

## Action Replay

*Diesmal erwartet Sie in der Software-Corner eine reichhaltige Sammlung von Kniffen rund um das Modul "Action Replay". Mit unseren Tricks umschiffen Sie all seine Tücken.*

von Sebastian Theiß

### Der Freezer

Möchte man vom Freezer aus einen Reset zum Fastload ausführen, genügt die günstig gelegene Tastenkombination <Commodore + Linkspfeil>. Dabei geht aber der Speicherbereich von \$0800-\$0900 verloren, ein OLD wirkt also nicht.

Ein Reset wird ebenfalls ausgeführt, wenn man (meist versehentlich) nach Ausgabe des Directorys per Taste <D> bei der Meldung "Press Any Key" die Commodore-Taste drückt. Soll ein Basic-Programm nach einem Absturz des Prozessors (z.B. durch BRK im Maschinencode) gerettet werden, sollten Sie folgendermaßen vorgehen: Mit <F3> wird der Freezer kurzzeitig verlassen, und sobald der Bildschirm erscheint (meist hört man parallel dazu ein Knacken im Lautsprecher), wird die Restore-Taste in gewohnter Weise betätigt. Im Monitor überzeugt man sich dann, ob der Tastendruck erkannt wurde (der Programmzähler steht dann auf \$EBD4), wenn nicht, wird der Vorgang wiederholt. Ansonsten kann man mit <F3> den Freezer verlassen; der Computer verhält sich dann wie nach einem illegalen Code (z.B. \$02). Mit <Commodore + Resetknopf> und OLD ist das Basic-Programm gerettet. Diese etwas komplizierte Prozedur kann unter Umständen stundenlange Arbeit retten, wenn das Basic-Programm nicht gespeichert und das Maschinenprogramm nicht getestet wurde.

### Der Turbo-Lader

Manchmal ist es ärgerlich, wenn ein Programm nicht mit der Cartridge zusammenarbeitet. Man ist dann gezwungen, das Modul abzuschalten. Damit man trotzdem nicht auf den Turbo-Lader verzichten muß, gibt es aber Tricks:

1. Man schaltet das Modul mit <CTRL> + Reset ab. Im Freezer-Menü klinkt man noch vor dem Laden mit <L> den Turbo-Lader wieder ein. Nach <F3> kann man das Programm dann laden und starten.

2. Man lädt das Programm mit <F2> und schaltet die Cartridge dann mit OFF ab. Danach startet man mit <F7>. Dies funktioniert natürlich nur, wenn das Programm nicht mit einem Autostart versehen ist.

3. Wenn nichts hilft, lädt man wieder das Programm mit <F2> und gibt mit <F5> das Listing aus. Programme, die sich mit den vorhergehenden Tricks nicht starten ließen, enthalten oft nur eine Zeile mit einem SYS-Befehl. Man merkt sich also dessen Argument (die Startadresse), schaltet das Modul mit <CTRL> + Reset aus und gibt SYS (Startadresse) und <Return> ein. Dies funktioniert in 99% aller Fälle.

### Der Pokefinder

Der Pokefinder soll es ermöglichen, in einem Spiel unendlich viele Leben zu besitzen. Leider funktioniert er in neueren Spielen

immer seltener. Manchmal liegt das aber schlicht daran, daß die Zahl der Leben im ASCII-Format abgelegt ist: Im Pokefinder gibt man z.B. bei drei Leben den ASCII-Code von "3", also 51 ein und startet den Suchvorgang wie gewohnt. Dieser Trick funktioniert in etwa 30-40% der Fälle, wenn der "normale" Pokefinder versagt.

Oft ist es auch so, daß die Zahl der Leben nicht mit DEC, sondern mit SBC heruntergezählt wird; diese Fälle werden vom Pokefinder natürlich nicht erkannt und erfordern zum Eliminieren einige Maschinensprachekenntnisse.

Der Pokefinder verweigert übrigens seine Arbeit, wenn man als Anzahl der Leben 0 eingibt, aber auch, wenn die Zahl eine 0 enthält. Bei zehn Leben etwa fängt der Finder erst gar nicht an zu suchen.

### Basic im Freezer

Neben der Option, einzelne Basic-Befehle im Freezer einzugeben, gibt es die Möglichkeit, kleine Basic-Programme zu schreiben, die man nicht mit Einzelbefehlen ausführen kann, z.B. Schleifen mit IF-THEN-Abfragen oder eine Spezial-Byte-Suchroutine. Im Monitor gibt man einfach ein:

```
.BPOKE 768,143
```

Nun stehen etwa 120 Bytes für Basic zur Verfügung. Genug für kleine Programme, wenn man auf Kommentare und Print-Texte verzichtet. Alle Funktionen, mit Ausnahme des Basic-Toolkits, stehen zur Verfügung. Eingegebene Programme können zwar gespeichert, vorbereitete Routinen aber nicht geladen werden (sie kommen ins "echte" RAM des gefreezten Programms).

Zurück zum Freezer-Menü kommen Sie einfach durch erneute Betätigung des Freeze-Knopfs. Man sollte sich übrigens nicht durch die Meldung "38911 Basic Bytes free" zu speicheraufwendigem Programmieren oder zur Benutzung von Array-Variablen verleiten lassen, denn hat man einmal sein Speicherlimit überschritten, stürzt das Modul bei der Rückkehr ab.

### Programmierung

In einigen Spielsimulationen hat der Spieler die Aufgabe, Kapital zu verwalten. Das Hauptproblem dabei ist, daß die zur Verfügung stehenden Mittel meistens viel zu knapp sind. Mit Action Replay kann man sich auf "legale" Weise mehr Geld verschaffen.

Zuerst ist es wichtig, sich den genauen Kontostand zu merken und zwar mit allen angegebenen Stellen. Aus Gründen der Verwaltung wird z.B. 320,95 DM intern fast immer als 32095 abgelegt und das Komma nur in der Anzeige eingesetzt (Auch hier hilft ab und zu die Suche nach den ASCII-Codes des Kontostands, wie z.B. H 0000 FFFF "32095" oder auch H 0000 FFFF "320,95"). Im Freezer geht man in den Monitor und konvertiert seinen Kontostand zunächst ins Hex-Format mit N32095. Ist dies nicht möglich, weil die Zahl größer als 65535 ist (z.B. 300000), muß man sie in Hi- und Lo-Bytes aufteilen:

```
.B?300000/65536
```

```
4.57763672 (Hi-Byte)
```

```
.B?300000-(4*65536)
```

```
37856
```

```
.N37856
```

```
$93E0 %1001001111100000 37856 (Lo-Bytes)
```

Damit haben wir als Hi-Bytes \$0004 und als Lo-Bytes \$93E0. Nun wird der Speicher nach diesen Bytes im Lo-Hi-Format (.H 0000 FFFF E093 0400) und im Hi-Lo-Format (.H 0000 FFFF 0004 93E0) durchsucht. Die gefundenen Adressen können verändert werden (.M Adresse) und man prüft dann wieder seine Geldmittel. Achtung: Das höchstwertige Byte enthält im Bit 7 oft das Vorzeichen; werden hier Werte größer \$7F eingetragen, steckt man in den roten Zahlen!

### Basic-Linker

Manchmal möchte man zwei Basic-Zeilen zu einer verbinden. Das kann man z.B. damit erreichen, daß man die zweite Zeile hinter der ersten neu eingibt. Vor allem bei Zeilen mit vielen Steuerzeichen ist dies aber nicht unbedingt angenehm. Mit AR geht es einfacher, wenn beide Zeilen samt Zeilennummer jeweils weniger als 40 Zeichen Platz benötigen:

1. Man gibt die Zeilen mit LIST so aus, daß sie direkt übereinander stehen.
2. Hinter der ersten Zeile wird solange <SPACE> gedrückt, bis der Editor eine Leerzeile zwischen den beiden Programmzeilen einfügt.
3. Drücken Sie dreimal <Cursor Down> und <F6>, um in den Freezer zu gelangen, dort wählen Sie dann mit <T> den Texteditor an.
4. Mit dem Cursor fahren Sie dann zwischen die Programmzeilen und drücken <CTRL+D>, anschließend verlassen Sie mit <RETURN> und <F3> den Freezer.
5. Die Zeilennummer der zweiten Zeile muß entfernt und die entstandene neue Zeile per <RETURN> in den Speicher übernommen werden.

Wie erwähnt, eignet sich dieses Verfahren nur für Zeilen mit vielen Steuerzeichen, oder zu Lernzwecken über "logische" und "echte" Bildschirmzeilen. Logisch, echt?

Echte Bildschirmzeilen sind alle sichtbaren, also 25 Stück, à 40 Spalten. Vom Basic-Interpreter werden unter bestimmten Umständen zwei echte Bildschirmzeilen zu einer "logischen" zusammengefaßt, normalerweise sind aber alle echten Zeilen auch logische. Logische Bildschirmzeilen können also 40 oder 80 Spalten haben, daher kann es zwischen 13 und 25 logische Bildschirmzeilen auf einem Bildschirm geben. Logische Bildschirmzeilen erkennt man daran, daß der Cursor – wenn man auf einer solchen mit <RETURN> bestätigt – zwei Zeilen nach unten springt. Wird eine logische "lange" Bildschirmzeile gescrollt, werden am unteren Rand zwei Leerzeilen eingefügt. Bei diesem Trick haben wir zunächst eine logische "lange" Bildschirmzeile hervorgerufen und dann alle folgenden echten Bildschirmzeilen mit dem Texteditor des Moduls verschoben.

### JSR verfolgt

Unter bestimmten Umständen kann man vom Freezer aus verfolgen, von woher die laufende Routine angesprungen wurde. Bedingungen sind: Es muß eine ordnungsgemäße Rücksprungadresse vorliegen, d.h. die Routine muß mit JSR aufgerufen worden sein, und es dürfen keine Stapelmanipulationen vorgenommen worden sein (PHA, PHP usw.). In der Registeranzeige findet man den Stapelzeiger unter der Bezeichnung SP (Stack-Pointer). Dieser Zeiger läuft abwärts, d.h. der Pointer steht normalerweise bei \$E0 bis \$FC.

Geben Sie also ein: .M 01 (Stapelzeiger), für \$FB zum Beispiel also .M 01FB. Die ersten beiden ausgegebenen Bytes erhöht man um 1 und erhält direkt (im Hi-Lo-Format) die Sprungadresse. Beispiel:

```
.M 01FB
.:01FB 60 02 AA ...
```

Die Rücksprungadresse wäre also \$6003. Verringert man die Sprungadresse um 3 und findet dort einen JSR-Befehl (D Adresse-3 -), wurde die aktuelle Routine wohl von dort aus aufgerufen.

### Der Diskmonitor

Da er im Handbuch nur schlecht kommentiert wurde, wird der Disk-Monitor von den meisten Usern nur selten eingesetzt, obwohl er ziemlich leistungsfähig ist.

Die Standard-Blockbefehle lauten:

@ BR tr se (pa) – Block auf Spur tr, Sektor se lesen nach pa00, falls angegeben, sonst nach \$CF00.

@ BW tr se (pa) – Analog zu BR; Block schreiben.

@ ME addr – Programm im Floppyspeicher ab addr ausführen.

Neben den Blockbefehlen existiert die unscheinbare, aber leistungsfähige Funktion @\*(x(y)). Für x und y werden zwei einstellige Zahlen eingesetzt, die das Gerät zum Lesen/Schreiben kennzeichnen. Dabei gelten:

- 0 – Computer,
- 8 – Floppy #8,
- 9 – Floppy #9, soweit vorhanden.

Danach können alle Funktionen des Speichermonitors wie gewohnt benutzt werden, nur daß alle Befehle, die Speicherbereiche lesen, sich nun auf das Gerät x beziehen, während alle Schreib-

operationen auf das Gerät y laufen.

Wird y weggelassen, gilt y=x, also bezeichnet x das Gerät für Ein- und Ausgaben. Fehlt auch x, werden x und y gleich 0 gesetzt, das bedeutet, daß der Diskmonitor beendet wird.

Dazu einige Beispiele:

@\*8 (oder @\*88) – Lesen und Schreiben im Floppyspeicher.  
D 0400 – Disassemblieren. Änderungen werden in den Floppyspeicher geschrieben.

@\*80 – Lesen aus Floppy, Schreiben in Computer.

T C100 C200 C100 – Übertragen der ersten 256 Bytes des Floppy-ROMs in den Computerspeicher (ab \$C100).

I 0200 – Ausgabe des Fehlerspeichers. Auf Wunsch (Return) Übernahme in den Computerspeicher.

@\*89 – Lesen aus Floppy 8, Schreiben in Floppy 9.

M 0500 0580 – Speicherbereich ausgeben. Mit Return werden Daten übertragen (Selektion von Daten).

@\* (oder @\*0) – Lesen und Schreiben in Computer: Disk-Monitor beenden.

Weiterhin können natürlich alle Umrechnungsfunktionen und auch die Basic-Befehle aus dem Monitor heraus aufgerufen werden.

Die Funktion zur Umlenkung von Speicherein- und -ausgabe ermöglicht hundertprozentige Kompatibilität zwischen Speicher- und Disk-Monitor, wer sich also mit dem Speichermonitor schon auskennt, braucht keine neuen Befehle oder Syntax-Regeln für den Disk-Monitor zu lernen.

### Programmieren im Freezer

Wer im Freezer kurze Maschinenroutinen schreiben oder verändern wollte, hat es vielleicht schon gemerkt: Der Monitor verweigert die Annahme von vier Assemblerbefehlen. Es handelt sich hierbei um die Befehle CLC, CLD, CLI und CLV. Außerdem können im Disassemblerlisting keine Bytes mit \$CB oder \$CF überschrieben werden.

Das hat seinen Grund wohl darin, daß der Monitor bei Eingabe von C als erstem Buchstaben die Befehle CB (Clear Breakpoint) oder CF (Clear Freezepoint) erwartet. Abhilfe kann dadurch geschaffen werden, daß man statt der Mnemonics die Hex-Codes der Befehle einträgt:

CLC = 18

CLD = D8

CLI = 58

CLV = B8

Möchte man ein Byte auf \$CB oder \$CF ändern, kann man den Hex-Dump-Modus zu Hilfe nehmen (.M addr).

### Basic-Listing auf dem Drucker ausgeben

Basic-Programme zu drucken ist mit AR eine der einfachsten Aufgaben: Mit nur acht (!) Tastendrücker wird das Basic-Programm ausgedruckt.

1. Man geht mit <F8> in den Monitor.

2. Eingabe:

PBL<Shift+I> <Return>

Mit P wird der Drucker zugeschaltet, B leitet den folgenden Basic-Befehl L<Shift+I> = LIST ein, und schon rattert der Drucker los.

3. Der Monitor wird mit

X <Return> verlassen.

### Monitor an der Speichergrenze

Der Monitor des Moduls hat eine seltsame Eigenschaft: Während M xxxx normalerweise nur die acht Bytes nach xxxx listet, verhält sich der Monitor nach Adressen größer \$FFF7 (z.B. M FFF8) wie nach Anhängen eines Minuszeichens (M FFF8-), d.h. er listet weiter. Befinden sich auf dem Bildschirm wichtige Daten, sollte dies wegen eines möglichen Scrollings beachtet werden.

### Lores-Hardcopy

Sowohl vom Basic als auch aus dem Monitor läßt sich mit @P\* der aktuelle (sichtbare) Bildschirm in ASCII-Zeichen drucken. Zusätzlich kann man vom Freezer aus den eingefrorenen Bildschirm mit @P\*\* ausdrucken lassen. (pk)