JSR \$61CB LDY #100  
 6891 A9 FC LDA #\$FC ablegen  
 6893 20 CB 61 JSR \$61CB  
 6896 A9 A0 LDA #\$A0  
 6898 20 CB 61 JSR \$61CB LDY #100  
 689B A9 00 LDA #\$00 ablegen  
 689D 20 CB 61 JSR \$61CB  
 68A0 A9 68 LDA #\$68 PLA  
 68A2 20 CB 61 JSR \$61CB ablegen  
 68A5 A9 91 LDA #\$91 STA (\$FB), Y  
 68A7 20 CB 61 JSR \$61CB LDY (\$FB), Y  
 68AA A9 FB LDA #\$FB ablegen  
 68AC 20 CB 61 JSR \$61CB ablegen  
 68AF 4C 00 65 JMP \$6500 nächste Zeile  
 68B2 C9 B9 CMP #\$B9 = "GOTO"  
 68B4 D0 0F BNE \$68C5 kein  
 68B6 A9 4C LDA #\$4C JMP  
 68B8 8D C9 4B STA \$68C9 in Routine  
 68BB 20 7A 63 JSR \$637A Zahl holen  
 68BE B0 08 BCS \$68C8 Konstante  
 68C0 A2 03 LDX #\$03 illegal parameter  
 68C2 4C B9 61 JMP \$61B9 ERROR  
 68C5 4C 65 69 JMP \$6965 Fortsetzung...  
 68C8 A9 20 LDA #\$20 Fortsetzung...  
 68CA 20 CB 61 JSR \$61CB ablegen  
 68CD 84 FB STY \$FB Y retten  
 68CF A0 00 LDY #\$00  
 68D1 8A TXA  
 68D2 91 04 STA (\$04), Y Zeilenr. low  
 68D4 A5 FB LDA #\$FB ablegen  
 68D6 CB INY  
 68D7 91 04 STA (\$04), Y Zeilenr. high  
 68D9 BB DEY ablegen  
 68DA AD C9 61 LDA \$61C9 Speicherpos. low  
 68DD 91 B5 STA (\$B5), Y in Tabelle  
 68DF CB INY  
 68E0 AD CA 61 LDA \$61CA high  
 68E3 91 B5 STA (\$B5), Y in Tabelle  
 68E5 20 E3 62 JSR \$62E3 Vektoren entnehmen  
 68E8 20 CB 61 JSR \$61CB 2 Bytes übergreifen  
 68EB 20 CB 61 JSR \$61CB  
 68EE 4C 00 65 JMP \$6500 nächste Zeile  
 68F1 20 33 62 JSR \$6233 von 66CE Zeichen holen  
 68F4 C9 28 CMP #\$28 = "C"? FEEK  
 68F6 F0 05 BEQ \$68FD ja  
 68F8 A2 06 LDX #\$06 MISSING CHAR  
 68FA 4C B9 61 JMP \$61B9 ERROR  
 68FD 20 7A 63 JSR \$637A Zahl holen  
 6900 90 1A BCC \$691C Variable  
 6902 A9 A0 LDA #\$A0 LDY #100  
 6904 20 CB 61 JSR \$61CB LDY #100  
 6907 A9 00 LDA #\$00 ablegen  
 6909 20 CB 61 JSR \$61CB ablegen  
 690C A9 AD LDA #\$AD

, 690E 20 CB 61  
 , 6911 94 TXA JSR TYA JSR \$61CB Zahl  
 , 6912 20 CB 61 JSR \$61CB ablegen  
 , 6916 20 CB 61 JMP \$66EE LDA #\$AD LDA \$xxxx ablegen  
 , 6919 4C EE 66 JSR \$61CB ablegen  
 , 691C A9 00 LDA \$61CB ablegen  
 , 691E 20 CB 61 JSR \$61CB ablegen  
 , 6921 8A TXA JSR \$61CB ablegen  
 , 6922 20 CB 61 TYA JSR \$61CB Variableoffset  
 , 6925 9B INX JSR \$61CB ablegen  
 , 6926 20 CB 61 LDA \$61CB ablegen  
 , 6927 99 01 NE \$692D } +1  
 , 692D 4C AC 1Y LDA \$61CB ablegen  
 , 692F 20 CB 61 #\$AC LDY \$xxxx ablegen  
 , 6932 BA \$61CB ablegen  
 , 6933 20 CB 61 LDY \$xxxx ablegen  
 , 6936 98 offset +1  
 , 6937 20 CB 61 JSR \$61CB ablegen  
 , 693A A9 85  
 , 693C 20 CB 61 JSR \$61CB ablegen  
 , 693F A9 FB LDA \$61CB ablegen  
 , 6941 20 CB 61 JSR \$61CB ablegen  
 , 6944 A9 84 LDY \$xxxx ablegen  
 , 6946 20 CB 61 JSR \$61CB ablegen  
 , 6949 A9 FC LDA \$74 \$FC ablegen  
 , 694B 20 CB 61 JSR \$61CB ablegen  
 , 694E A9 A0 LDA \$61CB ablegen  
 , 6950 20 CB 61 JSR \$61CB ablegen  
 , 6953 A9 00 LDA \$61CB ablegen  
 , 6955 20 CB 61 JSR \$61CB ablegen  
 , 6958 A9 B1 LDA \$61CB ablegen  
 , 695A 20 CB 61 JSR \$61CB ablegen  
 , 695D A9 FB LDA \$61CB ablegen  
 , 695F 20 CB 61 JSR \$61CB ablegen  
 , 6962 4C EE 66 JMP \$66  
 , 6965 C9 2E CMP #\$2E  
 , 6967 D0 03 BNE \$696  
 , 6969 4C 94 63 JMP \$6394  
 , 696C C9 8D CMP #\$BD  
 , 696E D0 08 BNE \$6978  
 , 6970 A9 20 LDA #\$20 JSR \$61CB ablegen  
 , 6972 8D C9 68 XTA \$68C9  
 , 6975 4C 6E 6B JMP \$68B  
 , 6978 4C BD 4A JMP \$6ABD besser  
 , 697B A9 AD LDA #\$AD LDY \$xxxx ablegen  
 , 697D 20 CB 61 JSR \$61CB ablegen  
 , 6980 A5 FB LDA \$61CB ablegen  
 , 6982 20 CB 61 JSR \$61CB ablegen  
 , 6985 A5 FC LDA \$61CB ablegen  
 , 6987 20 CB 61 JSR \$61CB ablegen  
 , 698A E6 FB INC \$FB ablegen  
 , 698C DO 02 BNE \$6990 ablegen  
 , 698E E6 FC INC \$FC ablegen  
 , 6990 A9 AC LDA #\$AC LDY \$xxxx ablegen  
 , 6992 20 CB 61 JSR \$61CB ablegen  
 , 6995 A5 FB LDA \$61CB ablegen  
 , 6997 20 CB 61 JSR \$61CB ablegen  
 , 699A A5 FC LDA \$61CB ablegen  
 , 699C A5 FC JSR \$6AC5 ablegen

699C 20 CB 61 JSR  
 699F 20 7A 63 JSR  
 69A2 0E LDA #FD  
 69A3 68 HND #01 Carry-Flag  
 69A4 20 01 BEQ \$60B6 Variable  
 69A4 85 FD LDA #\$00 COPY Flag  
 69A8 FO OC JSR \$61CB ablegen  
 69AA A9 00 TYA  
 69AC 20 CB 61 JSR \$61CB  
 69AF 98 STX \$FB offset  
 69B0 20 CB 61 JSR \$69CB ablegen  
 69B3 4C CB 69 JMP \$6A67  
 69B6 86 FB STY \$FC  
 69B8 B4 FC INX +1  
 69BA EB BNE \* COPY \$xxxx  
 69BB D0 01 INY B ablegen  
 69BD CB LDA \$61CB  
 69BE A9 CC JSR \$61CB low-High  
 69C0 20 CB 61 JSR \$61CB ablegen  
 69C3 9A JSR \$61CB  
 69C4 20 CB 61 JSR \$61CB BCC  
 69C7 98 JSR \$61CB ablegen  
 69D9 20 CB JSR \$61CB  
 69E2 A9 90 Konstante: 6  
 69CD 20 CB \$FD Variable: 7  
 69D0 A9 1C SEC  
 69D2 35 ADC #\$06  
 69D3 JSR \$61CB ablegen  
 69DE LDA #\$D0 BNE  
 69E1 JSR \$61CB ablegen  
 69E1 LDA #\$01  
 69E2 SEC Konstante: 6  
 69E3 SBC \$FD Variable: 8  
 69E4 CLC  
 69E5 ADC #\$07  
 69E6 00 CB 61 JSR \$61CB ablegen  
 69E7 FD LDA \$FD  
 69E8 F0 OC BEQ \$69FB Variable  
 69E9 C9 LDA #\$C9 CMP #\$xx  
 69EA 20 CB 61 JSR \$61CB ablegen  
 69EB 8A TXA low  
 69EC 20 CB 61 JSR \$61CB ablegen  
 69ED 4C 0A 6A JMP \$6AOA  
 69FB A9 CD LDA #\$CD  
 69FD 20 CB 61 JSR \$61CB ablegen  
 6A00 A5 FB LDA \$FB  
 6A02 20 CB 61 JSR \$61CB Variable offset  
 6A05 A5 FC LDA \$FC  
 6A07 20 CB 61 JSR \$61CB ablegen  
 6A0A A9 B0 LDA #\$B0 BCS  
 6A0C 20 CB 61 JSR \$61CB ablegen  
 6A0F A9 03 LDA #\$03 3 Bytes überprüfen  
 6A11 20 CB 61 JSR \$61CB weiter wie bei " >  
 6A14 4C 4F 6B JMP \$6B6E  
 6A17 A9 AD LDA #\$AD LDA \$xxxx  
 6A19 20 CB 61 JSR \$61CB ablegen  
 6A1C A5 FB LDA \$FB offset  
 6A1E 20 CB 61 JSR \$61CB ablegen  
 6A21 A5 FC LDA \$FC ablegen

6A26 E6 FD INC \$FB  
 6A28 D0 02 BNE \$6A2C  
 6A2A E6 FC INC \$FC  
 6A2C A9 AC LDA #\$AC LDY \$xxxx  
 6A2E 20 CB 61 JSR \$61CB ablegen  
 6A31 65 FB LDA \$FB JSR \$61CB } Offset +1  
 6A33 20 CB 61 JSR \$61CB LDA \$FC  
 6A36 A5 FC JSR \$61CB } ablegen  
 6A38 20 CB 61 JSR \$61CB Zahl holen  
 6A3B 20 7A 63 JSR \$637A  
 6A3E 08 PHP  
 6A3F 68 PLA } Carry-Flag  
 6A40 29 01 AND #\$01 nach \$FD  
 6A42 85 FD STA \$FD  
 6A44 F0 0C BEQ \$6A52 Variable  
 6A46 A9 C0 LDA #\$C0 CPY \$xx  
 6A48 20 CB 61 JSR \$61CB ablegen  
 6A4B 98 TYA high  
 6A4C 20 CB 61 JSR \$61CB ablegen  
 6A4F 4C 67 6A JMP \$6A67  
 6A52 86 FB STX \$FB } offset ablegen  
 6A54 84 FC STY \$FC  
 6A56 E8 INX  
 6A57 D0 01 BNE \$6A5A } offset ablegen  
 6A59 CB INY  
 6A5A A9 CC LDA #\$CC CPY \$xxxx  
 6A5C 20 CB 61 JSR \$61CB ablegen  
 6A5F BA TXA  
 6A60 20 CB 61 JSR \$61CB } offset +1  
 6A63 98 TYA  
 6A64 20 CB 61 JSR \$61CB  
 6A67 A9 90 LDA #\$90 BCC  
 6A69 20 CB 61 JSR \$61CB ablegen  
 6A6C A9 01 LDA #\$01 Konstante: \$B  
 6A6E 38 SEC Variable: \$C  
 6A6F E5 FD SBC \$FD Konstante: \$B  
 6A71 18 CLC Variable: \$C  
 6A72 69 0B ADC #\$0B  
 6A74 20 CB 61 JSR \$61CB ablegen  
 6A77 A9 D0 LDA #\$D0 BNE  
 6A79 20 CB 61 JSR \$61CB ablegen  
 6A7C A9 01 LDA #\$01 Konstante: 6  
 6A7E 38 SEC Variable: 7  
 6A7F E5 FD SBC \$FD Konstante: 6  
 6A81 18 CLC Variable: 7  
 6A82 69 06 ADC #\$06  
 6A84 20 CB 61 JSR \$61CB ablegen  
 6A87 A5 FD LDA \$FD  
 6A89 F0 OC BEQ \$6A97 Variable  
 6A8B A9 C9 LDA #\$C9 CMP #\$xx  
 6A8D 20 CB 61 JSR \$61CB ablegen  
 6A90 8A TXA low ablegen  
 6A91 20 CB 61 JSR \$61CB  
 6A94 4C A6 6A JMP \$6AA6  
 6A97 A9 CD LDA #\$CD CMP #\$xxxx  
 6A99 20 CB 61 JSR \$61CB ablegen  
 6A9C A5 FB LDA \$FB  
 6A9E 20 CB 61 JSR \$61CB } Variablen-  
 6AA1 A5 FC LDA \$FC Offset  
 6AA3 20 CB 61 JSR \$61CB ablegen

6AAB 20 08 61 JSR \$61CB ablegen  
 6AAB A9 05 LDA #\$05 über  
 6AAD 20 C8 61 JSR \$61CB  
 6AB0 A9 F0 LDA #\$F0 F0  
 6AB2 20 C8 61 JSR \$61CB ablegen  
 6AB5 A9 03 LDA #\$03 über  
 6AB7 20 C8 61 JSR \$61CB  
 6ABA 4C 6F 6B JMP \$6B6E wie bei "="  
 6ABD C9 8B CMP #\$BB = "F" ?  
 6AC1 20 7A 63 JSR \$637A nein  
 6AC4 90 0B BCC \$6ACE Parameter holen  
 6AC6 A2 03 LDX #\$03 Variable  
 6ACB 4C B9 61 JMP \$61B9 ERROR  
 6ACB 4C BD 6B JMP \$6B8D Fortsetzung...

6ADE 86 FB STX #\$FB Variablenzeiger ablegen  
 6AD0 84 FC STY #\$FC Variablenzeiger ablegen  
 6AD2 20 33 62 JSR \$6233 Byte holen  
 6AD5 C9 B2 CMP #\$B2 ?  
 6AD7 F0 13 BEQ \$6AE0  
 6AD9 C9 B3 CMP #\$B3 ?  
 6ADB F0 09 BEQ \$6AE6  
 6ADD C9 B1 CMP #\$B1 ?  
 6ADF F0 08 BEQ \$6AE9  
 6AE1 A2 06 LDX #\$06 MISSING CHAR  
 6AE3 4C B9 61 JMP \$61B9 ERROR  
 6AE6 4C 7B 69 JMP \$697B "S"-Routine  
 6AE9 4C 17 6A JMP \$6A17 "S"-Routine  
 6AEC A9 AD LDA #\$AD CDA \$xxxx  
 6AEE 20 C8 61 JSR \$61CB ablegen  
 6AF1 A5 FB LDA #\$FB  
 6AF3 20 C8 61 JSR \$61CB Variablenzeiger  
 6AF6 A5 FC LDA #\$FC ablegen  
 6AF8 20 C8 61 JSR \$61CB  
 6AFB A9 AC LDA #\$AC LDY \$xxxx  
 6AFD 20 C8 61 JSR \$61CB ablegen  
 6B00 E6 FB INC #\$FB Variablenzeiger  
 6B02 DO 02 BNE \$6B06  
 6B04 E6 FC INC #\$FC incrementieren  
 6B06 A5 FB LDA #\$FB Variablenzeiger  
 6B08 20 C8 61 JSR \$61CB +1  
 6B0B A5 FC LDA #\$FC ablegen  
 6B0D 20 C8 61 JSR \$61CB  
 6B10 20 7A 63 JSR \$637A Zahl holen  
 6B13 08 PHP Prozessorstatus  
 6B14 68 PLA nach Akku  
 6B15 29 01 AND #\$01 Carry-Flag isolieren  
 6B17 85 FD STA #\$FD ablegen  
 6B19 F0 0C BEQ \$6B27 Variable  
 6B1B A9 C9 LDA #\$C9 CMP #\$xx  
 6B1D 20 C8 61 JSR \$61CB ablegen  
 6B20 BA TXA low-Byte  
 6B21 20 C8 61 JSR \$61CB ablegen  
 6B24 4C 34 6B JMP \$6B34  
 6B27 A9 CD LDA #\$CD CMP \$xxxx  
 6B29 20 C8 61 JSR \$61CB ablegen  
 6B2C BA TXA low + High  
 6B2D 20 C8 61 JSR \$61CB ablegen  
 6B30 98 TYA  
 6B31 20 C8 61 JSR \$61CB  
 6B34 A9 DO LDA #\$DO BNE ablegen  
 6B36 20 C8 61 JSR \$61CB

6B3E 20 1E AB JSR \$6B3E ablegen  
 6B3F A9 35 LDA #\$35 ROM aus-  
 6B41 20 C8 61 JSR \$61CB ADC #\$01 schalten  
 6B44 A5 FD BEQ \$6B54 Variable  
 6B46 F0 0C LDA #\$FD JSR \$61CB ablegen  
 6B48 A9 C0 TYA High-Side  
 6B4D 98 JMP \$6B65 ablegen  
 6B51 4C 35 JSR \$61CB ablegen  
 6B54 A9 CC LDA #\$CC R \$61CB CPU XXXX ablegen  
 6B56 20 C8 61 JSR \$61CB Variablenzeiger erhöhen  
 6B59 EB LDA #\$EB JSR \$61CB ablegen  
 6B5A DO 01 BNE ablegen  
 6B5C CB  
 6B5D 8A  
 6B5E 20 C8 61 JSR \$61CB ablegen  
 6B61 98 TYA Low ablegen  
 6B62 20 C8 61 JSR \$61CB ablegen  
 6B65 A9 DO LD B High  
 6B67 20 C8 61 JSR \$61CB ablegen  
 6B6A A9 03 LDA BNE  
 6B6C 20 C8 61 JSR \$61CB ablegen  
 6B6F A9 4C LDA Pyte überprüfen  
 6B71 20 C8 61 JSR \$61CB ablegen  
 6B74 20 33 62 JSR \$61CB ablegen  
 6B77 C9 A7 CMP #? ablegen  
 6B79 F0 05 BEQ \$61CB ablegen  
 6B7B A2 06 LDX #\$0 ? ablegen  
 6B7D 4C B9 61 JMP \$61B9 ablegen  
 6B80 20 7A 63 JSR \$637A CHAR  
 6B83 B0 05 BCS \$6B83 ablegen  
 6B85 A2 03 LDX #\$03 ablegen  
 6B87 4C B9 61 JMP \$61B9 ablegen  
 6B8A 4C CD AB JMP \$6BCD ablegen  
 6BBD C9 BE CMP #\$BE = METER ablegen  
 6BDF DO 08 BNE \$6B99 ablegen  
 6B91 A9 60 LDA #\$60 RT ablegen  
 6B93 20 C8 61 JSR \$61CB ablegen  
 6B96 4C 00 65 JMP \$6500 nach ablegen  
 6B99 EA NOP  
 6B9A C9 80 CMP #\$80 Token ablegen  
 6B9C 90 09 BCC \$6BA7 kleiner ablegen  
 6B9E C9 CC CMP #\$CC Token ablegen  
 6BA0 B0 05 BCS \$6BA7 größer ablegen  
 6BA2 A2 02 LDX #\$02 INVALID STATEMENT ablegen  
 6BA4 4C B9 61 JMP \$61B9 Ende ablegen  
 6BA7 A2 01 LDX #\$01 MISSING ablegen  
 6BA9 4C B9 61 JMP \$61B9 Ende ablegen  
 6BAC A9 DC LDA #\$DC PASS 2 ablegen  
 6BAE A0 60 LDY #\$60 PASS 2 ablegen  
 6BBD 20 1E AB JSR \$AB1E ausgeben ablegen  
 6BB3 A9 35 LDA #\$35 ROM aus-  
 6BB5 7B SEI schalten ablegen  
 6BB6 B5 01 STA #\$01 benutzte Zeilenum.  
 6BB8 A5 04 LDA #\$04 Vektor ablegen  
 6BBA B5 FB STA #\$FB ablegen  
 6BBC A5 05 LDA #\$05 retten ablegen

# TINYCOMPILER

7!

,6BBE 85 FC	STA \$FC	,6C3B B1 B5	LDA (\$B5), Y wo es gebraucht wird
,6BC0 A5 BB	LDA \$BB	,6C3D 85 03	STA \$03 wdh holen wird
,6BC2 85 FD	STA \$FD	,6C3F A0 00	LDY #\$00
,6BC4 A5 8C	LDA \$8C	,6C41 B1 BD	LDA (\$8D), Y Speicherpos
,6BC6 85 FE	STA \$FE	,6C43 91 02	STA (\$02), Y in Programm
,6BC8 20 D8 62	JSR \$62CB Vektoren auf Ausgängen	,6C45 85 06	→ STA \$06
,6BCB A5 04	LDA \$04 (=0)	,6C47 C8	INY
,6BCD C5 FB	CMP \$FB bei Znr Ende	,6C48 B1 BD	LDA (\$BD), Y } High-Byte
,6BCF 90 0E	BCC \$6BDF es gibt Zeilennr	,6C4A 91 02	STA (\$02), Y } Kopieren
,6BD1 A5 05	LDA \$05 Ausgangswert mit	,6C4C 85 8F	→ STA \$BF
,6BD3 C5 FC	CMP \$FC Endwert vergl.	,6C4E 20 E3 62	JSR \$62E3 erhöhen
,6BD5 90 08	BCC \$6BDF ergibt Zeilennr	,6C51 A9 00	LDA #\$00
,6BD7 A9 37	LDA #\$37	,6C53 85 8B	STA \$8B
,6BD9 85 01	STA \$01 { ROMein	,6C55 85 8D	STA \$8D
,6BDB 58	CLI	,6C57 A9 A0	LDA #\$A0
,6BDC 4C 62 6C	JMP \$6C62	,6C59 85 8E	STA \$8C
,6BDF A0 00	LDY #\$00	,6C5B A9 B0	LDA #\$B0
,6BE1 B1 04	LDA (\$04), Y } benutzte Zeilen-	,6C5D 85 8E	STA \$8E
,6BE3 85 02	STA \$02 nummen } nummer	,6C5F 4C CB 6B	JMP \$6BCB nächste Zeilenr. JMP \$6BDC
,6BE5 C8	INY	,6C62 A9 98	LDA #\$9B ENDE LETZTES BYTE
,6BE6 B1 04	LDA (\$04), Y nach \$2B	,6C64 A0 60	LDY #\$60 BEI
,6BE8 85 03	STA \$03	,6C66 20 1E AB	JSR \$AB1E drucken
,6BEA A5 A6	LDA \$A6 Protokoll?	,6C69 AD C9 61	LDA \$61C9 akt. Byte low
,6BF F0 1B	BEQ \$6C09 nein	,6C6C 38	SEC
,6BEE A9 37	LDA #\$37 { ROM	,6C6D E9 01	SBC #\$01 minus 1
,6BF0 85 01	STA \$01 { ein	,6C6F AA	TAX
,6BF2 58	CLI	,6C70 AD CA 61	LDA \$61CA High-Byte
,6BF3 A9 E6	LDA #\$E6 SUCHEN ZEILE	,6C73 E9 00	SBC #\$00 Übertrag
,6BF5 A0 60	LDY #\$60	,6C75 20 CD BD	JSR \$BDCC derz. ausgeben
,6BF7 20 1E AB	JSR \$AB1E ausgeben	,6C78 4C D7 AA	JMP \$AAD7 CR ausgeben
,6BFA A6 02	LDX \$02	,6C7B 00	BRK
,6BFC A5 03	LDA \$03 Zeilennummer	,6C7C 00	BRK
,6BFE 20 CD	BD JSR \$BDCC decimal ausgeben	,6C7D 00	BRK
,6C01 20 D7 AA	JSR \$AAD7 CR drucken	,6C7E 00	BRK
,6C04 A9 35	LDA #\$35 { ROM	,6C7F 00	BRK
,6C06 78	SEI	,6C80 00	BRK
,6C07 85 01	STA \$01 { aus		
,6C09 A5 BB	LDA \$BB		
,6COB C5 FD	CMP \$FD		
,6COD 90 10	BCC \$6C1F es gibt noch Zeilenr		
,6COF A5 BC	LDA \$BC		
,6C11 C5 FE	CMP \$FE		
,6C1 90 0A	BCC \$6C1F es gibt noch Zeilenr		
,6C15 A9 37	LDA #\$37 { ROM		
,6C17 85 01	STA \$01 { ein		
,6C19 58	CLI		
,6C1A A2 07	LDX #\$07 LINE NOT FOUND		
,6C1C 4C B9 61	JMP \$61B9 ERROR		
,6C1F A0 00	LDY #\$00		
,6C21 B1 8B	LDA (\$BB), Y Zeilenr aus Tabelle holen		
,6C23 C5 02	CMP \$02 = gesuchte Zeilenr?		
,6C25 F0 06	BEQ \$6C2D ja		
,6C27 20 FE 62	JSR \$62FE erhöhen		
,6C2A 4C 09 6C	JMP \$6C09 LOOP		
,6C2D C8	INY		
,6C2E B1 8B	LDA (\$BB), Y High-Byte auch		
,6C30 C5 03	CMP \$03 identisch?		
,6C32 D0 F3	BNE \$6C27 nein		
,6C34 A0 00	LDY #\$00		
,6C36 B1 B5	LDA (\$B5), Y wo es gebraucht wird		
,6C38 85 02	STA \$02 low holen		
,6C3A C8	INY		

02/03: aktuelle Zeilennummer

04/05: Vektor, zeigt auf \$E000 (ff.), bewerte Zeilenum.

06: RUN STOP entauben: 0= keine STOP-Abfrage  
1= STOP-Abfrage

8B/BC Vektor, zeigt auf \$A000 (ff.), Zeilennr.-Tab.

8D/8E Vektor, zeigt auf \$B000 (ff.), Speicherpos.-Tab.

A6: Protokoll: 0=aus

1=ein

A8: Flag:

B5/B6: Vektor, zeigt auf \$F000 (ff.), Speicherpos. für \$00105

Machen:

-Compilate laufen nur an Adressen,  
deren Low-Byte < ~\$80 ist

FB/FC: Füllen-Vektor

607E " - Flag

607F/80 Addition

6081/2 Subtraktion

6083/4 Multiplikation - Vektor

6085/6 Division

6087/8 C/R - Routine

6089/A ~~echter Programmstart~~ +3 Compilier-Adr. +3 (?)

608B/C echter Programmstart

Aufbau des Compilats:

