
WOLFGANG BAER

Software für OSI Challenger 1P /Superboard II

Programmpaket 1 / Monitor-EPROM

Das Programm versieht den ClP mit neuen Input- und Output-Routinen. Dabei bleiben die alten Routinen voll erhalten und sind durch Umsetzen der Input- und Output-Vektoren alternativ benutzbar. Viele bisharige Cursor-Programme krankten daran, dass sie nur den Input beeinflussten und die Steuersignale der Output-Routine in Konflikt mit dem Cursor kamen. Dieses neue Programm besteht aus einem Input-Teil, der die Tastatur und den Betriebszustand des Gerätes abfragt, den ASCII-Code der gedrückten Taste abspeichert und ihn bei Bedarf als Steuersignal interpretiert. In diesem Fall sendet er intern ein Steuersignal (#\$7C), das die neue Output-Routine zu den entsprechenden Sonderfunktionen veranlasst.

Laden und Initialisierung der Kassettenversion:

Nach dem Basic-Coldstart wird das Programm mit LOAD eingeladen. Es lädt einen Checksum-Lader über DATA-Zeilen, geht dann selbst in Maschinsprache und lädt den Rest des Programmes im Checksum-Format. Am Schluss ist die Leertaste zu drücken; das Gerät befindet sich danach wieder in Basic. Um ein Überschreiben des Programms zu verhindern, muss der Speicher begrenzt werden: POKE 133,0:POKE134,27, oder beim Coldstart: Memory Size 6912. Die Normalversion ist für 8 KByte RAM ausgelegt; auf Wunsch (bitte bei Bestellung angeben) sind auch Versionen für 16 und für 32 KByte erhältlich. Die Programme sind normalerweise auf Kassetten mit 300 Baud zu haben, für Diskettensysteme auch auf Minidisketten für 32 oder 24 KByte. Da die Initialisierung sehr umfangreich ist, liegt im Programm ein Initialisierungsroutine bereit, die aufgerufen wird mit POKE 11,0: POKE 12,31: X=USR(X). (31 ist die Nummer der höchsten Page im 8KByte-System; nach Speichererweiterungen müssen entsprechend höhere Zahlen eingesetzt werden.) Nach dem Warmstart genügt auch X=USR(X); falls Adressen in Page 2 geändert worden sind, kann vorher POKE 11,11 gegeben werden.

Einsetzen des Monitor-EPROMs:

Das neue EPROM wird gegen das alte Monitor-ROM ausgetauscht. Dabei sind evtl. zwei Drahtbrücken umzusetzen. Da beim alten Monitor-ROM fast zwei Drittel im ClP nicht benutzt werden, sind die neuen Routinen dort untergebracht. Die alten Routinen stehen weiterhin mit unveränderten Eingangsadressen zur Verfügung. Die Initialisierung aller Vektoren usw. erfolgt automatisch beim Coldstart. Ein Checksum-Lader gehört nicht zum Programm. Eine Speicherbegrenzung ist nicht notwendig.

Beide Versionen benutzen einen Teil der von Basic nicht benutzten Page 2.

Funktionen

(ESC) -----Makrokeys

Die Escape-Taste ordnet allen Tasten neue Funktionen zu: Der jeweils nächste Tastendruck bringt einen ganzen Basic-Befehl auf den Bildschirm und in die eingegebene Zeile. Die Befehle sind den Tasten nach Anfangsbuchstaben zugeordnet. Auch (CONTROL)- und (SHIFT)- Tasten bringen Basic-Wörter. Z.B.: (ESC)(P) bringt PRINT, (ESC)(CONTROL P) bringt PEEK, (ESC)(SHIFT P) bringt POKE.

(SHIFT O) Löschen des zuletzt eingegebenen Zeichens. Der Cursor bewegt sich dabei nach links.

(CONTROL U) Bewegung des Cursors nach oben

(CONTROL Z) Bewegung des Cursors nach unten

Mit dieser Funktion kann man unter die von Basic bedruckte letzte Zeile schreiben; dieser Bereich wird durch Line Feed nicht nach oben gerückt. Braucht man etwa häufig benutzte Textteile, dann kann man sie hier ablegen und bei Bedarf mit dem Cursor in den Input holen.

(CONTROL >) Bewegung des Cursors nach rechts; dabei wird das überstrichene Zeichen in den Input übernommen. Dadurch ist die erneute Eingabe bestehender Zeilen (mit Änderungen) möglich.

(CONTROL C) Bewegung des Cursors nach rechts; dabei wird kein Zeichen in den Input übernommen. (Verwendbar zur Streichung überflüssiger Zeichen in bestehenden Zeilen.)

(CONTROL <) Bewegung des Cursors nach links. Die bereits im Input vorhandenen Zeichen bleiben erhalten. Beim erneuten Eingeben von Zeichen werden diese in die Zeile eingefügt. Verhängern bestehender Zeilen: Vom Zeilenanfang aus mit (CONTROL >) nach rechts bis auf das nächste Zeichen nach der Einfügung; dann (CONTROL <) nach links um so viele Zeichen, wie eingefügt werden sollen; Eintippen der neuen Zeichen, dann (CONTROL >) bis zum Ende der Zeile.

(Rubout) Löschen des Bildschirms

(CONTROL L) Checksum-Lader (nur in der Kassettenversion) Die Routine liest Bandaufzeichnungen in Maschinensprache im Checksum-Format, prüft sie und legt sie im Speicher ab. Bei Fehlermeldungen ein kurzes Stück zurückspulen, erneut einspielen und (G) drücken. Bei wiederholten Fehlermeldungen an der gleichen Stelle anderen Recorder oder anderes Band versuchen. Das Programm endet beim Drücken der Leertaste und kommt zurück nach Basic, ohne bereits eingegebene Teile von Zeilen zu löschen.

(CONTROL V) View

Die Routine zeigt auf dem Bildschirm, was vom Kassettenrecorder eingespielt wird, ohne den Speicher zu verändern. Dadurch ist eine Überprüfung möglich, ob ein Programm korrekt abgespeichert wurde. Auch das Auffinden von Programmen auf der Kassette wird dadurch erleichtert. Beendigung wie beim Checksum-Lader mit Leertaste.

(CONTROL T) Trace ein

Die Routine initialisiert die Trace-Funktion, die nach dem Bearbeiten jedes Basic-Befehls die Zeilennummer ausdrückt. Gleichwertig ist der Befehl POKE 540,140, der aber nur gegeben werden darf, wenn die Vektoren nicht durch ein anderes Programm (z.B. Single Stepper) verändert wurden.

(CONTROL A) Trace aus

Die Routine setzt den Control-C-Vektor zurück und hebt Trace auf. Das ist vor allem für Direktbefehle nötig, da sonst unsinnige Zeilennummern ausgedruckt werden. Falls Trace durch (CONTROL T) initialisiert wurde, ist der Befehl POKE 540,143 gleichwertig.

(CONTROL S) Single-Stepper

Das Programm wird durch diesen Tastendruck initialisiert, wobei man Variablen und Funktionen (bis zu einer Zeile) eingeben kann, die während des Programmlaufs ausgedruckt werden sollen. Der Single-Stepper fängt an zu arbeiten, wenn im laufenden Basic-Programm der Befehl POKE 600,1 angetroffen wird. Das Programm wartet dann, bis eine beliebige Taste gedrückt wird, arbeitet einen Befehl ab, listet die Zeile, in der der Befehl steht, druckt die bei der Initialisierung angegebenen Variablen aus und wartet auf den nächsten Tastendruck. Der Befehl POKE 600,0 schaltet den Single-Stepper ab.

Alle bisher genannten Befehle können auch über Basic ausgeführt werden. Soweit in der Programmteil-Auflistung unter 11 und 12 POKES angegeben sind, können die Routinen mit X=USR(x) gerufen werden. Die Position des Cursors ist in PEEK(224) und (225) gespeichert; durch POKE 224,PEEK (224)-32 wird der Cursor vor PRINT- und INPUT-Befehlen um eine Zeile nach oben versetzt. CHR\$(124) ist ein Steuerzeichen und sollte in PRINT-Befehlen nicht verwendet werden.

Die Makrokeys-Routine über Basic zu holen, ist nicht möglich. Das gesamte Programm (Kassette oder Monitor) schaltet um auf die alten Vektoren, wenn (z.B. beim Arbeiten mit Disk-Basic oder mit dem OSI-Extended Monitor) keine ordentliche Initialisierung des ROM-Basic vorangegangen ist.

PLOT-Routine: Das Programm ist vorbereitet für eine später einzusetzende Plot-Routine, die auf ein Steuerzeichen hin (CHR\$(123)) einen von 4096 Punkten auf den Bildschirm setzt. Sobald dieses Steuerzeichen den Output erreicht, erfolgt ein Sprung an die Adresse, die mit POKE 604/605 benannt ist. Das Programm ist noch in Arbeit.

Programmierbare Funktionstasten: Die CONTROL-Tasten können durch POKES initialisiert werden, ein anzuhängendes Maschinenunterprogramm aufzurufen. Dazu ist der ASCII-Code der Taste (siehe Anhang) in den unter "Page 2" genannten Adressen der Funktionstasten 1-3 unterzubringen. Die zugehörige Sprungadresse gehört in die unter "Adresse Routine 1-3" genannten Speicherstellen. Will man z.B. das Delete-Programm über die (CONTROL D)-Taste aufrufen, dann lautet die Initialisierung POKE 591,4:POKE593,101:POKE 594,2.

DELETE: Ein Zusatzprogramm, das in den freien Teil von Page 2 passt. Als Direktbefehl per Funktionstaste aufgerufen, ermöglicht es das Löschen von mehreren Basic-Zeilen auf einmal. Es erwartet zwei Zeilennummern, durch Kommas getrennt. Statt Zeilennummern können auch Variablen (die auf Zeilennummern initialisiert sind) eingegeben werden. Die Zeilennummern müssen nicht genau stimmen; das Programm sucht die Nummern der jeweils folgenden Zeilen selbsttätig. Anschließend muss die erste zu löschende Zeilennummer noch einmal per Hand (diesmal, genau) eingegeben werden.
(Bitte bei Bestellung angeben, kostet 5.-DM extra!)

Änderungen des Bildschirmformates

Speicherstelle initialisiert mit
dez. hex dez. hex

584	\$0248	187	\$BB	ASCII-Code des Cursor-Zeichens. Es ist sinnvoll, hier nur Graphik-Zeichen zu verwenden.
585	\$0249	211	\$D3	Letzte Bildschirm-Page. Die Zahl höher zu setzen, ist sinnlos, sofern nicht per Hardware mehr Bildschirm-RAM hinzugefügt wurde. Setzt man die Zahl niedriger an, dann bleibt ein Viertel (oder mehr) des Bildschirms frei von per Basic ausgedruckten Zeilen. Die Line-Feed-Routine hat hier ebenfalls keine Wirkung. Man kann z.B. hier durch POKes oder mit dem Cursor Text ablegen, der erhalten bleibt. Allerdings wird durch die Screenclear-Routine auch dieser Teil des Bildschirms gelöscht.
586	\$024A	4	\$04	Linker Randabstand für Druckroutine. Kann geändert werden, falls z.B. der TV-Monitor nicht das volle Bildschirmformat bringt oder mehr Zeichen pro Zeile sichtbar sind. Die Zahl muss mindestens 1 betragen.
587	\$024B	32	\$20	Zeilenbreite des Bildschirms. Diese Zahl ist per Hardware festgelegt. Sie sollte also nur geändert werden, falls Eingriffe in das Gerät vorgenommen worden sind.
588	\$024C	128	\$80	Unterste Zeile der Druckroutine. Diese Zahl muss durch die unter 587 genannte Zahl teilbar sein. Sie definiert die Zeile, in der die Druckroutine arbeitet. Änderungen sind sinnvoll, wenn am unteren Rand mehr oder weniger Zeilen frei bleiben sollen. (Wirkung ansonsten gleich wie 585)
589	\$024D	29	\$1D	Rechter Randabstand der Druckroutine (vom linken Rand ab gerechnet). Erreicht der Cursor diese Position, wird ein Line Feed -Carriage Return für den Bildschirm ausgegeben, der auf Kassette und Drucker keine Wirkung hat. Diese Zahl muss kleiner sein als PEEK (587), also normalerweise nicht höher als 31.

Um ein unbeabsichtigtes Springen des Cursors beim Editieren zu verhindern, wird Terminal Width (falls nichts anderes beim Coldstart oder mit POKE 15 angegeben wurde), auf 255 Zeichen gesetzt. Diese Zahl sollte nur vorübergehend geändert werden, wenn ein Drucker mit geringer Zeilenbreite betrieben werden soll. Beim Monitor-EPROM geschieht diese Änderung noch nicht beim Coldstart, sondern erst dann, wenn die Control < -Taste berührt wird. Wurde beim Coldstart unter "Terminal width" oder mit POKE 15 eine andere Zahl angegeben, dann hat Control < keine Wirkung auf Terminal width.

Adressen Programmpaket 1+3 - Page 2

Adresse			initialisiert	
\$hex	dezimal		Kassetten-	EPROM-
			Version	
\$00E0, 00E1	224,225	Cursor-Position	\$D384	\$D384
\$0200	512	Save X		
\$0201	513	Save A (Output-Char)		
\$0218, 0219	536,537	Input-Vektor	\$1D00	\$FA00
\$021A, 021B	538,539	Output-Vektor	\$1B00	\$F800
\$021C, 021D	540,541	Control-C-Vektor	\$1F2E	\$FE8F
\$0222	546	(Single-Stepper)	#\$97	#\$97
\$0222- 0245	546-581	benutzt vom Single-Stepper		
\$0246	582	Save Input-Char		
\$0247	583	Save Y		
\$0248	584	ASCII-Code des Cursor-Zeichens	#\$BB	#\$BB
\$0249	585	Letzte Bildschirm- Page	#\$D3	#\$D3
\$024A	586	Linker Randabstand f. Druckroutine	#\$04	#\$04
\$024B	587	Zeilenbreite des Bildschirms	#\$20 (dez.32)	#\$20
\$024C	588	Unterste Zeile d. Druckroutine (Cursor Home Pos. =\$0249+\$024C+\$024A)	#\$80 (dez.128)	#\$80
\$024D	589	Rechter Randabstand f. Druckroutine	#\$1D (dez.29)	#\$1D
\$024E	590	Save Char.		
\$024F	591	ASCII-Code Funktions- taste 1	#\$7C	#\$7C
\$0250	592		#\$4C	#\$4C
\$0251, 0252	593,594	Adresse Routine 1	\$1CD9	\$F9D9
\$0253	595	ASCII-Code Funktions- taste 2	#\$7C	#\$7C
\$0254	596		#\$4C	#\$4C
\$0255, 0256	597,598	Adresse Routine 2	\$1CD9	\$F9D9
\$0257	599	ASCII-Code Funktions- taste 3	#\$7C	#\$7C
\$0258	600	Single-Stepper-Flag	#\$00	#\$00
\$0259	601		#\$4C	#\$4C
\$025A, 025B	602,603	Adresse Routine 3	\$1CD9	\$F9D9
\$025C, 025D	604,605	Zeiger auf Programm, das v. #\$7B aufger.wird	\$1B67	\$F867
\$025E	606	Save X		
\$025F	607	Save Y		
\$0260- 0264	608-612	benutzt vom Checksum- lader		

BASIC-MAKROKEYS

Dieses Programm, das in die Input-Routine eingearbeitet ist, ordnet den Keyboard-Tasten die Basic-Befehle zu, wenn die (ESC)-Taste gedrückt wird.

Taste	einfach	Control	Shift
A	ABS	ATN	AND
Q	CLEAR	CONT	
D	DATA	DIM	
E	END	EXP	
F	FOR	FRE	
G	GOSUB	GOTO	
I	IF	INPUT	INT
L	LOAD	LEN	LET
M	MID\$		CHR\$
N	NEXT	NEW	NULL
O	ON	OR	
P		PEEK	
Q		SQR	
R	RETURN	READ	REM
S	SAVE	STOP	STEP
T	THEN	TAN	TAB(
V	VAL	ASC	RND
W	WAIT	COS	SIN
X	NOT	DEF	
Ø			POKE
-	SGN		
?			PRINT
LF	LIST	LOG	
RETURN	RUN	RESTORE	
<	POS		LEFT\$
>	STR\$		RIGHT\$
Leertaste	SPC(

Diese neue Tastenbelegung gilt ab Version 6 (September 1981). Durch eine neue Tastaturroutine (§FDØØ) sind alle Kleinbuchstaben direkt durch (SHIFT) erreichbar. Bei gelöster (SHIFT LOCK)-Taste ist die Tastatur schreibmaschinenkompatibel. Abweichend von früheren Versionen gelten folgende Funktionstasten:

SHIFT Ø	↑ (Potenz)
RUBOUT	Zeichen löschen (bisher SHIFT O)
SHIFT RUBOUT	Bildschirm löschen
CONTR RUBOUT	Zeile löschen (bisher SHIFT P)
CONTR ([(Anfangs- und
CONTR)] Ende- Zeichen für indirekte Files bei Diskettenbetrieb)

Die neuen Input- und Output-Routinen gelten jetzt auch für den Betrieb im Extended Monitor. Dabei werden die Funktionstasten automatisch abgeschaltet.

ASCII-Codes der Funktionstasten - soweit nicht bereits für andere Zwecke vorgesehen (gleichzeitig muss die Control-Taste gedrückt werden)

Taste	ASCII	Taste	ASCII	Taste	ASCII
Ø	249 \$FØ	Q	249 \$F9	K	11 \$ØB
1	241 \$F1	B	2 \$Ø2	N	14 \$ØE
2	242 \$F2	D	4 \$Ø4	Q	17 \$11
3	243 \$F3	E	5 \$Ø5	R	18 \$12
4	244 \$F4	F	6 \$Ø6	W	23 \$17
5	245 \$F5	G	7 \$Ø7	X	24 \$18
6	246 \$F6	H	8 \$Ø8	Y	25 \$19
7	247 \$F7	I	9 \$Ø9	LF(ohne CTRL)	1Ø \$ØA
8	248 \$F8	J	1Ø \$ØA	LF(mit CTRL)	2Ø2 \$CA

Die Funktionstaste 1 ist so initialisiert, dass mit (CONTR)(3) das Toolkit-Menü aufgerufen werden kann. Dies kann durch POKE 501,255 unterdrückt werden.

Außerdem wird beim Warmstart getestet, ob das Toolkit bei §88ØØ vorhanden ist; in diesem Fall wird SAVE entsprechend initialisiert.