

1) Der Start

Der erste Unterschied zum Original-ROM, der Ihnen auffallen sollte, ist die geänderte Einschaltroutine:

Wenn Sie das erste Mal den Netzstecker des Computers hineinstecken, erhalten Sie das gewohnte Bild. Der Bildschirm wird für einige Sekunden schwarz, bevor die Copyrightmeldung erscheint. Während dieser Zeit wird der gesamte Arbeitsspeicher des SPEC-TRUM gelöscht und auf richtige Funktion geprüft. Wir nennen dies einen sogenannten "Kaltstart". Nachdem dies einmal durchgeführt worden ist, erfolgt bei jedem weiteren Durchlauf der Einschalt-routine ein "Warmstart". Hierbei werden nur die Systemvariablen mit den richtigen Werten initialisiert. Der Speicher wird nicht gelöscht. Nach dem ersten Einschalten gibt es also nur noch einen Reset ohne Programmverlust.

Geben Sie jetzt einmal ein: PRINTUSR 0. oder betätigen Sie den Reset-Taster. Der Bildschirm wird nicht schwarz und die Copyrightmeldung erscheint sofort. Daran erkennen Sie den Warmstart. Möchten Sie softwaremäßig einen Kaltstart durchführen, geben Sie "NEW" (an dieser Funktion hat sich nichts geändert) bzw. "PRINTUSR 100 (entspricht dem alten "PRINTUSR 0") ein.

FÜR MC-PROGRAMMIERER: Es gibt jetzt ein Unterprogramm zur Initialisierung der Systemvariablen bei 4699 dez. Denken Sie daran, daß die Einschaltroutine geändert ist, falls Sie Programme haben, die Adressen in der alten Einschaltroutine aufrufen.

2) Der erweiterte Editor

Laden Sie jetzt einmal ein längeres BASIC-Programm. Suchen Sie sich eine Zeilennummer aus und geben Sie ein: LIST <Nummer>. Der Zeilencursor, der anzeigt, welche Zeile im Augenblick die aktuelle ist, (die mit EDIT editiert werden kann) ist im ISO-ROM blinkend. So erkennen Sie auch in längeren Listings auf einen Blick die aktuelle Zeile.

Suchen Sie eine Zeile aus, die besonders lang ist. Geben Sie jetzt ein: # (Symbol-Shift und 3) gefolgt von der Zeilennummer. Die gewünschte Zeile wird direkt editiert (also ohne den Umweg

Über LIST). Beachten Sie, daß Sie den neuen EDIT-Befehl (# Nummer) nur als Direktkommando eingeben können. Es ist also nicht möglich, diesen Befehl in BASIC-Zeilen zu verwenden.

Drücken Sie jetzt Caps-Shift und 6. Der Cursor wandert eine Zeile nach unten. Innerhalb einer editierten Zeile können Sie bei ISO-ROM den Cursor in alle 4 Richtungen bewegen. Beachten Sie, daß bei Caps-Shift und 6 bzw. Caps-Shift und 7 der Cursor meist nicht genau oberhalb bzw. unterhalb der bisherigen Position landet, weil er nicht mitten in ein Keyword gesetzt werden kann. Wenn mit "Cursor nach unten" das Ende der Zeile erreicht ist, wird wie bisher der Zeilencursor nach unten verschoben (d.h. die aktuelle Zeilennummer wird um eins erhöht). Ebenso wird am Zeilenanfang der Zeilencursor nach oben verschoben.

Verschieben Sie den Cursor jetzt wieder zum Zeilenbeginn und löschen Sie das Keyword. Bewegen Sie den Cursor dann wieder mitten in die Zeile und drücken Sie <Enter>. Wegen des fehlenden Keywords enthält die Zeile jetzt einen Syntaxfehler. Beachten Sie, daß der Cursor jetzt automatisch zu der Stelle mit dem Fehler (direkt neben das blinkende Fragezeichen) fährt.

3) Deutsche Umlaute

Nach ASCII-Norm. Sie finden:

ä auf {
å auf [
ö auf |
ø auf \
ü auf }
Û auf]
ø auf ~!

4) Löschen der neuen Systemvariablen

Nach einem NMI oder einem Reset bleiben die von Interface 1 oder Beta-Disk Interface geschaffenen neuen Systemvariablen zunächst erhalten. Beim Versuch auf das Microdrive zuzugreifen, kommt es zur Fehlermeldung "Microdrive not present". Geben Sie in einem solchen Fall "\$" (Symbol-Shift und 4) gefolgt von <ENTER> ein.

5) Verbessertes Cat für Microdrive

Zum Aufruf der erweiterten CAT-Routine geben Sie bitte ein:

```
CAT USR 15360 <ENTER>
```

Als erstes erscheint dann in der obersten Zeile des Bildschirms ein Strichmuster, die Sektormap. Jeder senkrechte Strich zeigt den Zustand eines Sektors der Cartridge an. Dabei bedeutet ein Strich in der INK-Farbe: Der Sektor ist nicht zu beschreiben, d. h., er ist belegt oder nicht formatiert. Die letzten Sektoren der Cartridge (Striche ganz rechts) werden nie formatiert.

Das verbesserte CAT liefert dann die Headerinformation der auf der Cartridge vorhandenen Files. Bei Basic wird die Autostartzeile (wenn vorhanden), bei Arrays der Variablenname und bei Codeblöcken Startadresse und Länge ausgegeben.

Eine mit OPEN eröffnete Schreibdatei wird durch OPEN # gekennzeichnet. Die Informationen über Art und Länge der Files werden dabei so ausgegeben, wie man Sie auch nach dem SAVE-Befehl eingeben müßte. Beispiel:

```
Abgesaved wurde mit:   SAVE "Name" CODE 40000,711
```

```
Anzeige der CAT-Routine      Name CODE 40000,711
```

Es lassen sich maximal 22 Files katalogisieren. Es ist nicht ratsam, den Vorgang mittels BREAK zu unterbrechen, bevor die ersten Headerinformationen auf dem Bildschirm erscheinen.

6) USR-Taste

Wenn Sie bei ihrer Bestellung eine gewünschte USR-Adresse angegeben haben, können Sie die entsprechende Routine jetzt durch ". <ENTER>" aufrufen. Ansonsten nimmt der Computer diesen Befehl nicht an.

7) Vereinfachte Syntax

Durch ein vorangestelltes "!" wird -je nach ihrer Bestellung- entweder die Microdrive- oder die Beta-Disk-Syntax generiert.

7) Beta-Disk Lade/Save-Befehle

Wenn ein Beta-Disk Interface angeschlossen ist, wird automatisch das DOS aufgerufen, wenn Sie vor die entsprechenden Befehle ein "!" setzen. Statt "RANDOMIZE USR 15363; REM: CAT" schreiben Sie also z.B. einfach "! CAT". Enthielt der nach dem "!" folgende Befehl einen Syntaxfehler, erhalten Sie die Fehlermeldung "C Nonsense in BASIC". Findet das DOS einen anderen Fehler, wird die Fehlermeldung "E Invalid file name" ausgegeben. Den Beta-Disk Aufruf "!" können Sie ohne weiteres auch in Programmzeilen verwenden. Beachten Sie aber, daß in diesem Punkt dann keine Softwarekompatibilität mehr gegeben ist, d.h. das entsprechende Programm ergibt auf einem SPECTRUM ohne ISO-ROM die Fehlermeldung "Nonsense in BASIC". Ohne angeschlossenem Beta-Disk Interface erhalten Sie auch bei ISO-ROM diese Fehlermeldung.

8) Fehlerkorrektur / Syntaxerleichterung

Die Fehler des Original-ROM's sind bei ISO-ROM beseitigt. Insbesondere können Sie die SCREEN\$-Funktion jetzt uneingeschränkt in Stringzusammensetzungen verwenden und die Zahl -65536 wird im richtigen Format auf dem Kalkulatorstapel abgelegt. CHR\$ 9 ist jetzt ein druckbares Steuerzeichen. Es verschiebt die Printposition um einen nach rechts.

In einigen Punkten ist die Syntaxprüfung jetzt etwas weniger streng:

Um ein Programm mit Auto-Start am Anfang abzusaven, können Sie jetzt auch "SAVE <Name> LINE" statt "SAVE <Name> LINE 0" schreiben (was fälschlicherweise bereits im Original Handbuch steht). Statt "PAUSE 0" können Sie jetzt auch einfach "PAUSE" schreiben.

7b) Microdrive Befehle

Die häufig wiederkehrende Zeichenfolge 'x"m";1;' bzw. '"m";1;' wird im ISO-ROM automatisch generiert, wenn Sie einfach vor die entsprechenden Befehle ein '!' schreiben. Statt 'LOAD x"m";1; "Name"' schreiben Sie also einfach '! LOAD "Name"'. Die Befehle LOAD, SAVE, VERIFY, MERGE, ERASE und FORMAT werden so enorm vereinfacht. Wichtig ist, daß die Befehle sich auf das Microdrive Nummer 1 beziehen müssen. Da die 'richtige' Syntax durch das '!' generiert wird, können Sie diese Befehle bei voller Softwarekompatibilität auch in Programmzeilen verwenden.

8) Der Monitor

ISO-ROM hat einen eingebauten Maschinensprachenmonitor. Zum Aufruf dieses Monitors gibt es zwei Wege:

a) Vom BASIC aus geben Sie den Befehl "Q" (Symbol-Shift und 2) gefolgt von <Enter> ein.

b) Der weitaus wichtigere Weg ist der Aufruf über NMI.

NMI (non maskable interrupt) nennt man die Unterbrechungsart der CPU mit der höchsten Präferenz. Wenn der entsprechende Eingang der CPU aktiviert wird, hält jedes laufende MC-Programm an und verzweigt zum Monitor. Betätigen Sie jetzt einmal den NMI-Taster Sie sehen auf dem Bildschirm nebeneinander vier Spalten mit Adressen (invers gedruckte 2 Byte Hex-Zahlen). Rechts daneben sehen Sie jeweils die Inhalte der entsprechenden Speicherstellen. Die ersten Bytes der vorgegebenen Seite enthalten wichtige Informationen: In AASC findet sich der Wert des Programmcounters, bei dem der NMI auftritt (wenn diese Adresse nicht im ROM liegt). In AASC ist der Wert des Stapelzeigers zum Zeitpunkt des NMI's. In ACSC ist der Wert der Systemvariablen CHARS gerettet. Beachten Sie, daß der Monitor für sich selbst CHARS immer auf den Character-Set im ROM setzt.

In den untersten zwei Zeilen des Bildschirms sehen Sie die Inhalte der CPU-Register und Flags zum Zeitpunkt des NMI. In der zweituntersten Zeile finden Sie dabei nebeneinander die Inhalte von a, bc, de, hl und iy-Register. In der untersten Zeile die der entsprechenden Alternativregister (a' bis hl') und des ix-Registers. Rechts daneben sind die Inhalte der Flags z, c und s. (letzteres als p oder m) verzeichnet. Bei gesetzten Flags blinken die entsprechenden Buchstaben. Mit verzeichnet finden Sie auch den Zustand des maskierbaren Interrupts IRQ. "ei" bedeutet, der Interrupt war freigegeben, "di" er war gesperrt. Beachten Sie daß der Monitor für sich den IRQ immer freigibt.

Mitten auf dem Bildschirm sehen Sie einen blinkenden Cursor. Den Inhalt dieser Speicherstelle können Sie ändern. Wenn Sie nach der Eingabe der ersten Hex-Ziffer diese korrigieren wollen, drücken Sie bitte <Delste> (Caps-Shift und 0). Nach beendeter Eingabe drücken Sie <Enter>. Es wird eine neue Seite mit um eine erhöhte Anfangsadresse ausgegeben. Der Cursor bleibt aber an seiner Position. Wenn Sie Caps-Shift und <Enter> drücken, wird die Anfangsadresse um einen vermindert.

Zum Ansehen von Adressen, die weit von der bisherigen Startadresse entfernt sind, ist das wiederholte Drücken von <Enter> bzw. Caps-Shift und <Enter> viel zu langsam. Drücken Sie daher "N" (wie "nächste Seite"). Sie erhalten die Seite, die genau dort anfängt, wo die bisherige aufhört. So können Sie "nach vorn umblättern". Umgekehrt erhalten Sie mit "V" die vorige Seite (umblättern rückwärts).

Eine andere Möglichkeit ist das direkte Eingeben einer Startadresse. Drücken Sie "S" für "Startadresse". Der Cursor springt in die Spalte mit den Adressen. Geben Sie jetzt die Adresse ein, auf die der Cursor zeigen soll. Beachten Sie, daß 2 Byte-Zahlen im Monitor immer so ausgegeben werden (und auch eingegeben werden müssen), wie sie im Speicher stehen, d.h. das 10-Byte steht grundsätzlich vorne.

Schreiben Sie jetzt an einer beliebigen freien Stelle im RAM ein Maschinenprogramm. Drücken Sie danach "R". Das Wort "RUN" erscheint und der Cursor springt in die Adressenspalte. Geben Sie die Startadresse Ihres MC-Programms ein und drücken Sie <Enter>. Ihr Programm wird ausgeführt. Nach dem ret-Befehl springt das Programm in den Monitor zurück. Die ausgedruckten Register und Flags gelten jetzt für den Zeitpunkt der Rückkehr aus Ihrem Programm in den Monitor.

Um den Monitor wieder zu verlassen, drücken Sie Caps-Shift und <Break>. Ein Wort der Warnung:

Mit dem eingebauten Monitor können Sie zwar jedes Programm anhalten und jeden Absturz auffangen, das heißt aber lange noch nicht, daß eine Rückkehr ins BASIC möglich ist. Gerade bei Systemabstürzen können umfangreiche Neu-Initialisierungen von Systemvariablen notwendig werden. Notfalls können Sie solange den Monitor wieder aufrufen, bis alles bereinigt ist.

Allgemein ist der Monitor so geschrieben, daß er möglichst wenig mit anderen Programmen kollidieren kann. An Variablen benutzt er nur die Bytes von A85C bis AF5C in MEMBOT. Verändert werden folgende Systemvariablen: CHARS, FLAGS, FLAGS2, DFSZ, PFLAG und natürlich DFCC/DFCCL und SPOSN/SPOSNL.

C O P Y R I G H T - H I N W E I S

Sämtliche im Original-ROM nicht vorhandenen Programmteile des ISO-ROM's, insbesondere alle Bytes ab 14446, sind das Copyright von INDIVIDUAL SOFTWARE. Eine Weitergabe dieser Programmteile an Dritte oder das Kopieren Ihres EPROM's als Ganzes ist ausdrücklich untersagt.

VERSTÜSSE GEGEN DIESE BESTIMMUNG WERDEN STRAFRECHTLICH VERFOLGT!

Wir machen darauf aufmerksam, daß jedes ISO-ROM eine geheime Seriennummer in sich trägt, an der notfalls der Kunde zu identifizieren ist.

9) Anhang

a) Hardware-Kompatibilität

Viele Hardwareerweiterungen, vor allem Druckerinterfaces, die ohne zu ladende Steuersoftware auskommen, benutzen wie ISO-ROM den Erweiterungsbereich des ROM's ab Adresse 14446. Dabei wird der entsprechende Bereich des ROM's durch das Interface stillgelegt und an seiner Stelle das in einem EPROM vorhandene Steuerprogramm eingeblendet. Die Erweiterungen des ISO-ROM's werden somit durch das Steuerprogramm überschrieben. In den meisten Fällen hat sich folgendes Vorgehen bewährt, um beide Erweiterungen zu nutzen:

Bauen Sie in die +5 V-Leitung des Interface einen Schalter ein. Lassen Sie danach das Druckerinterface immer angesteckt, schalten Sie es jedoch nur ein, wenn gedruckt werden soll. Bei eingeschaltetem Interface benutzen Sie nur die Funktionen und Befehle die auch im normalen ROM vorhanden sind. Insbesondere benutzen Sie nicht die "Cursor nach oben"- und "Cursor nach unten"-Funktion des Editors.

b) Softwarekompatibilität

Einige wenige Spielprogramme laufen auf dem ISO-ROM nicht bei angeschlossenem Kempston Joystick-Interface. Das hat nichts mit irgendwelchen Einsprungsadressen zu tun (einige Spiele z.B. Chequered flag oder die ersten Fassungen von Ghostbusters laufen auch mit dem Original-ROM und Kempston Joystick-Interface nicht), sondern resultiert aus folgendem Umstand: Die Programme benutzen den Interruptmodus 2 und legen die entsprechende Adressentabelle im ehemals leeren Erweiterungsbereich des ROM's an. (In dem leeren Bereich stand überall das Byte 255, entsprechend einer Adresse von 65535)

Einfache Abhilfe:

Die Probleme treten nur beim Original Kempston Joystick-Interface auf. Die meisten kempston-kompatiblen Joystick-Interfaces (z.B. dk'tronics) haben das Problem nicht.

10) BACKUP-Routinen

Es gibt drei verschiedene Backup-Routinen, die sich nur durch das verwendete Speichermedium unterscheiden:

Backup M (BM) ... saved auf Microdrive und Opus-Disk
Backup B (BB) ... " " Beta-Disk
Backup K (BK) ... " " Kassette

Bedienung:

Laden Sie das zu kopierende Programm völlig normal ein und unterbrechen Sie es durch Drücken des NMI-Tasters (vorzugsweise an einer ruhigen Stelle im Programmablauf, z.B. Menü- oder Highscore-tabelle).

Sie hören dann einen Ton, der die Aktivierung der Backup-Routine anzeigt und Ihnen auch Zeit gibt, den Taster wieder loszulassen. Im obersten Bildschirmdrittel werden einige Bytes mit Daten überschrieben. Danach folgt eine kurze Pause. Während dieser Zeit wird der gesamte Speicher auf nicht belegte Bytes untersucht und diese gegebenenfalls entfernt. Der Speicherinhalt wird also "komprimiert".

Im unteren Bildschirmdrittel erscheint die Meldung "ISO-ROM 8" gefolgt von der Version (BM, BB oder BK) und einem blinkenden Cursor. Sie können jetzt, genau wie Sie es von BASIC her gewohnt sind, Befehle eingeben, um die verschiedenen Optionen der Backup-Routine anzusteuern. Dabei sollten Sie aber die "Cursor nach oben", "Cursor nach unten" und "EDIT"-Taste nicht benutzen. Hier nun eine Liste der Optionen mit Beispielen für die Syntax:

OPTION	SYNTAX	Bemerkungen
SAVE	SAVE "ALIEN 8" oder SAVE ALIEN 8	Bei Backup K vorher den Rekorder starten!
POKE	POKE 51736,0	
CLS	CLS	
RETURN	RETURN	

Was leisten nun die einzelnen Optionen?

SAVE: Es werden drei Teile auf Ihr Speichermedium abgesaved, nämlich ein BASIC-Loader, (der den von Ihnen eingegebenen Namen erhält) und zwei Codeblöcke. Bei diesen Codeblöcken wird an den eingegebenen Namen noch eine "1" bzw. "2" angehängt. Folgende Fehler können während des Absavens auftreten:

- a) Nicht genügend Platz auf der Cartridge / Diskette
- b) Name auf der Cartridge / Diskette bereits vorhanden
- c) Keine Cartridge / Diskette eingelegt

(Bei BACKUP K gibt es diese Fehler natürlich nicht)

- d) BREAK-Taste gedrückt (alle Versionen)

Bei einem Fehler macht sich der Computer optisch und akustisch bemerkbar und geht zurück zur Befehlseingabe. Nach dem Absaven läuft das unterbrochene Programm automatisch weiter.

Zur Übertragung des abgesaveden Programms auf ein anderes Speichermedium (wichtig für BACKUP K) können Sie den BASIC-Loader MERGE'n und seine Variablen abfragen:

Variable R --> Verwendetes RAMTOP. Nicht verändern!

Variable B --> Länge des ersten Codeblocks. Beginn ist 16384

Variable S --> Beginn des zweiten Codeblocks.

Variable L --> Länge des zweiten Codeblocks.

POKE: Führt den entsprechenden POKE aus. Ein Fehler tritt auf, wenn Adresse oder Wert kleiner als 0 oder größer als 65535 sind.

CLS: Wenn es in Ihrem Programm eine Stelle gibt, an der auf einen Tastendruck hin der gesamte Bildschirm neu aufgebaut wird, können Sie mit der CLS-Funktion dafür sorgen, daß der Screen nicht mit abgesaved wird. So sparen Sie weiteren Speicherplatz.

RETURN: Rückkehr zum unterbrochenen Programm (z.B. nach POKE)

BACKUP Q, die Quicksave-Backuproutine:

Bei dieser Backup-Routine macht sich der Computer nach dem Druck auf den NMI-Taster durch regelmäßige Streifen auf dem BORDER bemerkbar. Durch Drücken einer der Tasten von 0 bis 9 wird Ihr Programm mit einer der folgenden Geschwindigkeiten gesaved:

QS-Stufe:	0	1	2	3	4	5	6	7	8	9
Baudrate:	2990	3180	3390	3640	3920	4080	4250	4640	5110	6400

Ab etwa Stufe 6 benötigen Sie Chromdioxid-Kassetten und einen entsprechenden Rekorder.

Bei der Backup Q-Version mit einer festen Baudrate können Sie zum Saven eine beliebige Taste drücken.

Das Wiedereinladen geschieht bei beiden Backup Q-Versionen mit dem Befehl @ (Symbol-Shift + 2) <ENTER>.

11) Drucken mit variabler Zeichenbreite

Bei der Monitorversion des ISO-ROM's können Sie mit 5 verschiedenen Zeichenbreiten drucken. Somit haben Sie bis zu 64 Zeichen pro Zeile. Folgende Zeilenlängen stehen zur Verfügung:

Zeichenbreite:	8	7	6	5	4
Zeichen pro Zeile:	32	36	42	51	64

Die Modi mit mehr als 32 Zeichen pro Zeile sind reine Textmodi. Farbenweisungen wie INK oder PAPER werden zwar gespeichert, aber erst beim Drucken im 32-Zeichen-Modus wieder ausgeführt.

Das Anwählen und Steuern der Druckmodi geschieht durch einfache POKE's und zwar in die STREAMS-Information des Kanals # 15. Wenn Sie also mit mehr als 32 Zeichen pro Zeile drucken wollen, können Sie nicht gleichzeitig den Kanal # 15 benutzen. Beispiele:

POKE 23605,-5

wählt eine Zeichenbreite von 5 an. Das Minuszeichen unterscheidet die Speicherstelle 23605 von einem geöffneten Kanal # 15. Den 32-Zeichen-Modus erreichen Sie einfach durch POKE 23605,0.

POKE 23604,3

legt die Druckposition genauer fest. Es wird an der aktuellen Stelle im Bildschirm (die z.B. mit PRINT AT festgelegt ist) beginnend mit dem 3. Bit gedruckt. So können also durch PRINT AT... und POKE 23604,... insgesamt 256 Druckspalten angewählt werden.

4) Besonderheiten des Toolkit-ROM's

Das Toolkit-ROM enthält keine eigene NMI-Routine. Bei Betätigung des NMI-Tasters wird vielmehr zur Haupteingabeschleife des BASIC's verzweigt. Es erscheint die Meldung "0 OK,".

Die Befehle des Toolkits werden nicht durch "Symbol-Shift + Zifferntaste" wie beim Monitor aufgerufen, sondern mit eigenen Befehlswörtern. Diese Befehlswörter beginnen alle mit einem Punkt. Die Befehle können durch einen Punkt, ein Space oder (wenn keine Parameter folgen) durch ein <ENTER> abgekürzt werden. Bei den meisten Toolkit-Befehlen müssen nach dem Befehl noch ein oder mehrere Parameter eingegeben werden (im folgenden jeweils in eckigen Klammern aufgeführt). Wenn mehrere Parameter folgen, müssen sie jeweils durch ein Komma getrennt werden. Als Parameter sind nicht nur Zahlen sondern beliebige numerische Ausdrücke erlaubt. Alle Toolkit-Befehle dürfen auch in BASIC-Zeilen verwendet werden.

Folgende Befehls sind implementiert:

.edit <Zeilennummer>

Die Zeile wird direkt editiert. (Entspricht dem #-Befehl im Monitor-ROM)

.clr

Die Systemvariablen von Interface 1 oder Beta-Disk werden gelöscht. (Entspricht dem \$-Befehl im Monitor-ROM)

.renum <Beginn>,<Schrittweite>,<Neuer Beginn>

Nummeriert das BASIC-Programm ab der Zeile <Beginn> bis zum Ende neu. Dabei werden die Zeilennummern jeweils um <Schrittweite> erhöht. Die Zeile <Beginn> erhält die neue Zeilennummer <Neuer Beginn>. Dabei werden folgende Zeilennummern als Ersatzwert genommen, falls diese nicht spezifiziert sind:

Statt <Beginn> --> Erste Zeilennummer des Programms

" <Schrittweite> --> 10

" <Neuer Beginn> --> <Beginn>

BASIC-Befehle, wie GOTO, GOSUB, RESTORE u.s.w., die sich auf die unnummerierten Zeilen beziehen, werden automatisch geändert. Berechnete GOTO's werden nur dann richtig geändert, wenn keine Variable als Anfangszeile darin vorkommt. Beispiel:

GOTO 100 + 10 * x

wird richtig umgewandelt, wenn die Schrittweite gleichgeblieben ist, aber

GOTO A + 10 * x

kann nicht umgewandelt werden. Hier müssen Sie selber im Programm die Definition der Variablen A ändern.

Die Fehlermeldung "A Invalid argument" tritt beim .renum-Befehl unter folgenden Bedingungen auf:

a) Wenn eine Zeilennummer größer als 9999 werden würde.

Beispiel: .renum 9990,20

b) Wenn eine Zeilennummer bereits existiert.

Beispiel: .renum 1000,10.100 ;wobei Zeile 100 bereits existiert.

.delete <Beginn>,<Ende>
Löscht alle BASIC-Zeilen von <Beginn> bis einschließlich <Ende> aus dem Programm. Beide Werte müssen eingegeben werden. Fehlermeldung A, wenn <Ende> kleiner als <Beginn> ist.

.auto <Beginn>
Durch diesen Befehl werden die Zeilennummern beim Eingeben eines BASIC-Programms automatisch vorgegeben. Wenn Sie den Wert von <Beginn> nicht spezifizieren, wird als Ersatzwert 10 genommen. Um den Auto-Modus wieder zu verlassen, löschen Sie einfach die vorgegebene Zeilennummer mit DELETE (Caps-Shift und 0).

.onerror <Zeilennummer>
Mit diesem Befehl haben Sie die Möglichkeit, eine eigene Fehlerbehandlungsroutine in BASIC zu schreiben. Nachdem dieser Befehl ausgeführt ist, führt jeder Fehler nicht mehr zur Ausgabe einer Fehlermeldung, sondern dazu, daß das BASIC-Programm bei <Zeilennummer> weiterarbeitet. Dort können Sie dann Ihre eigene Fehlerbehandlung durchführen. Folgende Variablen können Ihnen dabei helfen:

PEEK 23610 ergibt die Fehlernummer - 1 (bzw. 255, bei 0 OK)
PEEK 23662 + 256 * PEEK 23663 ergibt die Zeilennummer, bei der der Fehler auftrat

PEEK 23664 ergibt die Statementnummer innerhalb dieser Zeile, bei der der Fehler auftrat.

Ganz wichtig: Um die Fehlerbehandlung durch Ihre Routine wieder abzustellen, lassen Sie den Befehl .onerror (also ohne jede Zeilennummer) abarbeiten. Wenn Sie dies vergessen, bleibt der Rechner nach Beendigung Ihres Programms in Ihrer Fehlerroutine hängen (0 OK !!). Wenn das Programm also nach Ihrer Fehlerroutine stoppen soll, beenden Sie diese mit .onerror: STOP.

Wenn das Programm nach Abarbeitung Ihrer Fehlerroutine im Programmablauf weitermachen soll, beenden Sie die diese mit:

POKE 23618, PEEK 23662: POKE 23619, PEEK 23663: POKE 23620, PEEK 23664+1 (bzw. wenn der Befehl mit dem Fehler wiederholt werden soll ...PEEK 23664).