

16K Bibomon Handbuch

(c)1986 Compy-Shop
Version 1.6 vom 29.6.1986
Programmiert von E.Reuss

Handbuch von E.Reuss, P.Bee und F.Guth

Copyright Notiz:

ATARI, ATARI 1050, ATARI 800XL, ATARI 130XE und ATARI 850 sind eingetragene Warenzeichen
der Firma ATARI CORP. DEUTSCHLAND.

MAC/65, ACTION!, BASIC XL, BASIC XE und DOS XL sind eingetragene Warenzeichen der Firma
OPTIMIZED SYSTEM SOFTWARE INC., USA.

Die Informationen im vorliegendem Handbuch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Die Autoren und Herausgeber dieses Handbuches können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind die Autoren dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien sowie der Übersetzung in fremde Sprachen. Die gewerbliche Nutzung der in diesem Handbuch gezeigten Modelle, Programme und Arbeiten ist nur mit der ausdrücklichen Genehmigung der Autoren erlaubt.

16K BIBOMON HANDBUCH
(c) 1986 Compy-Shop Ohg, 4330 Mülheim Ruhr
Alle Rechte vorbehalten

Druck: DER DRUCKER, BOCHUM
Printed in Germany

Inhaltsverzeichnis

0.0 Vorwort	1
1.0 Einführung	2
1.1 Der Hardware Aufbau	4
1.2 Hinweise zum Handbuch	5
2.0 Der 16K Bibomon	6
2.1 Der Monitor Befehlssatz	8
2.2 Die Bibomon Befehle	9
3.0 Das DOS des Bibomon	37
3.1 Der Befehlssatz des DOS	39
3.2 Die Befehle des DOS	40
3.3 Die Anwendung des DOS	50
4.0 Die Hardware Uhr	52
4.1 Einstellen der Uhr	54
5.0 Die Hardcopy Routine	55
5.1 Einstellen der Hardcopy Routine	56
5.2 Einstellen der Hardcopy Routine durch Programm	58
6.0 Der Centronics Port	59
6.1 Programmierung des Ports	60
7.0 Anhang	61
7.1 Clock Demo	62
7.2 Port Demo	63
7.3 Hardcopy einstellen in BASIC	64
7.4 Der Centronics Druckerstecker	65
7.5 Der Centronicsport	66
7.6 Port Demo Hardware	67
8.0 Index	68
9.0 Referenzkarte	69

Vorwort

Die meisten Vorworte beginnen mit einem Glückwunsch an Sie, weil Sie soeben das beste Zubehör für Ihren ATARI Computer gekauft haben.

Dieses nicht. Ob Sie mit dem 16K Bibomon das beste Zubehör gekauft haben, müssen Sie selbst Entscheiden. Mit der Zeit werden Sie sich allerdings fragen, wie Sie jemals ohne diesen Monitor gearbeitet haben.

Bevor wir Ihnen nun alle Funktionen des 16K Bibomon erklären, noch ein paar warme Worte vorweg.

Wir haben ca. 6 Monate an diesem Monitor gearbeitet und glauben, daß wir dem ernsthaften Programmierer mit diesem Monitor ein gutes Werkzeug für die tägliche Arbeit mit dem Computer gegeben haben. Leider müssen wir aber auch noch folgendes bemerken:

Der 16K Bibomon ist *nicht* dazu gedacht, Programme zu *knacken*, obwohl das auch mit dem Monitor geht. Wir haben den Monitor geschrieben, weil wir uns ständig über die unzureichenden Monitore der angebotenen Programmiersprachen geärgert haben. Auch die Uhr, oder die Harcopy Möglichkeit fehlte uns sehr oft.

Nun aber genug der Vorworte. Wir wünschen Ihnen viel Spaß und viel Erfolg bei der Arbeit mit dem 16K Bibomon.

Ihr Compy-Shop Team

EINFÜHRUNG

1.0 Überblick

Der 16K Bibomon ist ein Maschinensprachmonitor, der fest in Ihrem ATARI Computer eingebaut ist. Dieser Monitor gibt Ihnen jederzeit die Möglichkeit, ein laufendes Programm zu unterbrechen, und Veränderungen an diesem Programm vorzunehmen. Zu diesem Zweck steht Ihnen ein Zeilenassembler und ein Monitor zur Verfügung. Sie haben also ständig die Kontrolle über Ihr Programm.

Sie erreichen diesen Monitor durch vier verschiedene Wege. Der erste führt durch den Druck auf die Tasten *SELECT* und *SYSTEM RESET* in den Monitor. Bei diesem Monitor-einsprung werden alle System Adressen gerettet und zwischengespeichert. Sie haben dann später die Möglichkeit diese Adressen durch verschiedene Befehle abzufragen.

Der zweite Weg führt Sie über den Druck auf die Tasten *OPTION* und *SYSTEM RESET* in den Monitor. Bei diesem Monitor Einsprung werden alle System Adressen unterbrochen, aber nicht zwischengespeichert.

Der dritte Weg ist nur für den Notfall gedacht. Sollte Ihr Computer einmal **abgestürzt** sein, der Fall also, daß nur noch das Ausschalten und wieder Einschalten des Computers hilft, wobei Sie allerdings alle Daten im Arbeitsspeicher verlieren, so können Sie nun die Taste *SELECT* und den *HARDWARE RESET* Taster drücken. So gelangen Sie auch in den Monitor. Von hier aus können Sie nun Ihre Daten retten oder normal weiterarbeiten.

Der vierte Weg ist das Starten des Monitors durch einen direkten **JUMP** oder ein **USR** Aufruf. Für den **USR** Aufruf vom **ATARI Basic** gilt die Startadresse **49152 (\$C000)**, für den **JMP** von einem Assembler Programm, oder einem **DOS**, die Startadresse **\$C001 (dez. 49153)**. Auch bei diesem Einsprung werden die System Adressen zwar unterbrochen, aber nicht zwischengespeichert.

Eine weitere sehr nützliche Funktion des 16K Bibomon ist die nun in Ihrem Rechner eingebaute Quartzuhr. Alle Daten der Uhr können Sie abfragen und in eigene Programme einbauen. Viele neue Programmiermöglichkeiten werden sich Ihnen durch diese Möglichkeit eröffnen.

Mit Hilfe des 16K Bibomon können Sie aber auch jederzeit eine Hardcopy (einen Ausdruck) des sichtbaren Bildschirms erstellen. Auch diese Funktion des 16K Bibomon können Sie in eigene Programme einbinden und nutzen.

Nicht vergessen wollen wir auch den eigenbauten Druckerport. Alle Druckerausgaben des 16K Bibomon laufen über diesen Port. Auch hier gilt wieder: Sie können den Port für eigene Zwecke nutzen, da Sie ihn frei programmieren können.

Der 16K Bibomon ist ein starkes Werkzeug für den Maschinensprach Programmierer. Sie werden nach einer kurzen Einarbeitungszeit feststellen, daß das Arbeiten mit diesem Monitor leichter und effektiver ist als mit jedem anderen Monitor. Auch die programmierbare (Centronics-) Schnittstelle wird Ihnen nach kurzer Zeit zu einer Reihe neuer Ideen verhelfen. Die Anwendungsmöglichkeiten des 16K Bibomon und der Schnittstelle sind fast unbegrenzt.

1.1 Der Hardware Aufbau

Um mit dem 16K Bibomon arbeiten zu können, brauchen Sie die folgenden Geräte:

COMPUTER

Der 16K Bibomon kann in alle **ATARI** Computer der Serien **800XL** und **130XE** eingebaut werden. Eine Begrenzung der **RAM** Grösse gibt es nicht. Der 16K Bibomon kann mit allen **Compy-Shop Speichererweiterungen** bis zu einer Grösse von **320K** Ram arbeiten.

LAUFWERKE

Der 16K Bibomon arbeitet mit allen **ATARI** Laufwerken der Serien **810** und **1050**, und kompatiblen Laufwerken (**RANA 1000**, **TRAK 2D/4D**, **INDUS GT**) zusammen. Es sind sogar einige Spezialbefehle für **ATARI 1050** Laufwerke, die mit einer **SPEEDY 1050** modifiziert wurden, im Monitor enthalten. Sie können Daten in **SINGLE (88k)** und in **DOUBLE (176k)** Density lesen und schreiben.

DRUCKER

Bei den Druckern gibt es einige Einschränkungen. Der 16K Bibomon arbeitet mit einem Drucker nur über den eingebauten Centronics Parallel Port. Das bedeutet, daß Sie keinen Drucker benutzen können, der über die Serielle ATARI Schnittstelle betrieben wird. Zusätzlich gilt folgendes:

Die Ausgabe der ASCII Zeichen funktioniert auf allen Druckern. Die integrierten Hardcopy Routinen des 16K Bibomon lassen sich auf **EPSON FX80** und **OKIDATA MICROLINE 182/183** und **192/193 (CENTRONICS-PARALLEL)** Druckern einstellen.

1.2 Hinweise zum Handbuch

Wie der 16K Bibomon ist auch dieses Handbuch in vier Teile unterteilt. Im ersten Teil erklären wir Ihnen den Befehlssatz des Monitors. Im zweiten Teil finden Sie dann eine genaue Erklärung des DOS und des Befehlssatzes des DOS. Im dritten Teil finden Sie eine Erklärung der Uhr und schliesslich im vierten Teil eine genaue Beschreibung des Port's. Im Anhang dieses Handbuches geben wir Ihnen dann noch ein paar Hinweise zur Benutzung des 16K Bibomon, geben Ihnen eine Beschreibung einer Centronics - Schnittstelle und des Port's. Auch einige Beispielprogramme werden Sie im Anhang finden.

Viele Teile des 16K Bibomon arbeiten sehr eng zusammen. Wir verweisen daher im Handbuch des öfteren auf andere Teile des Handbuches. Zum Beispiel greifen viele Teile des Monitors auch auf das DOS zurück. Darum lässt es sich nicht vermeiden, daß wir bei der Beschreibung einiger Monitor Befehle auf die Beschreibung der DOS Befehle hinweisen.

Wir haben uns bemüht, dieses Handbuch so verständlich wie möglich zu schreiben. Sollten Sie trotzdem Schwierigkeiten haben, oder Fragen, so melden Sie sich bitte bei uns. Soweit es uns möglich ist, werden wir Ihnen helfen.

Sollten Sie **Anregungen** oder **Verbesserungsvorschläge** haben, so schreiben Sie uns bitte. Falls es möglich ist, werden wir Ihren Vorschlag in einer späteren Update Version berücksichtigen.

2.0 Der 16K Bibomon

Bevor wir Ihnen nun den Befehlssatz des Monitor's erklären, hier noch einmal einen Gesamtüberblick mit den Möglichkeiten des 16K Bibomon.

1. Der Monitor

Mit diesem Monitor haben Sie die folgenden Möglichkeiten:

- => *Sektoren lesen, schreiben, editieren*
 - => *Speicherstellen verändern durch einfaches überschreiben*
 - => *Speicherstellen verschieben oder vergleichen*
 - => *kurze Assemblerprogramme direkt eingeben*
 - => *ASCII oder Bytefolgen suchen*
 - => *Ausgabe der Daten in Bildschirmcode oder ASCII Code*
 - => *Programm Trace oder Singlestep*
 - => *direkte System Kontrolle*
- und vieles mehr.

2. Das DOS

Mit dem eingebautem DOS des Bibomon können Sie:

- => *Single Density formatieren, lesen, schreiben*
 - => *Double Density formatieren, lesen, schreiben*
 - => *Files Tracen, mit oder ohne laden*
 - => *Speicherbereiche als .COM File abspeichern*
 - => *Speicherbereiche als .BIN File abspeichern*
 - => *Binärfiles in beliebige Speicherstellen einlesen*
 - => *Sie brauchen kein DOS mehr zu booten*
 - => *Sie können das DOS jederzeit zu ihrem Programm*
 - => *oder Steckmodul zuschalten*
 - => *Textfiles direkt einlesen und anzeigen*
- und vieles mehr.

3. Die Uhr

Die Uhr des 16K Bibomon ist eine Akku gepufferte Quartzuhr. Das bedeutet, daß Sie diese Uhr nur einmal einzustellen brauchen. Die Daten dieser Uhr können Sie in eigene Programme einbauen.

4. Der Port

Der Port des 16K Bibomon ist programmierbar, das bedeutet, daß Sie mit Hilfe dieses Port's Daten sehr schnell an Aussengeräte übertragen können. Im Normalfall wird dieser Port als Centronics Parallel Port zum Anschluß eines Druckers benutzt. Aber als Programmierer stehen Ihnen alle Möglichkeiten offen.

5. Die Hardcopy Routine

Mit dieser Hardcopy Routine können Sie jederzeit von fast jedem sichtbaren Bildschirm einen Ausdruck machen. Einschränkungen müssen bei Programmen gemacht werden, die die Adresse Ihrer Displaylist nicht nach **\$230, \$231** ablegen. Diese Bildschirme kann der 16K Bibomon nicht finden und somit auch nicht ausdrucken.

Soweit der Überblick über die einzelnen Funktionen des 16K Bibomon. Bevor wir aber nun mit der Besprechung der Befehle des Monitors beginnen, noch einen Hinweis! Bei allen Speicherlisten, die Sie auf dem Bildschirm sehen, handelt es sich um normale ASCII Zeichen. Sie haben aber auch die Möglichkeit, sich statt der ASCII Zeichen den entsprechenden Bildschirmcode ausgeben zu lassen. Dazu drücken Sie, wenn die Bildschirmausgabe läuft, die **OPTION** Taste. Sie erhalten dann statt der ASCII Zeichen den Bildschirmcode.

2.1 Der Monitor Befehlssatz

Befehl - Bedeutung

=====

- (G) - Starte Programm (GOTO ADDRESS)
- (H) - Speedy 1050 Steuerung
- (I) - Disketten Index
- (J) - Springe zur Adresse (JUMP ADDRESS)
- (K) - SIO Kurzbefehl
- (L) - Disassembliere Speicher
- (M) - Verschiebe Speicher (MOVE)
- (N) - Ändere Spapelinhalte
- (O) - Öffne Bildschirm
- (P) - Druckersteuerbefehl
- (Q) - Ende, verlasse Monitor
- (R) - Lese Sektor(en) (READ)
- (S) - Einzelschritt (SINGLESTEP)
- (T) - Verfolge Programm (TRACE)
- (U) - Uhr ein/aus
- (V) - Vergleiche Speicher (VERIFY)
- (W) - Schreibe Sektor(en) (WRITE)
- (X) - Exklusiv Oder (EXOR)
- (Y) - Aktiviere Assembler
- (Z) - Verfolge File (TRACE FILE)
- (:) - Schreibe in Adresse (STORE)
- (;) - Schreibe in Adresse (STORE)
- (.) - Liste Speicher von..bis (DUMP)
- (+) - Rechne Hexzahl + Hexzahl
- (-) - Rechne Hexzahl - Hexzahl
- (&) - OR
- (^) - AND
- (,) - Liste halbe Seite (DUMP 1/2 PAGE)
- (/) - Letzte Disketten Operation
- (=) - Fülle Speicher
- (\$) - Wandelt Hex in Dez
- (#) - Wandelt Dez in Hex
- (') - Bildschirmcode
- (") - ASCII Code
- (>) - Suche nach (SEARCH)
- (!) - System Adressen
- (*) - DOS Kommando
- (?) - Registeranzeige
- (@) - Hauptmenu
- (\) - Bankumschalter
- (f) - Hardcopy Setup Menu

2.2 Die Bibomon Befehle

Hier nun die Monitor Befehle und Ihre Anwendung.

Befehl: G - Gehe zur Adresse (GOTO ADDRESS)

Format: (Adresse)(G)

Beispiel: 9000G

Erklärung: Dieser Befehl dient zum Testen von Unterroutinen. Die Kontrolle über das Programm bleibt beim Monitor. Der Monitor meldet sich nach dem Auftreten eines BRK oder eines RTS wieder zurück.

Befehl: H - Laufwerkssteuerrung (*SPEEDY 1050*)

Format: (H)

Beispiel: H

Erklärung: Dieser Bibomon Befehl kann nicht von allen Bibomon Benutzern angewendet werden. Dieser Befehl dient der Steuerung der *SPEEDY 1050*. Über diesen Befehl gelangen Sie in ein kleines Untermenü, mit dem Sie die *SPEEDY 1050* langsam stellen, oder ein RESET des Laufwerkes durchführen können. Die Besitzer einer *SPEEDY 1050* werden diese Funktion als sehr hilfreich empfinden.

Befehl: I - Disketten Index

Format: (I)

Beispiel: I

Erklärung: Entspricht dem DOS Befehl (*I). Entgegen einem Disketten Inhaltsverzeichnis erhalten Sie beim Disketten Index auch den genauen Status der auf der Diskette befindlichen Files und die Startsektoren. Das ist für einige Bibomon Befehle sehr wichtig.

Befehl: J - Springe zur Adresse (JUMP ADDRESS)

Format: (Adresse)(J)

Beispiel: 9000J

Erklärung: Mit diesem Befehl geben Sie die Kontrolle an das Programm ab, dessen Startadresse sich an der angegebenen Speicherstelle befindet. Sie verlassen mit diesem Befehl den Monitor. Unterschied zum GOTO Befehl: Bei einem RTS wird ein Kaltstart des Monitors ausgeführt. Ein BRK bewirkt ein Systemabsturz des Rechners.

Befehl: K - SIO Kurzbefehl

Format: K

Beispiel: K

Erklärung: Dieser Befehl entspricht der Anweisung E459G, also der Einsprung in die SIO-Routine die bei allen Disketten Befehlen benutzt wird. Mit diesem Befehl wird allerdings auch der Status in der Speicherstelle \$0303 richtig gesetzt, so daß der gleiche Befehl auch mehrfach hintereinander ausgeführt werden kann.

Befehl: L - Disassembliere Speicher

Format: (Adresse)(L)

Beispiel: 9000L

oder: L

oder: 9000LLLL

Erklärung: Zum Disassemblieren bestimmter Speicherbereiche benutzen Sie den Befehl L. Beim Disassemblieren listet der Bibomon immer nur soviel Speicherstellen auf einmal, wie auf dem Bildschirm Platz haben. In unserem ersten Beispiel erhalten Sie 22 Bildschirmzeilen disassembliert. Wollen Sie nun die nächsten 22 Zeilen sehen, brauchen Sie nur ein L einzugeben. Da aber auch Mehrfacheingaben in einer Zeile möglich sind, können Sie hinter der Startadresse auch gleich mehrere L Befehle eingeben. Wie Sie in unserem letzten Beispiel sehen können. Sie erhalten bei diesem Beispiel 4 mal 22 Bildschirmzeilen ohne Unterbrechung ausgegeben. Dieses Listing können Sie, wie alle Listen die auf dem Bildschirm ausgegeben werden, durch einen einfachen Druck auf die Leertaste anhalten. Wenn Sie nun auf die **RETURN** Taste drücken, erhalten Sie die nächste Zeile des Listings. Drücken Sie nun noch einmal auf die Leertaste, so läuft das Listing normal weiter. **ACHTUNG!** Der Disassembler des Bibomon kennt auch die Zusatzbefehle des **65C02**, des Processors in der **SPEEDY 1050**. Diese Befehle werden inverse dargestellt. Diese Befehle werden vom **6502C** der Atari XL-XE Computer nicht interpretiert und führen zum Systemabsturz!

Befehl: M - Verschiebe Speicher (MOVE)

Format: (ziel)(<)(start)(.)(end)(M)

Beispiel: 1000<9000.A080M

Erklärung: Der Bibomon bietet Ihnen die Möglichkeit, Speicherbereiche zu verschieben. Die Grösse des Speicherbereiches den Sie verschieben wollen, ist nur begrenzt durch die Grösse des Arbeitsspeichers Ihres Computers. So können Sie zum Beispiel das Betriebssystem Ihres Computers in den Rambereich verschieben und Veränderungen vornehmen. In unserem Beispiel wird der Speicherbereich von \$9000 bis \$A000 nach \$1000 verschoben.

Befehl: N - Register ändern

Format: (N)

Beispiel: N

Erklärung: Nach Eingabe von N fragt der Bibomon nach den neuen Werten für den Accu-, das X- und das Y- Register, sowie nach den neuen Werten für den Stackpointer und die Flags des Statusregisters. Durch einfaches Drücken der RETURN Taste wird der alte Wert übernommen, wenn dieser nicht geändert werden soll. Die geänderten Werte werden bei einem GOTO-Aufruf dem Programm übergeben.

Befehl: O - Öffne Bildschirm

Format: (O)

Beispiel: O

Erklärung: Das *BASIC* Gegenstück zu diesem Bibomon Befehl ist *GRAPHICS 0*. Was passiert nach Eingabe von O ? Nun zum einen wird die Uhr abgeschaltet, zum zweiten wird der Bildschirm gelöscht und der Curcor in die linke obere Bildschirmecke gesetzt. Sie werden diesen Befehl sehr hilfreich finden wenn Sie Veränderungen an der Displaylist vorgenommen haben und schnell wieder in den normalen Modus zurückkehren möchten.

Befehl: P - Druckersteuerung

Format: (P)

oder: (P1)

Beispiel: P

oder: P1

Erklärung: Bei dem 16K Bibomon haben Sie die Möglichkeit, alles was Sie auf dem Bildschirm sehen, auch über den Druckerport auf einem Drucker ausgeben zu lassen. Diese Druckerausgabe aktivieren Sie durch Eingabe von P1. Ab jetzt werden alle ASCII Zeichen die Sie auf dem Bildschirm sehen, auf dem Drucker ausgegeben. Normalerweise schalten Sie die Druckerausgabe durch Eingabe von P wieder ab. Verlassen Sie jedoch den Monitor vorzeitig, so wird die Druckerausgabe automatisch abgeschaltet. Sie haben übrigens auch die Möglichkeit, Steuerbefehle an den Drucker zu schicken. Dazu brauchen Sie den gewünschten Steuercode hinter dem P1 einzugeben. **Beispiel:** P1(ESC)4 stellt bei EPSON Druckern den Drucker auf Italic Schrift. (Um das ESC Zeichen zu bekommen drücken Sie bitte 2 mal die ESC Taste). Diese Funktion wird aber nicht durch den Bibomon wieder abgeschaltet und bleibt im Drucker.

Befehl: Q - Ende, Verlassen des Bibomon

Format: (Q)

Beispiel: Q

Erklärung: Wenn Sie vom Basic her, oder von einer anderen Programmiersprache, oder durch den **MON** Befehl des *BIBODOS* in den Monitor eingesprungen sind, können Sie ihn durch diesen Befehl wieder verlassen.

Befehl: R - Lese Sektor(en) - (READ)

Format: (Adresse)(<)(Start)(.)(End)(R)(,)

Beispiel: 3000<1R

oder: 3000<1.4R

oder: 3000<1R,

oder: 3000<1.4R,

Erklärung: Einer der grossen Vorteile des 16K Bibomon ist der direkte Diskettenzugriff *ohne* daß Sie ein DOS geladen haben. Mit dem Bibomon Befehl R können Sie zum Beispiel Sektoren von der Diskette in den Arbeitsspeicher Ihres Computers einlesen. Wie wird nun dieser Befehl angewendet? Sehen wir uns zuerst das Format dieses Befehls an. Sie geben zuerst die Speicheradresse an, wo der Bibomon die einzulesenden Sektoren ablegen soll. Diese Adresse ist in jedem Fall nur die Startadresse. Pro eingelesenem Sektor wird diese Bufferadresse automatisch jeweils um 128 oder 256 Bytes erhöht, je nach Density. Beim auslesen des RAM-Speichers der **SPEEDY 1050** wird der Sektor automatisch um jeweils 128 oder 256 Bytes erhöht. Nach der Bufferadresse folgt die Befehlskennung für den 16K Bibomon. Danach geben Sie den Startsektor der einzulesenden Sektoren an. Wollen Sie nur einen Sektor einlesen, geben Sie jetzt den Monitor Befehl R ein. Wollen Sie mehrere Sektoren einlesen, so geben Sie bitte einen Punkt ein. Danach die Nummer des Endsektors der Sektorgruppe, die eingelesen werden soll. Wenn Sie nun **RETURN** drücken, werden die von Ihnen angegebenen Sektoren eingelesen. Der letzte Teil des Befehles ist eine Option, die Sie nicht unbedingt benutzen müssen. Geben Sie hinter dem R ein Komma ein, so wird Ihnen der 16K Bibomon nach dem einlesen des letzten Sektors sofort die ersten 128 Bytes des von Ihnen angegebenen Speicherbereiches anzeigen. Sie haben dann sofort den ersten Sektor, bei **SINGLE-**, oder die erste Hälfte des ersten Sektors bei **DOUBLE DENSITY** auf dem Bildschirm. Haben Sie die gewünschten Sektoren erst einmal im Arbeitsspeicher, so können Sie sie nach Belieben verändern. Nach der Änderung können Sie diese dann auch wieder auf die Diskette schreiben, mit dem Bibomon Befehl **W**.

Befehl: S - Einzelschritt (SINGLESTEP)

Format: {Startadresse}(S)
oder: {Startadresse}(SSS)

Beispiel: 3000S
oder: 3000SSSS

Erklärung: Dieser Befehl entspricht dem nachfolgend beschriebenen **TRACE** Befehl T, mit einem Unterschied. Wenn Sie den **S** Befehl anwenden, wird immer nur ein Maschinensprach Befehl pro Durchlauf ausgeführt. Die Kontrolle über das Programm bleibt beim Bibomon.

Befehl: T - Verfolge Programm (TRACE)

Format: {Startadresse}(T)
Beispiel: 3000T

Erklärung: Zum Testen von kurzen Maschinenprogrammen können Sie diesen Befehl benutzen. Sie geben dazu nur die Startadresse des Programmes gefolgt von einem **T** ein. Der Bibomon gibt Ihnen daraufhin, beginnend mit der Startadresse, den Opcode und den Inhalt der Register aus. Der Tracer verfolgt das Programm, bis er auf ein **RTS** ohne vorhergehende **JSR** Anweisung trifft. Mit der **OPTION** Taste können Sie die Geschwindigkeit des Tracers erhöhen. Anhalten können Sie das Listing mit der Leerstate, Einzelschrittausgabe durch Druck der **RETURN** Taste. Sie verlassen den Tracer mit der **BREAK** Taste.

Befehl: U - Uhr an/aus

Format: (U)

Beispiel: U

Erklärung: Jedesmal, wenn Sie in den Bibomon einspringen, erscheint die Statuszeile mit der Uhr oberhalb des Bildschirms. Mit U können Sie diese Statuszeile nun abschalten oder wieder sichtbar machen, ganz wie Sie wollen. Das abschalten der Uhr ist zum Beispiel nützlich bei zeitkritischen I/O Operationen über die serielle Schnittstelle oder wenn Sie den VBD- Interrupt anderweitig nutzen wollen.

Befehl: V - Vergleiche Speicherbereiche

Format: (Zieladresse){<}(Startadr){.}(Endadr){V}

Beispiel: 1000<A000.E400V

Erklärung: Dieser Befehl gibt Ihnen die Möglichkeit zwei Speicherbereiche miteinander zu vergleichen. Sehen wir uns das Befehlsformat an. Als erstes geben Sie die Startadresse des Zielspeichers an. Das heisst, den Speicherbereich, wo das Programm steht, das Sie vergleichen wollen. Als zweites kommt ein Trennungssymbol für den Bibomon, gefolgt von der Startadresse und der Endadresse des zu vergleichenden Speicherbereiches. Getrennt werden die beiden Adressen von einem Punkt. Als letztes kommt dann der Befehl V. Falls die beiden Speicherbereiche nicht miteinander übereinstimmen, bekommen Sie eine Liste der Speicherbereiche, die unterschiedliche Werte enthalten, angezeigt. Sie sehen sowohl die Inhalte des Zielspeichers als auch den des Originalspeichers. Auch dieses Listing können Sie jederzeit durch den Druck auf die **LEER-TASTE** anhalten, oder weiter laufen lassen. Durch Druck auf die **BREAK** Taste verlassen Sie das Listing.

Befehl: W - Schreibe Sektor(en) - (WRITE)

Format: {Adresse}(<){Start}{.}{End}{W}

Beispiel: 3000<1W

oder: 3000<1.4W

oder: 3000<1W

Erklärung: Einer der grossen Vorteile des 16K Bibomon ist der direkte Diskettenzugriff *ohne* daß Sie ein DOS geladen haben. Mit dem Bibomon Befehl **W** können Sie zum Beispiel Speicherbereiche aus dem Computer Sektorenweise auf die Diskette abspeichern. Der Befehl **W** ist das genaue Gegenstück des Befehles **R** (**Sektor lesen**). Wie wird nun dieser Befehl angewendet? Sehen wir uns zuerst das Format dieses Befehls an. Sie geben zuerst die Speicheradresse an, von wo der Bibomon die Sektoren heraus schreiben soll. Diese Adresse ist in jedem Fall nur die Startadresse. Pro geschriebenen Sektor wird diese Bufferadresse automatisch jeweils um 128 oder 256 Bytes erhöht, je nach Density. Nach der Bufferadresse folgt die Befehlskennung für den 16K Bibomon. Danach geben Sie den Startsektor der zu schreibenden Sektoren an. Wollen Sie nur einen Sektor schreiben, geben Sie jetzt den Monitor Befehl **W** ein. Wollen Sie mehrere Sektoren schreiben, so geben Sie bitte einen Punkt ein. Danach die Nummer des Endsektors der Sektorgruppe, die geschrieben werden soll. Wenn Sie nun **RETURN** drücken, werden die von Ihnen angegebenen Sektoren auf die Diskette geschrieben. Wie beim READ Befehl wird auch hier die Sektornummer beim Schreiben in den Ramspeicher der **SPEEDY 1050** um 128 oder 256 Bytes erhöht.

Befehl: X - EXOR - Exklusiv Oder

Format: (Hex)(X)(Hex)

Beispiel: 2X3

Erklärung: Mit dieser EXOR Funktion können Sie zwei Hexadezimalzahlen miteinander vergleichen. Beim Vergleich zweier Bitmuster werden die gleichen Bitmuster logisch null, die ungleichen Bitmuster logisch eins. Das EXOR wird benutzt um Bitmuster zu invertieren.

Beispiele: 1X1=0 - 1X0=1 - 0X1=1 - 0X0=0

Befehl: Y - Zeilenassembler

Format: (Y)

Beispiel: Y

9000:LDA14

oder: 9000:LDA14

Erklärung: Mit dem 16K Bibomon steht Ihnen auch jederzeit ein leistungsfähiger Assembler zur Verfügung. Sie brauchen also für kurze Assemblerprogramme keinen speziellen Assembler mehr. Der Zeilenassembler des Bibomon ist sehr komfortabel und erlaubt das Eingeben des Quellcodes ohne Leerstellen oder Hexadezimal Vorzeichen. Alle Zahlen werden als Hexadezimalzahlen interpretiert. Um den Zeilenassembler anwenden zu können, geben zuerst einmal Y ein. Der Bibomon meldet sich dann mit (ASSEMBLER). Sie geben nun die Bufferadresse gefolgt von dem ersten Opcode ein. Zum verlassen des Zeilenassemblers geben Sie einfach Q ein. Hier ein Beispiel:

Ihre Eingabe

Y

9000:LDA14

STA2C6

JMP9000

Q

Ausgabe Bibomon

(ASSEMBLER)

9000: A5 14 LDA \$14

9002: 8D C6 02 STA \$2C6

9005: 4C 00 90 JMP \$9000

Wie Sie sehen, zeigt Ihnen der Bibomon gleich den Assembler- und den Opcode an. Nach der einmaligen Eingabe der Bufferadresse brauchen Sie nur noch den Opcode einzugeben. **Neu:** ab der Version 1.6 können Sie auch die Befehle des Processors **65C02** eingeben. Der **6502C** Processor der XL-XE Computer kennt diese Befehle allerdings nicht, und reagiert mit einem Systemabsturz auf eine solche Anweisung.

Befehl: Z - Verfolge File

Format: (Startsektor)(Z)
oder: (Startsektor)(ZL)
oder: (Startsektor)(ZR)

Beispiel: 1D1Z
oder: 1D1ZL
oder: 1D1ZR

Erklärung: Mit diesem Befehl können Sie *COM* oder *EXE* Files untersuchen. Der Bibomon zeigt Ihnen die Speicherbereiche an, wo das File im Speicher abgelegt wird. Geben Sie hinter dem Z ein L ein, wird das File zusätzlich in den Arbeitsspeicher eingelesen. Auch hierbei werden die Speicherbereiche angezeigt, wo das File abgelegt wird. Geben Sie hinter dem Z ein R ein, wird das File geladen und nach dem Laden automatisch gestartet. In diesem Falle werden die Speicherbereiche, wo das File abgelegt wird, nicht angezeigt, da sonst bei einer Bildschirmausgabe Speicherbereiche verändert werden können.

Befehl: : - Schreibe in Adresse - (STORE)

Format: (Hex):(Hex)

Beispiel: 9000:A9 00

Erklärung: Mit diesem und dem nachfolgenden Befehl können Sie Werte in die Speicheradresse schreiben, die Sie vor dem Doppelpunkt angegeben haben. Es können mehrere Werte in aufeinanderfolgende Speicherstellen geschrieben werden. Die einzelnen Werte müssen hierbei durch einen Leerschritt getrennt werden.

Befehl: ; - Schreibe in Adresse - (STORE)

Format: (Hex);(Hex)

Beispiel: 9000;A9 00

Erklärung: Dieser Bibomon Befehl entspricht genau dem vorher besprochenem Bibomon Befehl.

Befehle: . - Speicherauflistung - (DUMP)

Format: (Hex).(Hex)

Beispiel: 1000.2000

Erklärung: Die Anwendung dieses Befehles ist ganz einfach. Mit Eingabe der ersten Hexadezimalzahl geben Sie den Startbereich, mit der zweiten Hexadezimalzahl den Endbereich des zu Listenden Speicherbereiches an. Der Bibomon listet nun den gewünschten Speicherbereich auf. Wie Sie sehen, geht das sehr schnell. Sie können jedoch alle Listings durch Druck auf die **LEER** Taste anhalten. Wollen Sie das Listing weiter laufen lassen, so genügt ein zweiter Druck auf die **LEER** Taste. Wollen Sie jedoch nur eine oder mehrere einzelne Zeilen ausgegeben haben, so drücken Sie bei angehaltenem Listing auf die **RETURN** Taste. Wollen Sie das Listing unterbrechen, so drücken Sie einfach auf die **BREAK** Taste. Wie Sie sehen, ist die Bedienung des Bibomon sehr leicht und alle Funktionen des Bibomon sind sehr Benutzerfreundlich.

Befehl: , - Speicherlisting - (DUMP)

Format: (Hex),

oder: (HEX),,,,

Beispiel: 9000,

oder: 9000,,,

Erklärung: Dieser Befehl gibt Ihnen die Möglichkeit, immer eine halbe Seite (Page=128 Bytes) des Arbeitsspeichers Ihres Computers auf dem Bildschirm zu sehen. Da bei allen Funktionen des Bibomon Mehrfacheingaben in einer Zeile möglich sind, können Sie auch hier mehrere Komma's hintereinander eingeben. Sie erhalten dann so eine fortlaufende Speicherlisting. Auch dieses Listing können Sie durch Druck auf die **LEER** Taste anhalten, oder weiter laufen lassen. Durch Druck auf die **BREAK** Taste verlassen Sie das Listing.

Befehl: & - AND

Format: (Hex)&(Hex)

Beispiel: 2&3

Erklärung: Bei diesem Bibomon Befehl handelt es sich um einen Logischen Vergleich. Sind beide Bitmuster der Operatoren logisch eins, ist das Ergebnis logisch eins. Ist jedoch eines oder beide Bitmuster logisch null, erhalten wir als Ergebnis logisch null.

Beispiele: 1&1=1 - 1&0=0 - 0&1=0 - 0&0=0

Befehl: ^ - OR

Format: (Hex)^(Hex)

Beispiel: 1^1

Erklärung: Auch dieser Befehl ist ein logischer Vergleich. Er ist das genaue Gegenteil des zuvor besprochenem Befehles. Ist eines oder beide Bitmuster logisch eins, erhalten wir als Ergebnis logisch eins.

Beispiele: 1^1=1 - 1^0=1 - 0^1=1 - 0^0=0

Befehl: / - Letzte Disketten Operation

Format: (f)

Beispiel: /

Erklärung: Alle Daten der letzten Disketten Operation stehen in den Speicherbereichen \$0300 bis \$030F. Mit dem Bibomon Befehl / erhalten Sie eine Liste dieser Speicherstellen auf dem Bildschirm ausgegeben. Hier die einzelnen Adressen und Ihre Bedeutung:

Adresse - Label - Inhalt

```
=====
$0300  DDEVIC  Device Serial Bus ID
                Disk Drives D1 - D4 ($31-$34)
                Printer P1 ($40)
                Printer P1 ($4F)
                RS232 Ports R1 - R4 ($50-$53)
$0301  DUNIT   Laufwerknummer 1-4
$0302  DCOMND  Die Nummer der Disketten Operation
                Die Befehle:
                Read ($52)
                Write (verify) ($57)
                Status ($53)
                Put (no verify) ($50)
                Format Single ($21)
                Format Double ($22)
```

Alle diese Befehle sind Diskette-Befehle, mit Ausnahme von Write und Status, die auch Printer Befehle sind.

Adresse - Label - Inhalt

=====

\$0303	DSTAT	Der Status, der an den User zurück gegeben wird.
\$0304	DBUFLO	Datenbuffer Adresse für Disketten Operationen
\$0305	DBUFHI	im Low/High Format
\$0306	DTIMLO	Time Out Wert für den Händler, in 1 Sekunden Schritten
\$0307	DUNUSE	nicht benutzt- immer 0
\$0308	DBYTLO	Die Anzahl der Bytes, die zum Datenbuffer
\$0309	DBYTHI	übertragen werden. Low/High Format
\$030A	DAUX1	Disketten Sektor der gelesen oder
\$030B	DAUX2	geschrieben wurde. Low/High Format

Diese Adressen sind die wichtigsten Adressen, die bei einer Disketten Operation genutzt werden.

Befehl: = - Fülle Speicherbereich

Format: (Hex)<(Start).(End)=

Beispiel: 00<1000.A000=

Erklärung: Mit Hilfe dieses Bibomon Befehles können Sie ganze Speicherblöcke mit einem Wert füllen. Die Anwendung dieses Befehles ist wieder sehr leicht. Als erstes geben Sie den Wert ein, mit dem der Speicherbereich gefüllt werden soll. Als zweites kommt wieder die Trennungsanweisung für den Monitor. Danach die Start- und die Endadresse des Speicherbereiches den Sie füllen möchten, getrennt von einem Punkt. Als letztes dann der Füllbefehl. In unserem Beispiel wird der Speicherbereich von **\$1000** bis **\$A000** mit dem Wert **\$00** gefüllt.

Befehl: \$ - Wandelt Hexadezimalzahlen in Dezimalzahl

Format: \$(Hex)

Beispiel: \$AF

Erklärung: Alle Zahlen, die Sie eingeben, werden vom Bibomon als Hexadezimalzahlen interpretiert. Mit diesem Befehl haben Sie aber die Möglichkeit, sich alle Hexadezimalzahlen in Dezimalzahlen umrechnen zu lassen. Dazu brauchen Sie nur *vor* der Zahl die Sie umgerechnet haben wollen, das Hexadezimal Zeichen zu setzen, wie in unserem Beispiel.

Befehl: # - Wandelt Dezimalzahlen in Hexadezimalzahlen

Format: #(Dezimal)

Beispiel: #34

Erklärung: In gewisser Weise stellt dieser Befehl das genaue Gegenstück zu dem oben beschriebenen Befehl dar. Jetzt können Sie Dezimalzahlen in Hexadezimalzahlen umwandeln. Die Syntax entspricht dem des vorherigen Befehles.

Befehl: ' - Texteingabe in Bildschirmcode

Format: (Hex)'(Ascii)

Beispiel: 1000'TEST

Erklärung: Wie bereits erwähnt, haben Sie immer die Möglichkeit Text gleich im Bildschirmcode darzustellen, einzugeben oder zu suchen. Das Suchen nach Bildschirmcode wird etwas später besprochen. Zur Darstellung des Bildschirmcode's während eines Listings (*DUMP*) drücken Sie einfach während des laufenden Listings die *OPTION* Taste. Sie werden dann im rechten Teil des Bildschirms, bei der ASCII Darstellung, sofort die Änderung feststellen. Wird zum Beispiel der Wert Null bei den ASCII Zeichen als Herzchen dargestellt, so werden Sie den gleichen Wert im Bildschirmcode als Leerzeichen finden. Wenn Sie nun Text gleich im Bildschirmcode eingeben wollen, so benutzen Sie bitte die folgende Syntax. Zuerst geben Sie die Speicherstelle an, in die Sie den Wert oder die Werte schreiben wollen. Danach geben Sie bitte den einzelnen Anführungsstrich ein. Als letztes kommt nun der Text, den Sie als Bildschirmcode ablegen möchten.

Befehl: " - ASCII code Eingabe

Format: (Hex)"(Ascii)

Beispiel: 1000"TEST

Erklärung: Genau wie beim vorherigen Befehl, haben Sie auch hier die Möglichkeit Text gleich im ASCIIcode darzustellen, einzugeben oder zu Suchen. Das suchen nach ASCIIcode wird etwas später besprochen. Wenn Sie nun Text gleich im ASCIIcode eingeben wollen, so benutzen Sie bitte die folgende Syntax. Zuerst geben Sie die Speicherstelle an, in die Sie den Wert oder die Werte schreiben wollen. Danach geben Sie bitte die Anführungsstriche ein. Als letztes kommt nun der Text, den Sie als ASCIIcode ablegen möchten. Nach Eingabe dieser ersten Anführungsstriche, werden alle folgenden Anführungsstriche als normale ASCII Zeichen erkannt.

Befehl: > - Suche Speicherbereich ab

Format: (Hex).(Hex)>(Hex) (Hex)

oder: (Hex).(Hex)>"(ASCII)

oder: (Hex).(Hex)>'(ASCII)

Beispiel: 1000.2000>A9 09

oder: 1000.2000>"TEST

oder: 1000.2000>'TEST

oder: 0>"TEST

Erklärung: Suchen Sie bestimmte Daten im Arbeitsspeicher Ihres Computers? Dann benutzen Sie diesen Befehl. Wie Sie aus unseren Beispielen erkennen können, ist der Suchbefehl sehr vielseitig. Sie können nach Daten in Hexadezimaler Form oder nach ASCII Zeichen oder nach Bildschirmcode suchen. In unseren ersten drei Beispielen wird der Speicherbereich von **\$1000** bis **\$2000** abgesucht. In unserem dritten Beispiel sucht der Monitor den gesamten Speicherbereich nach der ASCII Folge **"TEST"** ab. Wie Sie aus unseren Beispielen ersehen können, ist auch die Anwendung dieses Befehles sehr leicht. Eine Bemerkung noch. Die eingegebenen Daten in ASCII oder ATASCII (Bildschirmcode) werden immer im Bereich des Bildschirmspeichers von **\$9000** bis **\$BFFF** und/oder im Bereich des Eingabebuffers **\$D800** bis **\$DFFF** gefunden.

Befehl: ! - System Adressen

Format: (!)

Beispiel: !

Erklärung: Wie wir Ihnen bereits früher erklärt haben, gibt es mehrere Wege in den Bibomon zu gelangen. Aber nur bei dem Weg über den Druck auf die Tasten **SELECT** und **SYSTEM RESET** werden die wichtigsten Adressen des Programmes, das Sie gerade unterbrochen haben, abgespeichert. Und nun können Sie durch Eingabe dieses Bibomon Befehles die wichtigsten Adressen des Programmes sofort sichtbar machen. Nach Eingabe des Befehles erhalten Sie die folgende Tabelle. Sie gibt Ihnen die Adressen wieder, die beim Einsprung in den Bibomon unterbrochen wurden.

VDSLST	0200/01	-	E790
VKEYBD	0208/09	-	FFBE
VIMIRQ	0216/17	-	E706
VVBLKI	0222/23	-	E7AE
VVBLKD	0224/25	-	E905
SDLSTL	0230/31	-	BC20
SDMCTL	022F	-	22
COLDST	0244	-	00
COLOR0	02C4	-	28
COLOR1	02C5	-	CA
COLOR2	02C6	-	90
COLOR3	02C7	-	46
COLOR4	02C8	-	00

Was bedeuten nun alle dieses Adressen? Nachfolgend finden Sie nun eine Tabelle mit den Erklärungen dieser Adressen und Ihrer Funktionen.

Label	Adresse	Erklärung
VDSLST	\$0200/01	Diese beiden Speicherstellen sind der Vektor für den DLI
VKEYBD	\$0208/09	Vektor für den Tastatur Interrupt
VIMIRQ	\$0216/17	Maskierbarer Interrupt (IRQ)
VVBLKI	\$0222/23	Vertical Blanc Interrupt (VBI)
VVBLKD	\$0224/25	Einsprung für den Deffered VBI
SDLSTL	\$0230/31	Startadresse der Displayliste
SDMCTL	\$022F	Direkter Speicher Zugriff (DMA) Die ANTIC kann über diese Adresse abgeschaltet werden. Hier einige Daten:

Hex	Dez	Funktion
\$20	32	DMA ein/au
\$10	16	Einzeilige PM Auflösung.
\$08	8	DMA für Player
\$04	4	DMA für Missile
\$03	3	Breites Spielfeld
\$02	2	Normales Spielfeld
\$01	1	Kleines Spielfeld
\$00	0	ANTIC aus

COLDST	\$0244	Kaltstartflag. Steht in dieser Adresse ein Wert grösser als null, wird bei Druck auf SYSTEM RESET neu gebootet.
COLOR0	\$02C4	Wert für Setcolor 0
COLOR1	\$02C5	Wert für Setcolor 1
COLOR2	\$02C6	Wert für Setcolor 2
COLOR3	\$02C7	Wert für Setcolor 3
COLOR4	\$02C8	Wert für Setcolor 4

Befehl: * - DOS Kommando

Format: (*)

Beispiel: *H

Erklärung: Dieser Bibomon Befehl ist bereits ein Teil des DOS. Mit dieser Anweisung sagen Sie dem Bibomon, daß der nachfolgende Befehl ein DOS Befehl ist. Sie aktivieren also mit diesem Befehl das DOS. Eine genaue Erklärung des Befehles finden Sie im Kapitel über das DOS.

Befehl: ? - Register Anzeige

Format: (?)

Beispiel: ?

Erklärung: Wenn Sie in den Bibomon einspringen, erhalten Sie als erstes die Register Anzeige. Diese Anzeige können Sie mit diesem Befehl sichtbar machen. Was bedeuten nun diese Kürzel und Abkürzungen, die Sie da sehen? Nun als erste sehen Sie **PC**, die Abkürzung für **PROGRAMM COUNTER**. Dieser Counter sagt Ihnen, aus welcher Speicherstelle heraus Sie in den Monitor eingesprungen sind. Sie sehen dann sofort, an welcher Stelle Sie ein Programm unterbrochen haben. Die nächsten drei Abkürzungen **A**, **X** und **Y** geben Ihnen den Inhalt der Entsprechenden Register an. **S** steht für Stack, hier erhalten Sie den Inhalt des Stackpointer angezeigt. Das letzte Bitmuster zeigt Ihnen den Status der Flags an. Hier die Bedeutung der Kürzel. **N** = Negativ Flag, **V** = Überlauf Flag, **.** = Unbenutzt, **B** = Break Flag, **D** = Dezimal Flag, **I** = IRQ Flag, **Z** = Zero Flag, **C** = Carry Flag. Mit Hilfe eines anderen Bibomon Befehles können Sie alle Inhalte der Register ändern. Dazu geben Sie einfach **N** ein. Eine genaue Erklärung dieses Befehles finden Sie etwas früher in diesem Handbuch.

Befehl: @ - Hauptmenu

Format: (@)

Beispiel: @

Erklärung: Sie haben jederzeit die Möglichkeit mit diesem Befehl wieder zurück in das Hauptmenu zu gelangen. Von hier aus gelangen Sie dann in den Monitor oder in das Hardcopy Setup Menu. Oder Sie können ganz leicht einen erneuten Kaltstart ausführen.

Befehl: \ - Bank Umschalter

Format: (\)

Beispiel: \F

Erklärung: Der 16K Bibomon ist in der Lage mit Ramspeichern bis zu einer Grösse von 320K zu arbeiten. Der Bibomon passt sich selbständig an die jeweilige Ram-grösse an. Sie haben aber die Möglichkeit, die Bank's von Hand umzu-schalten. Sie haben dann ab dem Speicherbereich \$4000 einen neuen 16K Block zur Verfügung. Dieser Block reicht bis \$7FFF. Um nun eine neue Bank zu aktivieren, brauchen Sie sich nur an unser Beispiel zu halten. Mit dieser Eingabe wird die letzte Bank (320K) umgeschaltet. Um nun wieder in den Grundrambereich zu gelangen, geben Sie einfach \ ohne einen Wert ein. Achtung! Bank 0 ist bereits die erste Bank in Zusatzspeicher. Die höchsten erreichbaren Bank's:

Bei 128K Ram - von Bank 00 bis Bank 03

192K Ram - von Bank 00 bis Bank 07

320K Ram - von Bank 00 bis Bank 0F

Befehl: [- Hardcopy Setup Menu

Format: ([])

Beispiel: [

Erklärung: Um vom Bibomon her das Hardcopy Menu zu ändern, geben Sie einfach diesen Befehl ein. Sie gelangen dann sofort in das Hardcopy Menu. Eine genaue Beschreibung dieses Menu's finden Sie im Kapitel **HARDCOPY**.

So, damit hätten wir alle Bibomon Befehle besprochen. Mit der Zeit werden Sie feststellen, daß Sie mit diesen Befehlen sehr schnell vertraut werden und daß Ihnen das Arbeiten mit Ihrem Computer leichter fällt.

3.0 DAS DOS DES BIBOMON

Um nun die Funktionsweise des DOS richtig zu verstehen, sind einige Kenntnisse über den Aufbau der Datenstruktur auf einer Diskette notwendig.

Zunächst einmal reicht es, wenn Sie wissen, daß es bei den ATARI Laufwerken drei verschiedene Speicherdichten gibt. Da wäre zuerst einmal die wohl bekannte **DOS 2.0** Speicherdichte, 88K Byte. Diese 88K Byte werden zu jeweils 18 Sektoren in 40 Spuren (TRACK'S) mit jeweils 128 Bytes per Sektor aufgeteilt. Dieses Format nennt man **SINGLE DENSITY**.

Das zweite Format ist das sogenannte **DOS 3** oder **1050er** Format. Hier werden 26 Sektoren in 40 Spuren untergebracht. Die Dateien werden in Blöcken zu je 1K Byte Länge abgespeichert. Diese Methode ist sehr platzraubend. Insgesamt erhalten Sie aber bei diesem Format 127K Byte Speicherplatz auf einer Diskette. Man nennt dieses Format **MEDIUM DENSITY**. Es gibt aber inzwischen eine modifizierte **DOS 2** Version, das **DOS 2.5**. Mit diesem DOS erhalten Sie auch 127K Byte Speicherplatz, aber im **DOS 2** Format.

Das dritte Format für die **ATARI** Laufwerke ist die **DOUBLE DENSITY**. Sie erhalten hier wieder 18 Sektoren pro Spur, aber mit 256 Bytes per Sektor. Sie erhalten also 176K Byte Speicherplatz. Dieses geht aber nur mit Fremdlautwerken oder modifizierten **ATARI 1050** Lautwerken. (Zum Beispiel mit einer **SPEEDY 1050**.) Das DOS für diese Speicherdichte ist das **DOS XL** von **OPTIMIZED SYSTEM SOFTWARE**, kurz **OSS**.

Welche Formate können Sie nun mit dem DOS des Bibomon lesen und schreiben?

Das **DOS** des Bibomon ist so ausgelegt, daß Sie alle diese Formate lesen und schreiben können. Aber wie so oft gibt es auch hier eine Ausnahme. Das **DOS 3.0** Format ist, wie wir weiter oben erklärt haben, nicht kompatibel zu den anderen **DOS** Varianten. Dieses Format kann mit dem **DOS** des Bibomon **nicht** gelesen oder geschrieben werden.

Hier noch einmal der Hinweis, daß das **DOS** des Bibomon voll kompatibel ist zum **DOS XL 2.3** von **OSS**.

Nun nachdem Sie etwas über die unterschiedlichen Speicherdichten erfahren haben, jetzt etwas über das Format auf der Diskette selbst. Da der einzige Unterschied zwischen *SINGLE* und *DOUBLE DENSITY* in der Anzahl der Bytes pro Sektor liegt, gilt die Erklärung für beide Speicherdichten.

Um nun eine Diskette beschreiben zu können, müssen Sie diese erst einmal **FORMATIEREN**. Was passiert nun bei diesem Formatieren?

Das DOS teilt die Diskette in 40 Kreise (SPUREN) ein. Jede dieser Spuren wird dann in 18 Teilkreise (SEKTOREN) unterteilt. Jeder dieser Sektoren beinhaltet nun 128 Bytes. Rechnen Sie nun $40 \text{ Tracks} * 18 \text{ Sektoren} * 128 \text{ Bytes}$ erhalten Sie $(40 * 18 * 128) = 91460$ Bytes. Da das *SINGLE DENSITY* Format aber nur 88K Byte, also 88000 Bytes, Speicherplatz frei lässt, wo bleiben die restlichen 3460 Bytes?

Nun, nachdem das DOS die Diskette *Formatiert* hat, werden noch die drei Boot Sektoren, die **VTOC** und die **Directory** geschrieben.

Die Boot Sektoren sind wichtig. Denn dort steht entweder die Lade Routine für das DOS (falls ein DOS auf der Diskette steht) oder eine kurze Routine damit Ihr Rechner nicht abstürzt, wenn Sie diese Diskette booten und es ist kein DOS auf dieser Diskette. Sie kennen ja vielleicht das berühmte *BOOT ERROR*. Diese drei Sektoren sind immer in *SINGLE DENSITY* geschrieben, egal ob Sie die Diskette in *SINGLE, MEDIUM* oder *DOUBLE DENSITY* formatiert haben.

Die **VTOC** ist die Belegungs Tabelle der Diskette. Anhand dieser Tabelle kann das DOS feststellen, ob und wo auf der Diskette noch Platz ist. Sie sehen das Ergebnis regelmässig wenn Sie die Directory abfragen. Sie erhalten am Schluss die Meldung "**### FREE SEKTORS**". Diese **VTOC** liegt im Normalfall im Sektor **\$168 (368)**. In den darauf folgenden Sektoren **\$169 (369)** bis **\$16F (377)** finden Sie die **Directory**, also das Inhaltsverzeichnis der Diskette. In diesem Inhaltsverzeichnis stehen neben dem Namen der Datei noch die Informationen ob die Datei geöffnet, gesichert oder gelöscht ist. Danach kommt die Information, welcher Sektor der Startsektor der Datei ist. Die nächsten zwei Byte geben Ihnen dann noch die Auskunft wie lang das File ist.

Mit diesen Informationen ausgerüstet, werden Sie einige Funktionen des **Bibodos** besser verstehen.

3.1 Der Befehlssatz des DOS

Auch bei der Besprechung der DOS Befehle geben wir Ihnen zuerst einen kompletten Überblick über alle DOS Befehle. Hier ist die Liste:

D - Inhaltsverzeichnis
I - Disketten Index
L - Load File
R - Run File
T - Trace File
G - Get Bin-File
V - View File
S - Save File (COM)
W - Save File (BIN)
P - Protect File
U - Unprotect File
E - Erase File
N - Rename File
F - Format Disk

Alle diese Befehle werden wir Ihnen jetzt noch einmal genauer erklären. Vorweg aber noch diese Bemerkungen. Vor jedem DOS Befehl muss ein Sternchen stehen. Zum Beispiel *D für die Directoryabfrage. Ausserdem arbeitet das DOS des Bibomon mit der *SPEEDY 1050* zusammen. Das bedeutet für alle Besitzer einer *SPEEDY 1050*, das Sie mit der hohen Geschwindigkeit der *SPEEDY* arbeiten können. Nun aber zu den Erklärungen.

3.2 Die Befehle des DOS

Befehl: *D - Directory Abfrage

Format: (*D)n(Name)

Beispiel: *D

oder: *D1 P*.*

DOS 2.0 Gegenstück: A (RETURN)

DOS XL Gegenstück: DIR (RETURN)

Erklärung: Wie Sie sehen, ist die Anwendung dieses Befehles ganz leicht. Wenn Sie nach dem *D die RETURN Taste drücken, erhalten Sie sofort das Inhaltsverzeichnis der Diskette in Lautwerk 1. Sie können nach dem D jede Lautwerksnummer angeben, von jedem angeschlossenen Lautwerk. Dabei spielt es keine Rolle, ob die Diskette nun in *SINGLE*- oder *DOUBLE DENSITY* beschrieben ist.

Befehl: *I - Index Abfrage

Format: (*I)n(Name)

Beispiel: *I

oder: *I| P*,*

DOS 2.0 Gegenstück: nicht vorhanden

DOS XL Gegenstück: nicht vorhanden

Erklärung: Dieser DOS Befehl entspricht dem Bibomon Befehl I. Entgegen der normalen Directory Abfrage mit D erhalten Sie hier nicht nur den Inhalt der Diskette angezeigt, sondern auch noch viele Informationen über die Programme, die auf der Diskette stehen. Wir geben Ihnen hier ein Beispiel:

```
*I
ST NAME          LEN $LEN $BEG
62 TESTPROGBAS  054 0036 0004
62 TESTPROGCOM  041 0029 003A
80 TEST        COM 099 0063 0063
42 APPA        BAS 008 0008 00C6
43 PROGDAT    OBJ 038 0026 01FA
$0007 007 FREE SECTORS
```

Was können Sie nun aus diesen Daten erkennen? Nun zuerst sehen Sie in der oberen Zeile eine ganze Reihe kürzel. **ST** steht für **Status**, **NAME** nennt Ihnen den Namen des Programmes, **LEN** gibt Ihnen die länge des Files in Dezimaler, **\$LEN** in Hexadezimaler Notation und schließlich **\$BEG** gibt Ihnen den Startsektor der Datei in Hexadezimal an.

Unter dem Kürzel **ST** sehen Sie vier verschiedene Daten. Es handelt sich dabei um Hexadezimal Zahlen. **\$62** bedeutet, daß die Datei gegen Überschreiben geschützt ist. **\$80** bedeutet, daß die Datei gelöscht ist. Wenn Sie Glück haben, sind die Daten der Datei aber noch erhalten und Sie können Sie mit Hilfe des Bibomon noch retten. Der Status **\$42** ist der normale Status einer Datei. Sie sehen aber noch eine Datei mit dem Status **\$43**. Wenn Sie mit einem normalen DOS versuchen diese Datei zu laden, erhalten Sie eine Fehlermeldung. **\$43** bedeutet, daß diese Datei zum Schreiben geöffnet wurde um mit ihr zu arbeiten. Aus irgendwelchen Gründen ist sie dann nicht mehr geschlossen worden. Daher können solche Dateien nicht mehr gelesen werden. Aber auch hier können Sie mit dem Bibomon Abhilfe schaffen. Aber seien Sie **vorsichtig!** Sie können zwar mit dem Bibomon die Sektoren der Directory einlesen und alle Daten verändern, aber Sie müssen dann auch gleich die **VTOC** ändern. So können Sie gelöschte Files (Status **\$80**) leicht wieder mit dem Status **\$42** versehen, aber in der **VTOC** sind die Sektoren, die das File normalerweise belegt, als frei eingetragen. Also, wenn Sie nun den Wert **\$80** in **\$42** oder **\$62** ändern, können Sie die Daten des Files wohl wieder einlesen, aber die **VTOC** stimmt dann nicht mehr. Das bedeutet, daß Sie die ganze Diskette nicht mehr beschreiben dürfen. Diese Files lassen sich aber ganz normal lesen, also auch kopieren!

Befehl: *L - Load File

Format: (*L)n(Name),(Buffer)

Beispiel: *L TEST.COM

oder: *LZ TEST.COM,A000

DOS 2.0 Gegenstück: nicht vorhanden

DOS XL Gegenstück: nicht vorhanden

Erklärung: COM und EXE Dateien können mit diesem Befehl geladen werden. Sie haben aber auch die Möglichkeit, das Programm in einen anderen als den Originalen Speicherbereich laden zu lassen. Dazu brauchen Sie nur die neue Ladeadresse hinter dem Namen des Programmes einzugeben. Aber auch hier gibt es Einschränkungen. Nur der erste Block des Programmes wird zu der neuen Adresse geladen. Alle anderen Blöcke werden gar nicht eingelesen, weil das DOS nicht weiss, wo diese Blöcke abgelegt werden sollen.

Befehl: *R - Run File

Format: (*R)n(Name)

Beispiel: *R TEST.COM

oder: *R2 TEST.COM

DOS 2.0 Gegenstück: L(RETURN)

TEST.COM

DOS XL Gegenstück: TEST(RETURN)

Erklärung: Entspricht dem gerade besprochenem Befehl L, nur mit dem Unterschied das das File bei dem Befehl R geladen und gestartet wird.

Befehl: *T - Trace File

Format: (*T)n{Name),(L)

Beispiel: *T TEST.COM

oder: *T2 TEST.COM,L

DOS 2.0 Gegenstück: nicht vorhanden

DOS XL Gegenstück: nicht vorhanden

Erklärung: Mit diesem DOS Befehl können Sie **COM** und **EXE** Files **Tracen** und **Laden**. Was bedeutet das nun für Sie? Nun **Tracen** bedeutet, das Sie die Adressen erhalten, von wo bis wo die Datei in den Arbeitsspeicher geladen wird. Sollten Sprünge in den Ladeadressen vorkommen, so werden auch diese angezeigt. Zusätzlich erhalten Sie die jeweiligen Startsektoren mit dem Beginn der neuen Ladeadresse angezeigt. Nach dem Tracevorgang bekommen Sie, bei **COM** und **EXE** Dateien, die Startadresse der Datei angezeigt. Geben Sie nach dem Namen der Datei ein **L** ein, so werden Ihnen nicht nur die Adressen und Startsektoren angezeigt, sondern das File wird auch in den Arbeitsspeicher Ihres Computers eingelesen.

Befehl: *G - Lade Binär File (Datenfile)

Format: (*G)n(Name),(Buffer)

Beispiel: *G TEST.OBJ,3000

oder: *G2 TEST.OBJ,4000

DOS 2.0 Gegenstück: nicht vorhanden

DOS XL Gegenstück: nicht vorhanden

Erklärung: Ein Datenfile, oder Binärfile, besteht nur aus Daten. Das heißt, das am Anfang der Datei weder Start- noch eine Endadresse angegeben ist. Sie müssen also, wenn Sie eine solche Datei laden wollen, nach dem Lade Befehl die Bufferadresse angeben, wo das Datenfile oder das Binärfile abgelegt werden soll.

Befehl: *V - View File

Format: (*V)n(Name)

Beispiel: *V TEST.OBJ

oder: *V2 TEST.OBJ

DOS 2.0 Gegenstück: C(RETURN)

TEST.OBJ,S:(RETURN)

DOS XL Gegenstück: TYPE TEST.OBJ

Erklärung: Ein sehr nützlicher Befehl. Mit seiner Hilfe können Sie sich jedes File ansehen, das Sie sehen wollen. Die Anwendung ist sehr leicht. Nach dem *V geben Sie einfach den Namen des Files ein, das Sie sehen wollen. Gleichzeitig ist natürlich eine Druckerausgabe möglich! (Mit dem Bibomon Befehl P1) So können Textfiles ganz leicht ausgedruckt werden.

Befehl: *S - Save COM File

Format: (*S)n(Name),(Beg),(End)

Beispiel: *S TEST.COM,2000,3FFF

oder: *S2 TEST.COM,2000,3FFF

DOS 2.0 Gegenstück: K(RETURN)

TEST.COM,2000,3FFF,2000,2000(RETURN)

DOS XL Gegenstück: SAV TEST.COM 2000 3000

Erklärung: Nach Eingabe von *S TEST.COM,2000,3FFF wird der Speicherbereich von \$2000 bis \$3FFF als COM File auf die Diskette in Laufwerk 1 abgespeichert. Natürlich können Sie auch hier, wie bei allen DOS Befehlen hinter dem Befehl eine Laufwerknummer angeben. Machen Sie von dieser Option keinen Gebrauch, wird Laufwerk 1 als gegeben voraus gesetzt.

Befehl: *W - Save BIN File (Daten File)

Format: (*W)n(Name),(Beg),(End)

Beispiel: *W TEST.BIN,2000,3FFF

oder: *W2 TEST.BIN,2000,3FFF

DOS 2.0 Gegenstück: nicht vorhanden

DOS XL Gegenstück: nicht vorhanden

Erklärung: Dieser Befehl entspricht dem gerade besprochenem Befehl *S, mit einem großen Unterschied. Nach Eingabe von *W TEST.COM,2000,3FFF wird der Speicherbereich von \$2000 bis \$3FFF als BIN File, also als Datenfile, auf die Diskette in Laufwerk 1 abgespeichert.

Befehl: *P - Schütze File (Protect File)

Format: {*P)n(Name)

Beispiel: *P TEST.BIN

oder: *P2 TEST.BIN

DOS 2.0 Gegenstück: F(RETURN)
TEST.BIN(RETURN)

DOS XL Gegenstück: PRO TEST.BIN

Erklärung: Dieser Befehl entspricht dem DOS 2.0 Befehl F. Sie können mit diesem Befehl Dateien gegen ein unbeabsichtigtes überschreiben schützen. Das Gegenstück ist der folgende Befehl.

Befehl: *U - Entschütze File (Unprotect File)

Format: {*U)n(Name)

Beispiel: *U TEST.BIN

oder: *U2 TEST.BIN

DOS 2.0 Gegenstück: G(RETURN)
TEST.BIN(RETURN)

DOS XL Gegenstück: UNP TEST.BIN

Erklärung: Dieser Befehl entspricht dem DOS 2.0 Befehl G. Sie können mit diesem Befehl Dateien, die Sie mit dem P Befehl gegen ein unbeabsichtigtes überschreiben gesichert haben, wieder entsichern.

Befehl: *E - Lösche File

Format: (*E)n(Name)

Beispiel: *E TEST.BIN

oder: *E2 TEST.BIN

DOS 2.0 Gegenstück: D(RETURN)
TEST.BIN(RETURN)

DOS XL Gegenstück: ERA TEST.BIN

Erklärung: Zum löschen von Dateien benutzen Sie diesen Befehl. Auch seine Anwendung ist sehr leicht und bedarf keiner grossen Erklärung. Sie geben einfach *E gefolgt von dem Namen des Files ein, das Sie löschen wollen. Aber seien Sie vorsichtig bei der Anwendung dieses Befehles. Haben Sie einmal ein File gelöscht, ist es sehr schwer diesen Befehl wieder rückgängig zu machen.

Befehl: *N - File Umbenennen

Format: (*N)n(Name alt),(Name neu)

Beispiel: *N TEST.BIN,TEST.COM

oder: *N2 TEST.BIN,TEST.COM

DOS 2.0 Gegenstück: E(RETURN)
TEST.BIN,TEST.COM(RETURN)

DOS XL Gegenstück: REN TEST.BIN TEST.COM

Erklärung: Wenn Sie Dateien neue Namen geben wollen, können Sie das mit diesem Befehl machen. Sie brauchen lediglich nach *N den alten und den neuen Namen der Datei, getrennt von einem Komma, eingeben. Nach Druck auf die RETURN Taste wird der alte Name des Files sofort von dem neuen Namen ersetzt.

Befehl: *FS - Formatiere Single Density
Befehl: *FD - Formatiere Double Density

Format: (*FS)
Format: (*FD)
Beispiel: *FS
 oder: *FD
 oder: *F2D

DOS 2.0 Gegenstück: I(RETURN)
 1(RETURN)
 J(RETURN)

DOS XL Gegenstück: INIT
 Init ist ein Hilfsprogramm auf der DOS XL Masterdisk

Erklärung: Der DOS Befehl *F ist ein vielseitiger Befehl. Sie haben die Möglichkeit, eine Diskette in **Single** oder **Double** Density zu formatieren. Sie können aber auch nur die Bootsektoren, die **VTOC** und die Directory neu schreiben. Wenn Sie *FS oder *FD eingegeben haben erscheint die folgende Abfrage:

*FS
Format SD Drive 1 (Y/N/C)?

Sie haben an dieser Stelle drei Wahlmöglichkeiten. Sie können die Diskette formatieren, dann drücken Sie **Y**. Wollen Sie diese Funktionen gar nicht ausführen, so drücken Sie **N**. Wollen Sie nur die **VTOC** und die Directory leeren (neu schreiben), so drücken Sie einfach **C**. Beim neuschreiben der Bootsektoren, der **VTOC** und der Directory bleiben alle alten Daten auf der Diskette erhalten.

3.3 Anwendung des DOS

Nachdem wir Ihnen nun alle Befehle des **DOS** erklärt haben, möchten wir Ihnen noch ein paar Tips zum Umgang mit dem **DOS** geben.

Das **DOS** des Bibomon ist ein **Arbeitsdos**. Das bedeutet, das Sie kein Menu erhalten, wenn Sie vom **BASIC** oder einer anderen Programmiersprache, **DOS** eingeben.

Was ist aber nun der Vorteil dieses **DOS**?

Der erste Vorteil ist, das es ständig im Arbeitsspeicher Ihres Computers steht. So können Sie zum Beispiel jede Programmiersprache im Steckmodul benutzen und mit dem **BibODOS** arbeiten. Als Beispiel nehmen wir hier das **Atari BASIC**.

Schalten Sie Ihre Diskettenstation aus und Ihren Computer ein. Wenn das **BASIC** sich mit **READY** meldet, schreiben Sie bitte ein kurzes Programm. Nachdem Sie Ihr Programm geschrieben haben, möchten Sie es gerne abspeichern. Aber die Diskette, die Sie da haben, ist die überhaupt leer? Oder ist Sie gar schon formatiert?

Um das festzustellen, gehen Sie nun bitte in den **BIBOMON** durch Druck auf **SELECT** und **SYSTEM RESET**. Sehen Sie sich die Directory mit dem **DOS** Befehl ***D** an. Ist genügend Platz auf der Diskette oder müssen Sie sie erst mit dem **DOS** Befehl ***F** formatieren? Wenn Sie sicher sind, das Sie genügend Platz auf der Diskette haben, kommen Sie über den Druck auf **SYSTEM RESET** wieder in das **BASIC** zurück. Listen Sie Ihr Programm auf, Sie werden sehen, es ist noch da.

Um nun das Programm auf die Diskette abspeichern zu können, müssen Sie zuerst einmal das **DOS** aktivieren. Durch Druck der Tasten **START** und **SYSTEM RESET** aktivieren Sie das **DOS**. Jetzt können Sie Ihr Programm abspeichern.

Geben Sie nun **NEW** ein und fragen Sie den freien Speicherplatz durch **PRINT FRE(0)** ab. Wie Sie sehen, verlieren Sie kein Byte Speicherplatz durch das **DOS** des Bibomon. Aber welche Befehle stehen Ihnen nun zur Verfügung?

Beim **Atari BASIC** nur **LOAD** und **SAVE**, Bei Programmiersprachen die eine eingebaute Directoryabfrage haben (zum Beispiel **BASIC XL** oder **ACTION!** von **OSS**), funktioniert auch diese.

Ab der Version 1.6 besteht nun die Möglichkeit das Interne **DOS** direkt beim einschalten des Computers zu aktivieren. Drücken Sie die **OPTION** Taste und schalten Sie Ihren Computer ein. Ist das **BASIC** oder eine andere Programmiersprache im **MODUL** vorhanden, wird die Diskette nicht angebootet und Sie haben den gesamten Speicherplatz frei für Ihre Programme, obwohl Sie nun auch das **DOS** zur Verfügung haben. Wenn Sie das **DOS** auf diesem Wege aktiviert haben, bleibt es auch aktiv wenn Sie die **SYSTEM RESET** Taste drücken.

Anhand dieser Beispiele sollten Sie die Leistungsfähigkeit des **DOS** erkannt haben, und die Möglichkeiten die in ihm stecken.

4.0 Die Hardware Uhr

Mit dem **16K Bibomon** haben Sie nun auch eine Hardware Uhr in Ihrem **Atari** Computer. Diese Hardware Uhr ist eine Akkumulator gepufferte Quartzuhr. Somit ist sie System unabhängig und im Normalfall brauchen Sie sie nur einmal einzustellen.

Falls Sie Ihren **Atari** Computer einmal längere Zeit nicht benutzen, brauchen Sie auch keine Angst zu haben, daß die Uhr stehen bleibt. Der Akkumulator hält etwa sechs Monate ohne externe Stromzufuhr. Schalten Sie Ihren **Atari** Computer ein, wird dieser Akku neu aufgeladen.

Aber, was haben Sie nun von dieser Quartzuhr? Oder was können Sie mit dieser Quartzuhr anfangen?

Die Antworten sind ganz leicht. Zum ersten können Sie die Uhrzeit jederzeit in einer Statuszeile oberhalb des Sichtbaren Bildschirmes darstellen. Sie haben somit immer das Datum und die genaue Uhrzeit im Blick.

Zum zweiten können Sie die Daten der Uhr jederzeit in eigene Programme einbauen. Sie können also Programme schreiben, in denen Sie immer die genaue Uhrzeit oder das Datum verwenden.

Und zum dritten sind jetzt Echtzeit Steuerungen mit den **Atari** Computern der Serien **800XL** und **130XE** möglich.

Was sehen Sie nun, wenn die Uhr aktiviert ist?

Die Uhr meldet sich mit dem Wochentag, dem aktuellen Tagesdatum, dem Monat und der Jahreszahl. Rechts daneben finden Sie die genaue Uhrzeit mit der Angabe der Stunden, Minuten und Sekunden.

Wie aktivieren Sie diese Uhr?

Es gibt mehrere Wege die Statuszeile sichtbar zu machen. Beim Einsprung in den **Bibomon** wird sie automatisch eingeschaltet. Wollen Sie die Uhr vom **BASIC** oder einer anderen Programmiersprache, oder von einem Programm aus sichtbar machen, so brauchen Sie nur eine der **SHIFT** Tasten und **SYSTEM RESET** zu drücken. Schon ist die Statuszeile mit der Uhr zu sehen. Leider funktioniert das nicht bei allen Programmen einwandfrei. Wenn Sie die Statuszeile wieder löschen wollen, drücken Sie einfach noch einmal **SHIFT** und **SYSTEM RESET**.

Im Anhang dieses Handbuches werden Sie ein Beispiel Programm in **BASIC** finden, wo wir Ihnen zeigen, wie Sie die Daten der Uhr in ein eigenes Programm übernehmen können.

4.1 Einstellen der Uhr

Zum Einstellen der Uhr gehen Sie bitte in den **Bibomon**. Dort geben Sie bitte **US** gefolgt von **RETURN** ein. Sie sehen das folgende Bild:

```
REAL-TIME-CLOCK SETUP
-----
Mi 26.02.86 21:42:36
```

Sie sehen die gleichen Daten wie in der Statuszeile. Aber wie können Sie diese Daten nun verändern?

Dazu benutzen Sie den Cursorblock. Sie brauchen allerdings **nicht** die **CONTROL** Taste zu drücken.

Die **RECHTS** und die **LINKS** Tasten können Sie benutzen um zu den einzelnen Daten zu gelangen und die **AUF** und **RUNTER** Tasten um die Werte zu ändern.

Haben Sie die richtigen Werte eingestellt, drücken Sie den Taster zum einstellen der Uhr (diesen finden an der Rückseite Ihres Computers), halten diesen gedrückt und drücken **RETURN**.

Die geänderten Werte werden sofort in der Statuszeile angezeigt. Wie Sie sehen ist auch das einstellen der Uhr sehr leicht.

5.0 Die Hardcopy Routine

Jederzeit eine Hardcopy des gerade sichtbaren Bildschirms zu machen war schon immer der Traum vieler ATARI Computer Besitzer. Nun, mit dem 16K Bibomon ist es endlich möglich!

Sie können jederzeit die Hardcopy Routine des Bibomon aktivieren und den gerade sichtbaren Bildschirm über die eingebaute Druckerschnittstelle ausdrucken.

Das aktivieren dieser Hardcopy Routine ist denkbar einfach. Sie brauchen nur die CONTROL Taste und die HELP Taste drücken, und während Sie diese Tasten gedrückt halten, die SYSTEM RESET Taste zu drücken. Sofort wird der sichtbare Bildschirm ausgedruckt. Dabei spielt es keine Rolle, ob es ein Grafik 0 oder ein Grafik 8 Bildschirm ist. Auch gemischte Grafikstufen können so ausgedruckt werden.

Wenn Sie diese Hardcopy Routine bei einigen Spielen ausprobieren, werden Sie feststellen, das Sie statt der erwarteten Grafik einen Zeichensatz ausgedruckt bekommen. So wissen Sie dann auch gleich, wie dieses Bild programmiert wurde.

Diese Programme, deren Grafiken Sie nicht ausgedruckt bekommen, legen sie Startadresse der Displaylist nicht nach **\$0230-\$0231** ab. Somit hat der Bibomon keine Möglichkeit die Startadresse des Bildes zu finden.

Sie können von jedem Bildschirm eine Hardcopy in 4 verschiedenen Grössen und in 2 verschiedenen Höhen machen. Dazu kommt noch die Möglichkeit der Invertierung des Bildschirms.

5.1 Einstellen der Hardcopy Routine

Um diese Einstellungen vorzunehmen, gibt es drei Wege. Der erste Weg ist der über das Hauptmenu, das Sie sehen, wenn Sie Ihren Computer einschalten, und Sie haben weder BASIC noch ein anderes Steckmodul im Schacht, und Sie haben den Bibomon eingeschaltet. Von dort aus gelangen Sie über SELECT in das Hardcopy Setup Menu.

Vom Monitor aus können Sie über den [Befehl in das Hardcopy Setup Menu gelangen.

Und schliesslich drittens können Sie die Daten der Hardcopy Routine von einem eigenen Programm her ändern. Dieses erklären wir Ihnen im nachfolgenden Kapitel **10.2**.

Wenn Sie nun vom Monitor aus oder vom Hauptmenu her das Hardcopy Setup Menu aktivieren, sehen das folgende Menu:

HARDCOPY SETUP MENU

Density: -1- -2- -3- -4-

Inverse: 5-NO 6-YES

Height: 7-SGL 8-DBL

Printer: E-EPSON O-OKI

Lines: -- <0>

SELECT>

Mit Hilfe dieses Menu's können Sie alle Funktionen der Hardcopy Routine einstellen. Die eingestellten Werte werden invers dargestellt. Zum ändern der Werte drücken Sie einfach die Taste, die vor der gewünschten Option steht. Wollen Sie zum Beispiel die Density verändern, so drücken Sie die Tasten 1, 2, 3 oder 4. Die jeweilige Änderung wird sofort invers dargestellt. Besitzen Sie einen Okidata Microline Drucker so drücken Sie bitte die Taste **O**. Als letzten MENU Punkt finden Sie **LINES**. Lines gibt Ihnen die Anzahl der Grafik 0 Zeilen wieder, die bei einer Hardcopy ausgedruckt werden. Sie können also auch nur kurze Ausdrücke der ersten Bildschirmzeilen machen, z.b. für Etiketten. Sie können die Anzahl der auszudruckenden Bildschirmzeilen sehr leicht ändern. Sie brauchen dazu nur die Tasten < und > zu drücken. **0** hebt diese Sonderfunktion auf, es werden wieder alle Bildschirmzeilen ausgedruckt.

Die so einmal eingestellten Werte bleiben solange erhalten, bis Sie Ihren Computer ausschalten.

5.2 Einstellen der Hardcopy Routine durch Programm

Wie bereits erwähnt, können Sie alle Veränderungen im Hardcopy Setup Menu auch durch in Programm vornehmen. Aber diese Veränderungen können nur durch ein kurzes Maschinenprogramm erfolgen. Dazu hier eine Liste der entsprechenden Speicherstellen und Ihrer Funktionen.

Zuerst müssen Sie einen Wert in die Speicherstelle \$D621 schreiben. Sie können hier jeden Wert zwischen \$00 und \$FF benutzen. Wichtig ist nur, das Sie nicht vergessen den Wert in diese Speicherstelle zu schreiben, denn dadurch wird die Wertetabelle für die Hardcopy Routine aktiviert. Nachfolgend die Tabelle:

Speicherstelle-Wert-Funktion

Density:	\$D87C	-\$00	-Normale Breite, einfache Dichte
		-\$01	-Schmale Breite, einfache Dichte
		-\$02	-Normale Breite, doppelte Dichte
		-\$03	-Dreifache Breite, doppelte Dichte
Inverse:	\$D87B	-\$FF	-Invers ein
		-\$00	-Invers aus
Höhe:	\$D87D	-\$FF	-Doppelte Höhe
		-\$00	-Einfache Höhe
Drucker:	\$D87A	-\$FF	-Okidata
		-\$00	-Epson
LINES:	\$D879	-\$00	-Funktion aus
		-\$01-\$17	-Anzahl der Zeilen

Haben Sie diese Änderungen gemacht, vergessen Sie bitte nicht, den Rambereich mit der Wertetabelle wieder abzuschalten. Hier liegen die Mathematikroutinen die vom BASIC benutzt werden. Schreiben Sie hierzu einfach irgendeinen Wert in die Speicherstelle \$D620. Das ist dann alles.

6.0 Der Centronics Port

Der nun in Ihrem Computer eingebaute Centronics Port entspricht genau der Centronics Norm. Das heisst, Sie können jeden Drucker mit einer Centronics Parallel Schnittstelle an diesen Port anschliessen.

Weiterhin haben Sie aber auch die Möglichkeit, diesen Port als "USER" Port zu nutzen. Das heisst, Sie können diesen Port programmieren. Wie Sie das machen können, werden wir Ihnen auf den nächsten Seiten erklären.

Eine genaue Beschreibung des Port's finden Sie im Anhang dieses Handbuches. Hier einige Daten:

Wie bereits erwähnt, entspricht der Druckerport der Centronics Norm. Er ist aber auch baugleich mit dem Centronicsport des ATARI 850 Interface Moduls. Somit laufen alle Programme, die mit diesem Interface Modul arbeiten auch auf unserem Port. Das heisst aber auch, daß Sie bei Ihrem Drucker den Line Feed aktivieren müssen. Diese Funktion wandelt den vom Computer kommenden Carriage Return Befehl in einen Carriage Return und einen Line Feed Befehl am Drucker um.

Nachdem Sie Ihren Drucker so eingestellt haben, können Sie ihn mit der Drucker-schnittstelle des 16K Bibomon betreiben.

Die Daten werden im 8-Bit Datenformat an den Drucker übertragen. Der Druckerport hat 8 Datenausgänge (Datenleitungen 0-7), 1 Strobe Leitung (Mitteilung für den Drucker das Daten anstehen), und 1 Busy Leitung (Rückmeldung vom Drucker, wenn LOW dann fertig für Daten aufnahme).

Soweit die Daten für den Port.

6.1 Programmieren des Port's

Wie wir bereits erwähnt haben, können Sie den Port auch selbst programmieren, das heisst, Sie können Programme selber schreiben, die zum Beispiel einen Akustik Koppler oder einen Tischroboter über den Port steuern. Als Beispiel finden Sie im Anhang eine kleine Testschaltung und ein Demoprogramm für eine ganz einfache Testschaltung. Mit dieser Testschaltung können Sie die Datenübertragung am Port mit Hilfe von 8 Leuchtdioden verfolgen.

Die Adressen für den Port:

- \$D600 - 54784 - Port B - Datenregister
- \$D602 - 54786 - Port B - Richtungsregister

Beispiele:

- \$FF - Port auf Ausgang
- \$00 - Port auf Eingang
- \$01 - Datenleitung 0 auf Ausgang, alle anderen Leitungen auf Eingang
- \$02 - Datenleitung 1 auf Ausgang, alle anderen Leitungen auf Eingang
- \$03 - Datenleitung 0+1 auf Ausgang, alle anderen auf Eingang

Usw.

Die Adressen \$D601 (dez.54785) und \$D603 (dez.54787) werden normalerweise von der Uhr benutzt, mit Ausnahme des Bit 7. Über dieses Bit wird die Busy Leitung des Druckerport's abgefragt.

7.0 Anhang

In diesem Anhang finden Sie diverse Programme für die Steuerung des Port's, zur Abfrage der Uhr oder zum Einstellen der Hardcopy Routine durch ein kurzes BASIC Programm.

Die Programme erklären sich selbst, so daß wir hier nicht weiter auf sie eingehen wollen. Anhand dieser Beispiele sollten Sie dann auch in der Lage sein, selber gute Programme für die Nutzung der Uhr oder der Hardcopy Routine oder des Port's zu schreiben.

Wenn Sie ein solches Programm geschrieben haben, lassen Sie es uns wissen. Vielleicht werden wir es in einem geplanten ANWENDERHANDBUCH ZUM 16K BIBOMON veröffentlichen.

Mit Hilfe der Portdemo können Sie die Funktionsweise des Parallelports ganz leicht demonstrieren. Das Programm auf Seite 63 und die Schaltzeichnung auf Seite 67 gehören zusammen. Das eine ist die Software, das andere die Hardware für diese Demonstration.

Am Schluß dieses Handbuches möchten wir Ihnen noch einmal viel Spaß und natürlich auch viel Erfolg bei Ihrer Arbeit mit dem 16K BIBOMON wünschen. Und noch ein letzter Hinweis. Lassen Sie sich bei uns registrieren. So haben Sie dann immer die Möglichkeit, sobald eine Update Version des 16K BIBOMON's verfügbar ist, oder wenn wir neue Unterlagen über oder für den 16K BIBOMON heraus bringen, sofort darüber informiert zu werden. Achtung! Nur Käufer eines Originals können sich registrieren lassen!

Ihr Compy-Shop Team

UHRDEMO BASIC

```
10 REM *****
20 REM * DEMO PROGRAMM *
30 REM * 16K BIBOMON *
40 REM * (C) 1986 COMPY-SHOP *
50 REM *****
60 GOSUB 430
70 FOR I=1 TO 12
80 GRAPHICS 1:SETCOLOR 0,3,12
90 SETCOLOR 1,13,12:SETCOLOR 2,0,0
100 DIM TIS(40),CLS(10),WTS(70),MON$(100),AB(12),LE(12)
110 TIS=" ":TIS(40)=TIS:TIS(2)=TIS
120 WTS=" MONTAG DIENSTAG MITTWOCH DONNERSTAG "
125 WTS(LEN(WTS)+1)="FREITAG SAMSTAG SONNTAG "
130 FOR I=1 TO 12:READ CLS
140 AB(I)=LEN(MON$)+1
150 MON$(LEN(MON$)+1)=CLS
160 LE(I)=LEN(CLS)
170 NEXT I
180 POSITION 0,0:GOSUB 410
190 POSITION 0,1:? #6;" bibomon n k"
200 GOSUB 410
210 POSITION 0,16:GOSUB 410
220 POSITION 0,17:? #6;" echtzeituhr"
230 GOSUB 410
240 POKE 752,1
250 ? "K BASIC-Programm mit Maschinen-Unter-"
260 ? " routine zur Abfrage des Uhrenchips"
270 ? " RTC 58321 auf der BIBOMON 16k Pla-"
280 ? " tine. (C)Compy Shop 1986";
290 POKE 66,1
300 A=USR(1536,ADR(TIS))
310 POKE 66,0
320 CLS=TIS(32,39)
330 POSITION 6,6:? #6:CLS
340 WT=(PEEK(1001)-144)*10+1
350 POSITION 5,9:? #6;WTS(WT,WT+9)
360 MON=VAL(TIS(22,23))
370 POSITION 5-LE(MON)/2,12
380 ? #6;TIS(19,20);" ";
390 ? #6;MON$(AB(MON),AB(MON)+LE(MON)-1);" ";TIS(25,28)
400 GOTO 300
410 ? #6;"-----":RETURN
420 REM -----
430 FOR I=1536 TO 1594
440 READ A:POKE I,A
450 NEXT I:RETURN
460 DATA 104,104,133,204,104,133,203,32,41,6,160
465 DATA 39,185,192,3,41,127,133,205,10,10,42
470 DATA 42,41,3,170,189,37,6,69,205,145,203,136
475 DATA 16,232,96,32,96,64,0,104,24,105
480 DATA 1,170,104,105,0,72,138,72,72,72,8,76,183,214
490 DATA JANUAR,FEBRUAR,MAERZ,APRIL,MAI,JUNI
500 DATA JULI,AUGUST,SEPTEMBER,OKTOBER,NOVEMBER,DEZEMBER
```

PORTDEMO BASIC

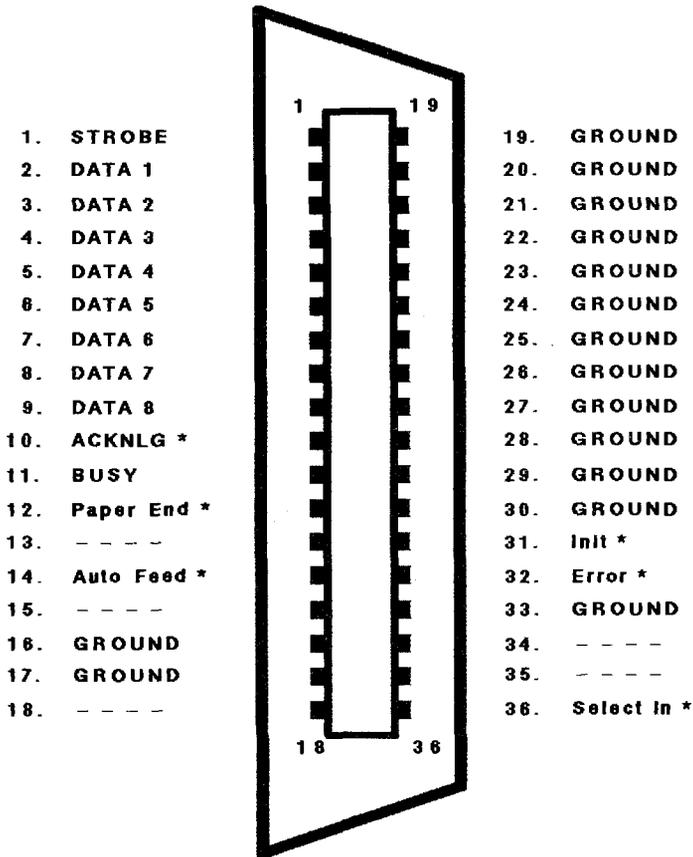
```
10 REM *****
20 REM * PORT - DEMO *
30 REM *****
40 PBD=54784:PBDD=54786
50 POKE PBDD,255:REM PORT ALS AUSGANG
60 POKE PBD,0:REM AUSGAENGE AUF LOW
70 DIM A(20)
80 RESTORE
90 TRAP 250
100 READ MAX
110 FOR N=1 TO MAX
120 READ AA
130 A(N)=AA
140 NEXT N
150 REM
160 FOR K=1 TO 5
170 FOR I=1 TO MAX
180 AA=A(I)
190 POKE PBD,AA
200 FOR TI=1 TO 30:NEXT TI
210 NEXT I
220 NEXT K
230 POKE PBD,0
240 GOTO 100
250 FOR I=0 TO 100
260 A=INT(256*RND(0))
270 POKE PBD,A
280 FOR TI=1 TO 20:NEXT TI
290 NEXT I
300 GOTO 80
310 REM
320 DATA 8,1,2,4,8,16,32,64,128
330 DATA 14,1,2,4,8,16,32,64,128
340 DATA 64,32,16,8,4,2
350 DATA 9,0,1,3,7,15,31,63,127,255
360 DATA 15,1,3,7,15,31,63,127,255
370 DATA 254,252,248,240,224,192,128
380 DATA 8,24,24,36,36,66,66,129,129
```

HARDCOPY DEMO

```

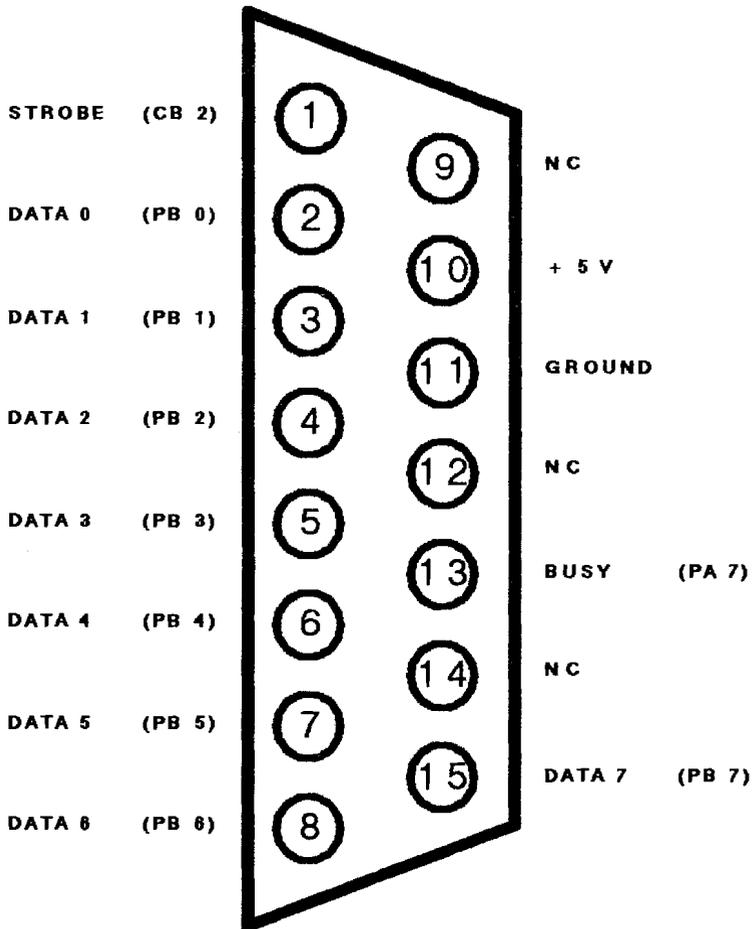
10 REM
20 REM DEMOPROGRAMM:
30 REM HARDCOPY VOM BASIC AKTIVIERT
40 REM
100 DIM SET$(20),HCS$(7)
110 REM MASCHINENPROGRAMM ZUM EINSTELLEN DER PARAMETER
120 SETS="hh,kl,bb,10,vfl,0"
130 REM MASCHINENPROGRAMM ZUM AKTIVIEREN DER BIBOMON-HARDCOPYROUTINE
140 HCS="h,CUL,@"
145 REM BILDSCHIRMAUSGABE
160 ? "K";:POKE 752,1
180 POSITION 2,0
190 ? "r";
200 FOR I=1 TO 3
210 ? "-----r";
220 NEXT I
230 ? "-----"
240 GOSUB 560
250 ? " | 320 K | BASIC | BIBOMON | OLD- | "
251 ? " | RAM | AUS | 16 K | RUNNER | "
260 GOSUB 560
270 ? "l";
271 FOR I=1 TO 3
272 ? "-----l";
273 NEXT I
274 ? "-----"
280 GOSUB 560
290 ? " | 64 K | BASIC | 2.ZEI- | XL | "
291 ? " | RAM | EIN | CHENSATZ | "
300 GOSUB 560
310 ? "L";
320 FOR I=1 TO 3
330 ? "-----L";
340 NEXT I
350 ? "-----"
360 GOSUB 530
370 IF PEEK(53279)>6 THEN 370
380 XX=PEEK(53279)
390 IF XX>6 THEN 370
400 IF XX=6 THEN GOSUB 440:GOTO 370
410 IF XX<5 THEN 370
420 POKE 752,0
430 ? "K↓Ende der Demo":END
440 REM HARDCOPY EINSTELLEN UND STARTEN
450 IF PEEK(53279)<>7 THEN 450
470 POSITION 2,14:?"QPRINTING...";
480 A=USR(ADR(SETS),55417,11):REM D879=11 / ANZAHL DER ZEILEN
490 A=USR(ADR(SETS),55419,0):REM D87B=0 / INVERSE AUS
495 A=USR(ADR(SETS),55418,0):REM D87A=0 / DRUCKERTYP: EPSON
500 A=USR(ADR(SETS),55420,1):REM D87C=1 / SCHMALER AUSDRUCK
510 A=USR(ADR(SETS),55421,0):REM D87D=0 / EINFACHE HOEHE
520 A=USR(ADR(HCS)):REM AKTIVIEREN DER HARDCOPYROUTINE
530 POSITION 2,14:?"SELECT=ENDE START=PRINT";
540 IF PEEK(53279)>6 THEN 540
550 RETURN
560 ? "|";
570 FOR I=0 TO 3
580 ? " |-----|";
590 NEXT I
600 ?
610 RETURN

```



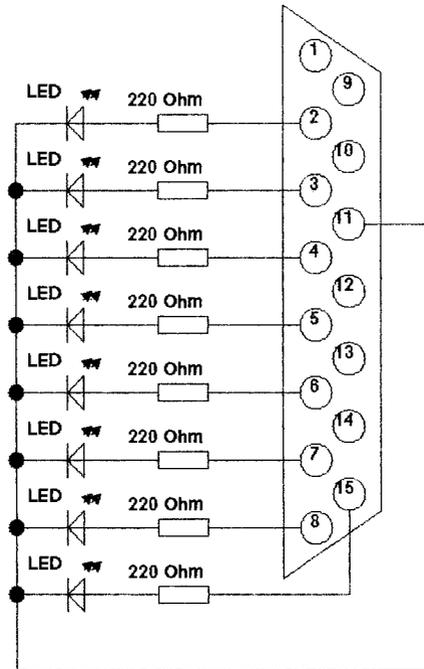
ANSICHT VON DER RUECKSEITE DES DRUCKERS

(*) Drucker-spezifische Anschlüsse werden nicht mit dem BIBOMON-Centronics-Port verbunden.



ANSICHT VON DER RUECKSEITE DES COMPUTERS

PORT - DEMO



Referenzkarte 16K BIBOMON

** BibOMON Befehlssatz **

G	-	Starte Programm (GOTO)	-	[addr]G
H	-	Speedy 1050 Steuerung	-	H
I	-	Disketten Index	-	I
J	-	Springe zur Adresse (JUMP)	-	[addr]J
K	-	SIO Kurzbefehl (E459G)	-	K
L	-	Disassembliere Speicher	-	[addr]L
M	-	Verschiebe Speicher (MOVE)	-	[nach]<[von].[bis]M
N	-	Ändere Registerinhalte	-	N
O	-	Öffne Bildschirm	-	O
P	-	Druckersteuerung	-	P1=ein, P=aus
Q	-	Ende, verlasse Monitor	-	Q
R	-	Lese Sektor(en) (READ)	-	[addr]<[n1].[n2]R,
S	-	Einzelschritt (SINGLESTEP)	-	[addr]S
T	-	Verfolge Programm (TRACE)	-	[addr]T
U	-	Uhr an-aus	-	U
US	-	Uhr einstellen	-	US
V	-	Vergleiche Speicher (VERIFY)	-	[addr]<[von].[bis]V
W	-	Schreibe Sektor(en) (WRITE)	-	[addr]<[n1].[n1]W
X	-	Exklusive Oder (EXOR)	-	[n1]X[n2]
Y	-	Zeilenassembler	-	Y
Z	-	Verfolge File	-	[n1]Z
ZL	-	Verfolge File mit Laden	-	[n1]ZL
ZR	-	Starte File	-	[n1]ZR
:	-	Schreibe in Adresse (STORE)	-	[addr]:[n1][n2][n3]
:	-	Schreibe in Adresse (STORE)	-	[addr];[n1][n2][n3]
.	-	Liste Speicherbereich (DUMP)	-	[addr].[addr]
+	-	Rechne Hex+Hex	-	[n1]+[n2]
-	-	Rechne Hex-Hex	-	[n1]-[n2]
&	-	OR	-	[n1]&[n2]
^	-	AND	-	[n1]^[n2]
,	-	Liste halbe Seite	-	[addr],
/	-	Letzte Disketten Operation	-	/
=	-	Fuelle Speicher	-	[n1]<[von].[bis]=
\$	-	Wandelt Hex in Dez	-	\$[n1]
#	-	Wandelt Dez in Hex	-	#[n1]
'	-	Bildschirmcode (ATASCII)	-	[addr]'[TEXT]
"	-	ASCII Code	-	[addr]"[TEXT]
>	-	Suche nach Hexcode	-	[von].[bis]>[HEX][HEX]
>"	-	Suche nach ASCIIcode	-	[von].[bis]>"[TEXT]
>'	-	Suche nach Bildschirmcode	-	[von].[bis]>'[TEXT]
!	-	System Adressen	-	!
?	-	Registeranzeige	-	?
@	-	Hauptmenu	-	@
\	-	Bankumschalter	-	\[n1]
I	-	Hardcopy Setup Menu	-	I

** BibOMON DOS-Befehle **

*D	-	Directory	-	*D [n] [NAME]
*I	-	Index	-	*I [n] [NAME]
*L	-	Load File (COM)	-	*L [n] [NAME]
*R	-	Run File	-	*R [n] [NAME]
*T	-	Trace File	-	*T [n] [NAME],L
*G	-	Get File (BIN)	-	*G [n] [NAME], [addr]
*V	-	View File	-	*V [n] [NAME]
*S	-	Save File (COM)	-	*S [n] [NAME], [von], [bis]
*W	-	Write File (BIN)	-	*W [n] [NAME], [von], [bis]
*P	-	Protect File	-	*P [n] [NAME]
*U	-	Unprotect File	-	*U [n] [NAME]
*E	-	Erase File	-	*E [n] [NAME]
*N	-	Rename File	-	*N [n] [ALT], [NEU]
*FS	-	Format Single Density	-	*F [n] S
*FD	-	Format Double Density	-	*F [n] D

**Beachten Sie bitte auch die Produkt -
Informationen auf den folgenden Seiten.**

Die Compy-Shop Speichererweiterungen

Wir liefern Ihnen für Ihren ATARI Computer der Serien 600XL - 800XL und 130XE Speichererweiterungen in unterschiedlichen Größen. So bekommen Sie für den ATARI 600XL eine Speichererweiterung auf 64K Ram und eine Speichererweiterung auf 192K Ram. Wichtig ist aber bei allen Speichererweiterungen die über 64K Ram Arbeitsspeicher hinaus gehen, daß diese Speichererweiterungen kompatibel sind zum ATARI 130XE. Das bedeutet, daß alle Programme, die für den ATARI 130XE geschrieben wurden, auch auf den erweiterten Computern der Serien 600XL (mit 192K Ram) und 800XL (mit 320K Ram) laufen. Was haben Sie aber nun von so einem grossen Arbeitsspeicher?

Die Antwort ist ganz leicht. Sie haben eine grosse Ramdisk, daß heisst Sie haben ein zweites Laufwerk im Computer. Dadurch ergeben sich mehrere Vorteile.

1. Wenn Sie mit einer Programmiersprache arbeiten, die einen Compiler benutzt, können Sie diesen Compiler mit der Ramdisk arbeiten lassen. Die Compilierungszeiten werden dadurch sehr kurz. Fast alle Programmiersprachen, die Sourcecode von der Diskette zum Compilieren oder Assemblieren, laden können, können dies auch von der Ramdisk. Die Ladegeschwindigkeiten von der Ramdisk sind kaum noch meßbar. So arbeitet zum Beispiel der Compiler des kyan pascal mit der Ramdisk. Oder der Mac/65 kann Sourcecode aus der Ramdisk heraus assemblieren.
2. Wenn Sie eine Datenbank benutzen, die mit einer Ramdisk arbeiten kann, können Sie die Vorteile der Ramdisk jetzt auch mit diesem Programm nutzen. Es gibt zum Beispiel eine 128K Version des SynFile+, das Ihnen etwa 96K Arbeitsplatz im Computer frei lässt. Oder das Syncalc in der 128K Version hat ein freies Arbeitsplatz von 96K Grösse.
3. Sie haben nun auch die Möglichkeit, das BASIC XE von OSS auf einem ATARI 600XL oder einem ATARI 800XL laufen zu lassen. Dieses Basic lässt Ihnen etwa 65K Arbeitsspeicher frei.
4. Und dann können Sie den zusätzlichen Speicherplatz natürlich auch für Ihre eigenen Programme nutzen. So können Sie den zusätzlichen Arbeitsspeicher für die Datenablage oder für Maschinenunterroutinen nutzen. Aber eine Sinnvolle Nutzung dieses zusätzlichen Rambereichs kann nur von der Maschinensprache her erfolgen. Basic Programmierern ist dieser Vorteil leider verschlossen.

Wie Sie sehen gibt es jede Menge Vorteile, wenn Sie sich eine unserer Speichererweiterungen in Ihren Computer einbauen, oder einbauen lassen. Hier noch eine Liste der Speichererweiterungen, die wir Ihnen für Ihren ATARI Computer liefern können:

ATARI 600XL von 16K Ram auf 64K Ram -- ohne Ramdisk

ATARI 600XL von 16K Ram auf 192K Ram -- mit 128K Ramdisk

ATARI 800XL von 64K Ram auf 320K Ram -- mit 256K Ramdisk

ATARI 130XE von 128K Ram auf 192K Ram -- mit 128K Ramdisk

Speedy 1050

(c)1986 Compy-Shop

Was ist *SPEEDY 1050*?

SPEEDY 1050 ist ein Hardware Zusatz für die Atari Floppy 1050. Dieser Hardware Zusatz wird in das Laufwerk eingesteckt. Es sind keine Lötarbeiten dazu erforderlich.

(Diese Angabe bezieht sich auf die Grundaustführung.)

SPEEDY 1050 gibt es in vier Ausführungen.

1. Die Grundaustführung.

In dieser Grundaustführung haben Sie bereits alle Vorteile dieses Hardware Zusatzes. Sie haben echte *DOUBLE DENSITY*, das heißt also 176 K Byte Speicherplatz und eine höhere Arbeitsgeschwindigkeit. Die Geschwindigkeit ist abhängig von der verwendeten Software. Ohne Software ist das umgebaute Laufwerk doppelt so schnell wie eine normale 1050. Verwenden Sie spezielle Software, so wird das Laufwerk ca. 800% schneller. Alleine das Formatieren einer Diskette kann mit einer *SPEEDY 1050* in ca. *8 Sekunden* erfolgen.

2. Die Ausbaustufe mit Trackanzeige und Summer

Diese Ausbaustufe entspricht der normalen *SPEEDY 1050*. Aber hier bekommen Sie noch einen Einbausatz für eine Digitale Trackanzeige und einen Summer. Diese Trackanzeige ist ein nützliches Instrument für alle, die viel mit dem Laufwerk arbeiten. So erhalten Sie nicht nur Informationen über den Bereich der Diskette wo Ihr Laufwerk gerade arbeitet, sondern auch Informationen über den Status der Sektoren oder des *SPEEDY 1050* Bordes. In dieser Ausbaustufe müssen Sie die Trackanzeige in der Aussenfront Ihres Laufwerkes befestigen. Dazu müssen Sie einige Umbauarbeiten an Ihrem Laufwerk vornehmen, oder von uns vornehmen lassen. Über den Summer werden Fehlermeldungen akustisch angezeigt. Wenn Sie zum Beispiel beim schreiben der Daten die Laufwerkklappe öffnen, und das Laufwerk noch Daten schreiben muss, so ertönen solange Pieptöne, bis Sie die Diskette wieder eingelegt haben und die Klappe geschlossen haben. Das Laufwerk schreibt dann an der gleichen Stelle weiter, an der Sie die Arbeit unterbrochen haben. Es gehen Ihnen *keine* Daten mehr verloren.

3. Die Ausbaustufe mit allen Technischen Unterlagen

Diese Ausbaustufe der *SPEEDY 1050* ist für alle Interessierten Programmierer gedacht, die sich mit der Programmierung der *SPEEDY 1050* befassen wollen. Denn mit diesem Hardware Zusatz können Sie Ihre eigenen Kopierprogramme schreiben. Oder ein eigenes DOS. Mitgeliefert werden bei diesem Entwicklungssystem alle Technischen Unterlagen und ein ausführliches ROM Listing der *SPEEDY 1050*. Wer jemals versucht hat eine Happy zu programmieren wird das zu schätzen wissen. Ausserdem stehen wir Ihnen mit Rat und Tat zur Seite, falls Sie trotz der guten Unterlagen noch Schwierigkeiten bei der Programmierung der *SPEEDY 1050* haben sollten.

4. Wir liefern Ihnen die folgenden Versionen:

SPEEDY 1050N, Normal Ausführung

SPEEDY 1050NE, Entwicklungssystem, mit allen Technischen Unterlagen

SPEEDY 1050T, mit Digitaler Trackanzeige

SPEEDY 1050TE, Entwicklungssystem, mit allen Technischen Unterlagen

5. Die Lieferzeit

Die ersten Versionen der *SPEEDY 1050* werden Anfang *JULI '86* ausgeliefert werden. Sichern Sie sich jetzt schon eines der ersten Exemplare durch eine Rechtzeitige Vorbestellung!

kyan pascal für die ATARI XL und XE Serie

kyan pascal ist ideal dazu geeignet, die Pascal Programmiersprache auf den ATARI Computern der XL und XE Serie zu lernen. Aber auch für den fortgeschrittenen Programmierer ist **kyan pascal** zur Entwicklung von leistungsfähigen Programmen geeignet. **kyan pascal** umfasst eine volle Implementierung des Jensen/Wirth Pascal und entspricht somit dem ISO Standard. Sehr schnell hat **kyan pascal** viele Freunde sowohl unter Schülern und Studenten, als auch unter Erfahrenen Pascal Programmieren gefunden. Die Anwenderfreundliche Benutzeroberfläche macht das lernen und programmieren in Pascal schneller und leichter. Und ab sofort ist das **kyan pascal** auch für ATARI lieferbar!

kyan pascal ist ein extrem kraftvolles Programmier Werkzeug, doch es ist auch für den Anfänger ideal geeignet. In den Kommandomenüs, den Help-Screens und dem Handbuch finden Sie alle Hilfe die Sie brauchen um Pascal lernen zu können. Zudem zeigen Ihnen die umfangreichen Compiler Fehlermeldungen jeden Syntax Fehler ganz genau an. Ein sehr nützliches Werkzeug für den Anfänger wie für den Fortgeschrittenen.

kyan pascal ist aber mehr als nur ein "Lempascal". Es ist ein kraftvolles Software Entwicklungssystem das dem Programmierer viele Fähigkeiten und Vorteile bietet, die Sie selbst bei grösseren Pascal Compilern nicht finden. Und es ist superschnell beim Compilieren und beim Programmauf. Der Pascalcode wird in Sekunden compiliert und die Programme laufen mit der maximalen Geschwindigkeit, die auf einem 6502 Computer möglich ist.

kyan pascal Vorzüge:

6502 Maschinensprache Compiler: der Compiler erzeugt reinen 6502 Binärcode. Programme, geschrieben in **kyan pascal** laufen bis zu 30 mal schneller als BASIC Programme.

Bildschirm orientierter Texteditor: Mit voller Cursor Kontrolle und speziellen Editier Kommandos wie suchen und austauschen.

Kraftvolle Erweiterungen: wie INCLUDE File, CHAIN, und DISPOSE, dazu noch das String Handling und RANDOM ACCESS FILES.

Volle Compiler Fehlermeldungen: Sie erhalten eine komplette Liste aller Syntaxfehler auf dem Bildschirm oder auf dem Drucker, wie Sie wollen.

Graphics, Sounds und Random Number Generator: Sie können alle Vorteile der ATARI Computer in Ihre Pascal Programme einbinden.

Eingebauter Assembler: erlaubt das einbinden von Assemblerlistings im Pascal Listing, entweder in-Line oder als Include- Datei .

Optionale Bildschirm/Drucker Ausgabe: des Compilierten Assembler Code's.

Handbuch: mit vielen Programmbeispielen und einer genauen Erklärung aller Pascal Befehle und Anweisungen. (In Englisch)

kyan pascal braucht lediglich ein einzelnes Laufwerk (ATARI 810 oder 1050 oder ein kompatibles) und läuft auf allen ATARI Computer der Serien XL und XE mit 64K Ram. Keine weitere Hardware wird benötigt!

