

ATARI®

ATARI PROGRAMM

DXG 5724
Diskette

ATAMON

c 1983 Jegliche Rechte vorbehalten
ATARI ELEKTRONIK - Vertriebsges. mbH

INHALTSVERZEICHNIS

Kapitel	Seite
Inhaltsverzeichnis	1
Einleitung	2
Laden des Programms	2
Verwendete Abkürzungen	2
Erläuterung der Befehle:	3
1.) R - Befehl Register-Dump	3
2.) ; - Befehl Register-Dump ändern	3
3.) G - Befehl Starten eines Maschinenprogramms	3
4.) M - Befehl Memory-Dump	4
5.) : - Befehl Memory-Dump ändern	4
6.) T - Befehl Verschieben eines Bereichs	5
7.) V - Befehl Vergleichen eines Bereichs	5
8.) F - Befehl Auffüllen eines Bereichs	5
9.) C - Befehl Speicher ändern in binär	5
10.) S - Befehl dauerndes Speicher-Anzeigen in binär	5
11.) PE - Befehl Drucker einschalten	6
12.) PA - Befehl Drucker ausschalten	6
13.) PT - Befehl Text auf Drucker ausgeben	6
14.) E - Befehl Sektoren einlesen	6
15.) W - Befehl Sektoren rausschreiben	6
16.) % - Befehl Binär-Hex Umrechnung	7
17.) ! - Befehl Hex-Binär Umrechnung	7
18.) # - Befehl Dezimal-Hex Umrechnung	7
19.) \$ - Befehl Hex-Dezimal Umrechnung	7
20.) I - Befehl Text in Speicher einschreiben	7
21.) H - Befehl Suchen nach Zeichen	7
22.) H - Befehl Suchen nach Zeichen und Ersetzen	8
23.) *+ - Befehl Addition	8
24.) *- - Befehl Subtraktion	8
25.) *A - Befehl UND-Verknüpfung	8
26.) *O - Befehl ODER-Verknüpfung	8
27.) *E - Befehl EXOR-Verknüpfung	8
28.) D - Befehl Disassemblieren	8
29.) , - Befehl Disassembler Listing ändern	9
30.) X - Befehl Rücksprung ins DOS	9
31.) L - Befehl Testen eines Bereiches auf ein Zeichen	9
32.) Z - Befehl CIO-Aufruf	9
Load und Save eines Maschinenprogramms mit Diskette	10
Break und Reset	10
Atamon und Basic	11
Restaurieren gelöschter Files	11
Literatur	12
Kurzübersicht der Befehle	13
Index	15

EINLEITUNG

Das Programm ATAMON ist ein vielseitiger Monitor mit 32 Befehlen zum Analysieren und Austesten von Maschinenprogrammen.

Das Programm liegt im RAM von \$4000 bis \$5200. Beim ersten Start des Programmes wird die Systemvariable MEMLO automatisch so belegt, daß das Programm vor Basic oder Assembler geschützt ist.

LADEN DES Programms

- 1.) Schalten Sie den Rechner aus.
- 2.) Schalten Sie die Diskettenstation an.
- 3.) Legen Sie eine Diskette mit DOS II ein.
- 4.) Schalten Sie den Rechner ein. (Der Rechner lädt nun automatisch das DOS.)
- 5.) Legen Sie die Diskette mit dem Programm ATAMON ein.
- 6.) Laden Sie den Monitor mit Hilfe des "L"-Befehls (Näheres entnehmen Sie dem DOS II Manual). Das Programm wird damit automatisch gestartet. Sollten Sie vom ATAMON wieder ins DOS gesprungen sein und wollen den Monitor wieder starten, so drücken Sie "M" und RETURN. Dann geben Sie "4000" ein und drücken wiederum RETURN. Das Programm wird damit gestartet.

VERWENDETE ABKÜRZUNGEN

AAAA = Hexadresse von
BBBB = Hexadresse bis
PPPP = vier Digits lange Hexzahl, die beim Ausdruck zu AAAA hinzuaddiert wird.
CCCC = Hexadresse ab
YYYY = vier Digits lange Hexzahl
XX = Hexbyte
VV = zwei Digits lange Hexzahl
N = Dezimal im Bereich von 0-65535
IIIIIIII = acht Digits lange Binärzahl
SSSS = hexadezimale Sektornummer im Bereich von 0001-02D0
STRING = beliebige Zeichenkette (darf jedoch kein " " enthalten)
(R) = Drücken Sie RETURN-Taste

ICBA = vier Digits lange Hexadresse, die in die Pointer ICBAL und ICBAH geschrieben wird (Näheres siehe OS-User Manual von ATARI).
ICBL = vier Digits lange Hexadresse, die in der Pointer ICBLL und ICBLH geschrieben wird (Näheres siehe OS-User Manual von ATARI).

ERLÄUTERUNGEN DER BEFEHLE

1.) R (R)

Der "R" Befehl bringt einen Register-Dump der CPU auf den Schirm:

```
*R
PC SP AC XR YR NV/BDIZC
; 200F F0 90 33 4A 10100100
```

Der Buchstabe nach dem Asterix (Stern) gibt Ihnen an, wodurch der Registerdump erzeugt wurde. Dies kann durch folgende Ereignisse geschehen: Software Interrupt oder BRK-Instruktion (B), R-Befehl (R), Break-Taste (B invers) oder Kaltstart (C). PC gibt Ihnen den Stand des CPU-Programmmählers an. Er gibt immer die Speicherstelle an, die nach einem BRK diesem Befehl folgt. SP gibt den Wert des Stapelzeigers vor dem BRK an. AC, XR und YR geben die Inhalte der CPU-Register A, X und Y vor dem BRK an. Schließlich wird noch der Inhalt des Statusregisters vor dem BRK in binärer Schreibweise dargestellt. Sobald der ATAMON über einen BRK-Befehl angesprochen wird, werden diese Inhalte zwischengespeichert und bei einem eventuellen G-Befehl vor dessen Ausführung wiederhergestellt. Ein Drücken der Break-Taste erzeugt ebenfalls einen Register-Dump.

2.) ; AAAA XX XX XX XX IIIIIIII (R)

Wenn Sie bei dem vorher erklärten R-Befehl mit dem Cursor in die Zeile gehen, die die Registerinhalte anzeigt (diese fängt mit einem ";" an), so können Sie die Registerinhalte ändern. Diese werden nach einem RETURN zwischengespeichert.

3.) G AAAA (R) G (R)

Mit diesem Befehl können Sie ein Maschinenprogramm starten. Dabei werden vorher die Registerinhalte mit den zwischengespeicherten Werten gefüllt (siehe auch R- und ;-Befehl). AAAA gibt die Adresse des Maschinenprogramms

an. Wird AAAA nicht angegeben, so wird stattdessen der zwischengespeicherte Wert (PC bei Registerdump) genommen. So können Sie zum Beispiel mit "G 2000" einen Kaltstart des Monitors durchführen.

4.) M AAAA,BBBB,PPPP (R) M AAAA,BBBB (R) M AAAA (R)

Der M-Befehl bewirkt einen kombinierten Hex- und Ascii-Dump des Speichers. Dabei werden pro Zeile jeweils die Adresse im Speicher, acht Bytes in Hex- und dieselben acht Bytes in Ascii ausgedruckt:

```
:3000 30 30 31 32 33 34 35 36 00123456
```

Der Hexdump beginnt bei der Startadresse AAAA und endet mit der Endadresse BBBB. Wird die Endadresse BBBB nicht angegeben, so läuft der Hexdump solange weiter, bis er mit der BREAK-Taste unterbrochen wird.

Bei der Ausgabe der Hexadresse, bei der die Bytes stehen, wird der Wert von PPPP zu der augenblicklichen Adresse hinzuaddiert. Somit können Sie zum Beispiel einen Dump von einem Programm anfertigen, daß normalerweise an einer Stelle liegt, die bereits belegt ist. Sie laden dieses Programm dann einfach an eine Stelle, die frei ist und geben PPPP derart an, daß die ausgegebenen Adressen mit den ursprünglichen Adressen übereinstimmen.

5.) :AAAA XX XX XX XX XX XX XX XX (R)

Sie können auch beliebige Stellen im Speicher ändern, indem Sie mit dem Cursor im oben beschriebenen Hexdump an die entsprechende Stelle fahren und das gewünschte Byte ändern. Nach dem Drücken der RETURN-Taste wird der neue Wert in dem Speicher geschrieben. Sie sehen dies daran, daß sich hinten in der Ascii-Darstellung das entsprechende Byte ändert.

Danach wird die Adresse der nächsten acht Bytes ausgegeben und der Cursor springt an die Stelle des ersten Bytes, bereit für neue Änderungen.

Sie können zum Beispiel Tabellen eingeben, indem Sie ":" und die Adresse, ab der die Tabelle liegen soll, eingeben. Danach geben Sie einfach die ersten acht Bytes, jeweils durch ein Space (Leerzeichen) getrennt, ein und drücken RETURN. Sofort erscheint in der nächsten Zeile der :-Befehl und die Adresse der nächsten acht Bytes und sie können diese sofort eingeben. Sind Sie fertig, so drücken Sie einfach RETURN, ohne irgendwelche Bytes einzugeben oder drücken die BREAK-Taste.

Sollte das Ändern eines Bytes nicht möglich sein, weil Sie zum Beispiel versuchen ein Byte im ROM zu ändern, so erscheint hinter dem entsprechenden Byte ein Fragezeichen und das Programm hört auf, weitere Bytes zu ändern.

6.) T AAAA,BBBB,CCCC (R)

Mit diesem Befehl können Sie einen beliebigen Block im Speicher, der bei AAAA anfängt und bei BBBB endet, nach CCCC verschieben. Der Befehl erlaubt auch nach oben und unten überlappendes Verschieben, ohne daß das Programm dabei zerstört wird. Gleichzeitig wird ein Test durchgeführt, ob der Block an die neue Stelle geschrieben wird. So ist es zum Beispiel nicht möglich, einen Block in den Bereich eines ROMs hinein zu verschieben. In diesem Fall werden die entsprechenden Adressen ausgedruckt.

7.) V AAAA,BBBB,CCCC (R)

Dieser Befehl vergleicht einen Block von AAAA bis BBBB mit einem Block von CCCC und gibt alle Adressen des ersten Blocks aus, in dem dieser nicht mit dem zweiten übereinstimmt.

8.) F AAAA,BBBB,XX (R)

Damit können Sie einen beliebigen Speicherbereich AAAA bis BBBB mit dem Byte XX füllen. Die Adresse, in die das Byte nicht hineingeschrieben werden kann (z.Bsp. ROM oder fehlendes RAM), werden ausgedruckt. Damit können Sie auch auf einfache Weise Ihre RAMs prüfen, indem Sie verschiedene Hexmuster in den Speicher schreiben. Defekte Speicherstellen werden dann ausgegeben.

9.) C AAAA IIIIIIII,IIIIIII,...(R)

Sehr hilfreich für das Experimentieren mit Registern oder dem Game-Ports ist dieser Befehl. Er ermöglicht es, einzelne Speicherstellen binär zu ändern. Bis zu acht Bytes können Sie, beginnend bei AAAA, pro Befehl ändern.

10.) S AAAA (R)

Noch hilfreicher für Port-Experimente ist dieser Befehl, mit dem Sie vier Speicherstellen ab AAAA andauernd in binärer Darstellung anzeigen können. Diese Bytes werden solange angezeigt, bis Sie die BREAK-Taste drücken. Sie können somit zum Beispiel jede Pegeländerung an den Ports sofort auf dem Bildschirm ablesen.

11.) P E (R)

Dieser Befehl schaltet einen angeschlossenen Drucker an. Sämtliche Ausgaben auf dem Schirm werden dann automatisch auch auf dem Drucker ausgegeben. Es ist dabei egal, ob Sie einen Drucker über das Interfacemodul 850 oder die Gameports ansteuern. ATAMON benutzt den in der Handlertabelle eingetragenen Printer-Handler.

12.) P (R)
P A (R)

Damit schalten Sie den Drucker wieder aus.

13.) PT /XX,XX,'STRING',XX/
PT /'STRING',XX/
PT /XX,XX,XX,XX/
PT /'STRING'/'

Mit diesem Befehl können Sie einen Text auf den Drucker ausgeben. Der Text kann aus einem String, einer Reihe von Hexbytes (in Ascii) oder jeder Kombination von beiden bestehen.

14.) E SSSS (R)
E SSSS,AAAA (R)
E SSSS,AAAA,VV (R)

Dieser Befehl liest ab Sector SSSS VV Sektoren von Disk und speichert sie ab AAAA ab. Wird VV nicht angegeben, so wird nur ein Sector gelesen. Fehlt AAAA, so wird der Sector nach \$0680 abgespeichert. Dieser Befehl arbeitet nur mit Floppy Disk 810. Auftretende Fehler werden angezeigt, wobei zuerst die Sektornummer in hexadezimal und dann die Fehlernummer in dezimal ausgegeben werden. Die Bedeutung der Fehlermeldungen können Sie Ihrem Basic- oder Assembler-Manual entnehmen.

15.) W SSSS (R)
W SSSS,AAAA (R)
W SSSS,AAAA,VV (R)

Dieser Befehl schreibt ab Sector SSSS VV Sektoren von AAAA ab auf Disk. Wird VV nicht angegeben, so wird nur ein Sector geschrieben. Fehlt AAAA, so wird der Sector von \$0680 gelesen. Dieser Befehl arbeitet nur mit Floppy Disk 810. Auftretende Fehler werden angezeigt, wobei zuerst die Sektornummer in hexadezimal und dann die Fehlernummer in dezimal ausgegeben werden. Die Bedeutung der Fehlermeldungen können Sie Ihrem Basic- oder Assembler-Manual entnehmen.

16.) % I I I I I I I I (R)

Damit können Sie acht Digits lange Binärzahlen in Hexadezimal umrechnen.

17.) ! Y Y Y Y (R)

Vier Digits lange Hexzahlen können Sie mit diesem Befehl in sechzehn Digits lange Binärzahlen umrechnen.

18.) # N (R)

Dieser Befehl rechnet eine Dezimalzahl in Hex um. Die Dezimalzahl muß sich in den Grenzen von 0 und 65535 befinden.

19.) \$ Y Y Y Y (R)

Der \$-Befehl rechnet eine vier Digits lange Hexadezimalzahl in eine Dezimalzahl um.

20.) I A A A A , / X X , X X , X X / (R)
I A A A A , / ' S T R I N G ' / (R)
I A A A A , / X X , X X , ' S T R I N G ' / (R)
I A A A A , / ' S T R I N G ' , X X , X X , X X / (R)
I A A A A , / X X , X X , ' S T R I N G ' , X X / (R)

Mit diesem Befehl können Sie eine beliebige Zeichenkette ab A A A A abspeichern. Die Zeichenkette kann, wie oben dargestellt, entweder nur aus einem String, nur aus Hexbytes oder jeder Kombination von beiden bestehen. Jeder String muß dabei von zwei "'" eingeschlossen sein und von Hexbytes durch ein "," getrennt sein. Ebenso müssen Hexbytes untereinander durch ein "," getrennt sein. Die gesamte Zeichenkette wird durch einen beliebigen Begrenzer (im obigen Fall "/") eingeschlossen. Dabei kann der Rechte weggelassen werden.

21.) H A A A A , B B B B , / X X , X X , X X / (R)
H A A A A , B B B B , / ' S T R I N G ' / (R)
H A A A A , B B B B , / X X , X X , ' S T R I N G ' / (R)
H A A A A , B B B B , / ' S T R I N G ' , X X , X X , X X / (R)
H A A A A , B B B B , / X X , X X , ' S T R I N G ' , X X / (R)

Dieser Befehl ermöglicht es Ihnen, einen Block A A A A bis B B B B nach einer Zeichenkette und/oder einer Bytekombination oder einer beliebigen Kombination der beiden zu durchsuchen. Es werden die Adressen ausgegeben, bei denen das Gesuchte auftritt. Für die Syntax des Suchstrings gilt dasselbe wie bei dem Befehl "I".

22.) H A A A A , B B B B , / X X , X X , X X / X X , X X , X X / (R)
H A A A A , B B B B , / ' S T R I N G ' / X X , X X , X X / (R)
H A A A A , B B B B , / X X , X X , X X , ' S T R I N G ' / (R)
H A A A A , B B B B , / X X , X X , ' S T R I N G ' / X X , X X , X X , X X / (R)
H A A A A , B B B B , / X X , X X , X X , X X , X X , X X / ' S T R I N G ' / (R)
H A A A A , B B B B , / ' S T R I N G ' , X X , X X , X X / X X , ' S T R I N G ' , X X / (R)
H A A A A , B B B B , / X X , X X , ' S T R I N G ' , X X / ' S T R I N G ' , X X / (R)

Mit diesem Befehl können Sie einen Block A A A A bis B B B B nach einer Zeichenkette durchsuchen und durch eine andere Zeichenkette ersetzen. Die erste Zeichenkette im Befehl stellt den Suchstring dar. Die zweite diejenige, durch die der Suchstring ersetzt wird. Für beide Zeichenketten gelten die Syntaxregeln des "I"-Befehls. Beide Zeichenketten müssen gleich lang sein, andernfalls erscheint eine Fehlermeldung.

23.) *+ Y Y Y Y , Y Y Y Y (R)

Hiermit können Sie zwei Hexzahlen addieren. Außer dem Ergebnis wird auch noch ein eventueller Übertrag des Carry-Flags angezeigt.

24.) *- Y Y Y Y , Y Y Y Y (R)

Hiermit können Sie zwei Hexzahlen subtrahieren. Außer dem Ergebnis wird noch ein eventueller Übertrag des Carry-Flags angezeigt.

25.) *A Y Y Y Y , Y Y Y Y (R)

Mit diesem Befehl können Sie eine AND-Verknüpfung zweier Hexzahlen durchführen.

26.) *O Y Y Y Y , Y Y Y Y (R)

Mit diesem Befehl können Sie eine OR-Verknüpfung zweier Hexzahlen durchführen.

27.) *E Y Y Y Y , Y Y Y Y (R)

Mit diesem Befehl können Sie eine EXOR-Verknüpfung zweier Hexzahlen durchführen. Ein Sonderfall davon ist die Negation einer Hexzahl. Ist die eine Hexzahl F F F F , so wird die andere Hexzahl negiert.

28.) D A A A A (R)
D A A A A , B B B B (R)
D A A A A , B B B B , P P P P

Dieser Befehl disassembliert von A A A A bis B B B B ein

Maschinenprogramm. Zur Syntax der Parameter siehe auch 4.) : "M"-Befehl.

29.) ,AAAA XX XX XX (R)

Wenn Sie im Disassemblerlisting Änderungen vornehmen wollen, so fahren Sie mit dem Cursor an die entsprechende Stelle und ändern die Bytes nach der Adresse. Die Bytes werden dann vom Bildschirm geholt und in den Speicher geschrieben. Als nächstes wird in der selben Zeile ein (!) Befehl disassembliert und in der nächsten Zeile die auf den disassemblierten Befehl folgende Adresse. VORSICHT! Obwohl nur ein Befehl disassembliert wird, werden alle eingegebenen Bytes in den Speicher geschrieben.

Sie können somit auch ein Maschinenprogramm im Hexcode eingeben, indem Sie "," und die Startadresse eingeben und dann die Hexbytes des ersten Befehls eintippen. Nach dem RETURN wird der Befehl disassembliert und in der nächsten Zeile die nächste Adresse ausgegeben, so daß Sie nur noch die Bytes des nächsten Befehls eingeben brauchen.

Sie können den Eingabemodus wieder verlassen, indem Sie die BREAK-Taste drücken oder nach der Adresse keine Hexbytes eingeben.

30.) X (R)

Mit diesem Befehl können Sie den Monitor wieder verlassen und in das DOS zurückspringen.

31.) L AAAA,BBBB,XX

Mit diesem Befehl können Sie einen beliebigen Bereich AAAA bis BBBB auf ein Hexbyte XX überprüfen. Jede Speicherstelle, in der das Hexbyte XX nicht steht, wird ausgedruckt.

32.) Z XX,XX

Z XX,XX,ICBA,ICBL
Z XX,XX,ICBA,ICBL,A1
Z XX,XX,ICBA,ICBL,A1,A2
Z XX,XX,ICBA,ICBL,A1,A2,A3
Z XX,XX,ICBA,ICBL,A1,A2,A3,A4
Z XX,XX,ICBA,ICBL,A1,A2,A3,A4,A5
Z XX,XX,ICBA,ICBL,A1,A2,A3,A4,A5,A6

Mit diesem Befehl verfügen Sie über einen leistungsfähigen Zugriff auf die CIO. Sie haben die Möglichkeit, direkt vom Monitor jede mögliche CIO-Operation auszuführen.

Das erste Hexbyte wählt die gewünschte IOCB aus und muß eine durch 16 (\$10) teilbare Zahl zwischen 0 (\$00) und 112 (= \$70) sein. Das zweite Hexbyte ist das Command-Byte. Damit wählen Sie die gewünschte Operation an:

\$03	Open for Input/Output	\$05	Get Record (Text)
\$07	Get Character(s)	\$09	Put Record (Text)
\$0B	Put Character(s)	\$0C	Close Device
\$0D	Status Request	\$0E	Beginning of special Entry

Als nächstes kommen die Pufferadresse ICBA und die Pufferlänge ICBL. Bei Bedarf können danach noch die Auxiliary-Bytes 1-6 angegeben werden. Das Auxiliary-Byte 1 wird zum Beispiel beim OPEN-Befehl benötigt. Nähere Informationen über die Benutzung der CIO können Sie dem Os-User Manual (Nr. C016555) von ATARI entnehmen.

LOAD UND SAVE EINES MASCHINENPROGRAMMS MIT DISKETTE

Um ein Maschinenprogramm von Diskette zu laden oder auf Diskette abzuspeichern, verlassen Sie bitte den ATAMON mittels des X-Befehls. Bei DOS II ist zuvor die DOS-II Master-Diskette einzulegen. Danach verwenden Sie die entsprechenden DOS-Befehle zum Laden bzw. Abspeichern eines Programmes.

BREAK UND RESET

Im Monitor kommen Sie in den meisten Fällen durch Drücken der BREAK-Taste wieder in den Eingabe-Modus zurück, falls das Programm irgendwo "hängt". Sollte dies nicht wirken, so drücken Sie SYSTEM-RESET. Bei DOS-II ist vorher die DOS-II Master-Diskette einzulegen. Von DOS aus können Sie den Monitor wieder mit dem RUN-Befehl starten. Runadresse ist \$4000. Sollte der Monitor danach nicht mehr einwandfrei laufen, so ist er teilweise zerstört und muß neu geladen werden.

Sollte das Drücken der SYSTEM-RESET-Taste keine Wirkung zeigen, ist das System abgestürzt, so daß nur noch ein Ausschalten und anschließendes Neuladen des Monitors hilft.

ATAMON UND BASIC

Sie haben die Möglichkeit, mit ATAMON auch Basic-Programme und deren Struktur im Speicher zu analysieren. Dabei gehen Sie folgendermaßen vor:

- 1.) Schalten Sie den Rechner aus.
- 2.) Stecken Sie das Basic-Modul in den ROM-Schacht.
- 3.) Schalten Sie Ihre 810-Diskettenstation ein.
- 4.) Legen Sie die DOS-Masterdiskette ein, sobald das Busy-Licht verlöschen ist.
- 5.) Schalten Sie den Rechner ein.
- 6.) Laden Sie vom DOS aus ATAMON.
- 7.) Kehren Sie mittels des X-Befehls wieder in das DOS zurück.
- 8.) Springen Sie vom DOS ins Basic.
- 9.) Laden Sie das gewünschte Basic-Programm.
- 10.) Springen Sie mittels DOS wieder ins DOS zurück.
- 11.) Starten Sie vom DOS aus den ATAMON (Runadresse: \$4000).

Dadurch, daß Sie zuerst den ATAMON geladen und gestartet haben, wurde der Speicherbereich, in dem ATAMON liegt, vor dem Zugriff des Basics geschützt. Das Laden eines Basicprogramms kann nun den ATAMON nicht mehr zerstören. Nach dem Rücksprung zum DOS und dem erneuten Start des Monitors steht das Basicprogramm zum Analysieren zur Verfügung. Die Startadresse des Basicprogramms steht in den folgenden Speicherstellen:

\$80,81	Token Output Buffer	\$82,83	Variable Name Table
\$84,85	Variable Name Table End	\$86,87	Variable Value Table
\$88,89	Statement Table	\$8A,8B	Current Statement
\$8C,8D	String/Array area	\$8E,8F	Runtime Stack

Wollen Sie ein mit dem Monitor geändertes Programm wieder abspeichern, so verlassen Sie den Monitor mit Hilfe des X-Befehls und springen vom DOS wieder ins BASIC.

Weitere Informationen (insbesondere über die Speicherstruktur von Basic-Programmen) können Sie dem DE RE ATARI von ATARI entnehmen.

RESTAURIEREN GELÖSCHTER FILES

Sollten Sie per Zufall einmal ein File auf Disk gelöscht haben, so können Sie dieses wieder ohne Probleme restaurieren, solange Sie nicht auf diese Disk geschrieben haben.

Die Directory belegt die Sektoren \$0168 bis \$0170. Jeder Eintrag umfaßt 16 Bytes und hat folgendes Format:

AA LL LH SL SH XX XX XX XX XX XX XX XX

Dabei haben die Buchstaben folgende Bedeutung:

AA = Zustandsindicator: \$00 = kein Eintrag, \$42 = unlocked File, \$62 = Locked File, \$80 = deleted File

LL = Anzahl der Sektoren des Files low Byte

LH = Anzahl der Sektoren des Files high Byte

SL = erster Sector des Files low Byte

SH = erster Sector des Files high Byte

XX = Filenamen in Ascii

Wenn Sie nun durch Zufall ein File gelöscht haben, so ist das erste Byte des Directoryeintrags (AA) auf \$80 gesetzt. Alles, was Sie nun zu tun haben, ist, dieses Byte wieder auf \$42 zu setzen. Dazu laden Sie die acht Sektoren der Directory mit dem "E"-Befehl in den Speicher und suchen nach dem Eintrag des gelöschten Files. Dort ändern Sie das erste Byte und speichern die geänderten Directory mit dem "W"-Befehl wieder auf Disk. Was nun kommt, ist wichtig! Bevor Sie auf diese Diskette wieder schreiben, laden Sie das restaurierte File wieder in den Speicher (von Basic, Assembler etc.) und speichern es dann wieder auf Diskette ab. Nun ist das File vollständig wieder hergestellt. Das anschließende Laden und wieder Abspeichern ist deshalb notwendig, weil beim Löschen des Files die Sektoren, die es belegt, in der Volumetable (Sektor \$0168) als frei markiert werden. Bei einem anschließenden Schreiben auf Diskette werden diese Sektoren dann durch das neue File belegt und das gelöschte File damit endgültig zerstört.

LITERATUR

Operating System Source Listing
Operating System User Manual
DOS-II Source Listing
DE RE ATARI
DOS-II Manual
BASIC-Manual
ASSEMBLER EDITOR Manual

KURZÜBERSICHT DER BEFEHLE

1.) R	= Registerdump
2.) ; AAAA XX XX XX XX IIIIIIII	= Register ändern
3.) G AAAA	= Startet Maschinenprogramm ab Adresse AAAA
G	= Startet Maschinenprogramm ab Adresse PC
4.) M AAAA	= Memory Dump ab AAAA
M AAAA,BBBB	= Memorydump AAAA bis BBBB
M AAAA,BBBB,PPPP	= Memorydump AAAA bis BBBB, zu den ausgegebenen Adressen wird PPPP addiert.
5.) :AAAA XX XX XX XX XX XX XX XX	= Speicherinhalte ändern
6.) T AAAA,BBBB,CCCC	= Verschiebt Block AAAA bis BBBB nach CCCC.
7.) V AAAA,BBBB,CCCC	= Vergleicht Block AAAA bis BBBB mit Block ab CCCC.
8.) F AAAA,BBBB,XX	= Füllt Speicher von AAAA bis BBBB mit XX.
9.) C AAAA IIIIIIII,IIIIIIII...	= Ändern von Speicherinhalten in binär.
10.) S AAAA	= Dauernde Anzeige von vier Speicherstellen in binär
11.) P E	= Einschalten des Druckers
12.) P	= Ausschalten des Druckers
P A	= Ausschalten des Druckers
13.) PT /XX,XX,XX/	= Ausgabe der Zeichen in Ascii auf dem Drucker
PT /'STRING'/	= Ausgabe des Strings auf dem Drucker
PT /XX,'STRING',XX/	= Ausgabe von kombinierten Strings und Bytes
14.) E SSSS	= Liest Sector SSSS von Disk nach \$0680
E SSSS,AAAA	= Liest Sector SSSS von Disk nach AAAA
E SSSS,AAAA,VV	= Liest ab Sector SSSS VV Sektoren von Disk nach AAAA
15.) W SSSS	= Schreibt Sector SSSS auf Disk von \$0680
W SSSS,AAAA	= Schreibt Sector SSSS auf Disk von AAAA
W SSSS,AAAA,VV	= Schreibt ab Sector SSSS VV Sektoren auf D. von AAAA
16.) % IIIIIIII	= Binär-Hexadezimalwandlung

17.) ! YYYY	= Hexadezimal-Binärwandlung
18.) # N	= Dezimal-Hexadezimalwandlung
19.) \$ YYYY	= Hexadezimal-Dezimalwandlung
20.) I AAAA,/XX,'STRING',XX/	= Speichert STRING und/oder Bytes nach AAAA
21.) H AAAA,BBBB,/XX,XX,'STRING',XX/	= Sucht von AAAA bis BBBB nach STRING und/oder Bytes
22.) H AAAA,BBBB,XX,XX,XX/XX,XX,XX/	= Sucht und ersetzt von AAAA bis BBBB STRING, Bytes
23.) *+ YYYY,YYYY	= Addiert zwei Hexzahlen
24.) *- YYYY,YYYY	= Subtrahiert zwei Hexzahlen
25.) *A YYYY,YYYY	= AND-Verknüpfung zweier Hexzahlen
26.) *O YYYY,YYYY	= OR-Verknüpfung zweier Hexzahlen
27.) *E YYYY,YYYY	= EXOR-Verknüpfung zweier Hexzahlen
28.) D AAAA	= Disassembliert ab AAAA
D AAAA,BBBB	= Disassembliert AAAA bis BBBB
D AAAA,BBBB,PPPP	= Disassembliert AAAA bis BBBB, addiert PPPP zu den ausgedruckten Adressen
29.) ,AAAA XX XX XX	= Ändern des Disassemblerlistings
30.) X	= Rücksprung zu DOS
31.) L AAAA,BBBB,ZZ	= Test, ob Bereich AAAA - BBBB mit ZZ gefüllt ist.
32.) Z XX,XX,AAAA,BBBB,XX,XX,XX,XX,XX,XX	= Entspricht XIO aus BASIC
Z XX,XX	= IOCB#,ICCOM
Z XX,XX,AAAA,BBBB	= IOCB#,ICCOM,ICBA,ICBL
Z XX,XX,AAAA,BBBB,XX	= IOCB#,ICCOM,ICBA,ICBL,ICAU1
Z XX,XX,AAAA,BBBB,XX,XX	= IOCB#,ICCOM,ICBA,ICBL,ICAU1, ... ICAUX2
Z XX,XX,AAAA,BBBB,XX,XX,XX	= IOCB#,ICCOM,ICBA,ICBL,ICAU1, ... ICAUX3
Z XX,XX,AAAA,BBBB,XX,XX,XX,XX	= IOCB#,ICCOM,ICBA,ICBL,ICAU1, ... ICAUX4
Z XX,XX,AAAA,BBBB,XX,XX,XX,XX,XX	= IOCB#,ICCOM,ICBA,ICBL,ICAU1, ... ICAUX5

INDEX

Addieren	8
Ascii-Dump	4
ATAMON und Basic	11
Auxiliary-Byte	10
Basic	11
Basic Pointer	11
Bereich auf ein Byte überprüfen	9
Binär-Hex-Umrechnung	7
Block durchsuchen	7
Block durchsuchen und ändern	8
Block vergleichen	5
Block verschieben	5
Break	10
Break und Reset	10
BRK	3
C-Befehl	5
CIO	9
Command-Byte	10
CPA	2
CPU-programmzähler	3
D-Befehl	8
Dez-Hex-Umrechnung	7
Directory	12
Disassemblerlisting ändern	9
Disassemblieren	8
DOS-II	2
Drucker an	6
Drucker aus	6
E-Befehl	6
Einleitung	2
Erläuterung der Befehle	3
EXOR-Verknüpfung	8
F-Befehl	5
Find	7
G-Befehl	3
Gelöschte Files	11
H-Befehl (Search)	7
H-Befehl (Replace)	8
Hex-Binär-Umrechnung	7
Hex-Dez-Umrechnung	7
Hex-Dump	4
Hex-Dump ändern	4

I-Befehl	7
ICBA	9
ICBL	9
Index	15
Inhaltsverzeichnis	1
K-DOS	2
Kurzübersicht der Befehle	13
Laden des Programmes	2
L-Befehl	9
Literatur	12
Load und Save eines Maschinenprogramms mit Diskette	10
Maschinenprogramm starten	3
M-Befehl	4
Monitor verlassen	9
OR-Verknüpfung	8
PA-Befehl	6
PE-Befehl	6
PT-Befehl	6
R-Befehl	3
Registerdump	3
Registerinhalte ändern	3
Replace	8
Reset	10
Restaurieren gelöschter Files	11
Run-Adresse	10
S-Befehl	5
Sektoren lesen	6
Sektoren schreiben	6
Software-Interrupt	3
Speicherbereich füllen	5
Speicherbereich binär ändern	5
Speicherstellen dauernd binär anzeigen	5
Subtrahieren	8
System Reset	10
Tabellen eingeben	4
T-Befehl	5
Text auf Drucker ausgeben	6
UND-Verknüpfung	8
V-Befehl	5
Verwendete Abkürzungen	2
W-Befehl	6
X-Befehl	9

Z-Befehl	9
Zeichenkette abspeichern	7
;-Befehl	3
:-Befehl	4
, -Befehl	9
!-Befehl	7
#-Befehl	7
\$-Befehl	7
%-Befehl	7
*+-Befehl	8
*--Befehl	8
*A-Befehl	8
*O-Befehl	8
*E-Befehl	8

Wichtige Informationen

Lieber Computerfreund, lieber Kunde, lieber Händler!

Jeder, der sich einmal selbst damit beschäftigt hat, ein Computerprogramm zu fertigen, weiß, welche Arbeit und geistige Mühe aufgewendet werden muß, um eine Problemlösung zu finden und sie anwenderfreundlich zu programmieren. Die Erfüllung dieser Voraussetzungen erfordert viel Erfahrung und hohe finanzielle und zeitliche Investitionen. Das Ergebnis sind gute und erfolgreiche Computerprogramme, die von interessierten Anwendern nachgefragt werden und deshalb für den Händler verkäuflich sind.

Diese Tatsache machen sich einige dadurch zunutze, daß sie die mit hohen Voraufwendungen geschaffenen erfolgreichen Programme der Firma Atari kopieren oder ihren Kunden die Möglichkeit anbieten, die gewünschten Programme auf Diskette zu überspielen. Sie meinen, damit ihren Kunden ein gutes und billiges Angebot zu machen. Die Kunden wissen jedoch meist nicht, daß sie lediglich ein vermeintlich gutes und billiges Angebot erhalten.

Abgesehen davon, daß das Angebot zur Überspielung von Programmen und das Anbieten und Verkaufen illegal kopierter Programme strafrechtlich verboten ist, weil es sich dabei um Verletzungen des Urheberrechtes (COMPUTERPROGRAMM PIRATERIE) handelt, die von Atari gegenüber jedermann ohne Ansehen der Person gerichtlich verfolgt wird, so ist auch die Annahme falsch, das Angebot sei günstig oder billig:

- Gestohlene Ware ist immer billig. Der Dieb hat keine Voraufwendungen. Er eignet sich nur fremdes Eigentum an, für die der Käufer keine Gewährleistung erhält.
- Der Händler, der das Kopieren von Programmen anbietet, anstatt Originale zu verkaufen schmarotzt an fremder Leistung.
- Der interessierte Kunde wird bald keine guten Programme mehr kaufen können und illegale Programme wird der Handel bald auch nicht mehr anbieten können.

Letzteres deswegen, weil niemand mehr bereit und in der Lage sein wird, gute verkaufsfähige Programme zu entwickeln, wenn nicht die Möglichkeit besteht, die hohen Voraufwendungen durch Verkäufe wieder zu verdienen. Die Piraten sind geistig weder in der Lage noch überhaupt bereit, sich der Mühe zu unterziehen, Programme zu entwickeln. Sie können und wollen nur durch Diebstahl fremder guter Leistung eine schnelle bequeme Mark verdienen.

Wer also Interesse daran hat, daß das Angebot an guten Computerprogrammen wächst, sollte die illegalen „billigen“ Angebote meiden und mit dazu beitragen, daß den Totengräbern der Computer-Programmentwicklung und damit des Computerhandels das Handwerk gelegt wird.

Wir danken für Ihr Verständnis und freuen uns über jeden Hinweis von Ihnen.

Atari Elektronikvertriebsges. mbH

ATARI®

ATARI®



A Warner Communications Company

ATARI-Elektronik Vertriebsgesellschaft mbH
Postfach 60 01 69 · Bebelallee 10 · 2000 Hamburg 60

Jegliche Rechte vorbehalten.
Vermietung, Verleih, Vervielfältigung
und öffentliche Aufführung verboten.

ATARI PROGRAMM

DXG 5724
Diskette

ATAMON

© 1983 Jegliche Rechte vorbehalten
ATARI ELEKTRONIK - Vertriebsges. mbH