

AtariMax Flash-Cartridge Dokumentation (Deutsch/German)

14. Juli 2004

Geschrieben von

- Steve Tucker

Übersetzt von

- Matthias Lüdtkke
- Marc Brings
- Ralf Patschke
- Carsten Strotmann

Eine Publikation des

Atari Bit Byter User Club e. V.
c/o Wolfgang Burger
Wieschenbeck 45
D-45699 Herten

Germany

<http://www.abbuc.de/>

AtariMax Flash Cartridge



Inhaltsverzeichnis

1	Hardware/Cartridge Spezifikation	6
1.1	Physikalische Beschreibung	6
1.2	Kapazität	6
1.3	Programmierung	6
2	Software/Toolkit Spezifikation	7
2.1	Software-Übersicht	7
2.2	Hauptmerkmale des "multicard generator" Skripts (-EXEPACKER Option)	7
2.3	Hauptmerkmale des "disk-card generator" Skripts (-DISKPACKER Option)	8
2.4	Hauptmerkmale des "memory dump unpacker" Skripts (-MEMPACKER option)	9
2.5	Andere Eigenschaften des "build scripts"	9
3	Programmierung der Cartridge	10
3.1	Einleitung	10
3.2	Programmierung der Cartridge	10
3.2.1	Schritt 1: Laden des Flash-Programmier-Image	10
3.2.2	Schritt 2: Starten des Flash-Programmier-Images	10
3.3	Lösung von häufigen Programmierproblemen	12
4	Installieren der Maxflash Software	15
4.1	Voraussetzungen	15
4.1.1	Schritt 1: Die Installation von Perl	15
4.1.2	Schritt 2: Den Perl-Interpreter testen.	15
4.1.3	Schritt 3: Die Installation von TASM	16
4.1.4	Schritt 4: TASM testen	16
4.1.5	Schritt 5: Installation des Maxflash-Software-Paketes	17
4.1.6	Schritt 6: Konfiguration der Maxflash-Software.	17
4.1.7	Schritt 7: Test des Maxflash-Utility	18

INHALTSVERZEICHNIS

5	Benutzung der Maxflash Software	20
5.1	Einleitung	20
5.2	Schlüsseloptionen	20
5.3	Packen der EXE/COM Dateien	23
5.3.1	Eingabe der Menü-Texte	24
5.3.2	Eingabe des Cartridge Titels	24
5.3.3	Erstellen eines Disketten-Abbildes	24
5.4	Erstellen einer Disketten-Cartridge	25
5.4.1	Erstellen des Abbildes	25
5.4.2	Eingabe des Cartridge Titels	28
5.4.3	Erstellen eines Disketten-Abbildes	28
5.5	Erstellen eines Atari800 A8S Speicher-Abbildes	29
5.5.1	MEMPACKER	29
5.5.2	Erstellen eines Abbildes	30
5.6	Hilfsprogramm-Optionen	31
5.6.1	“-BIN2ALL”-Option	31
5.6.2	“-BIN2ATR”-Option	31
5.6.3	“-BIN2CAR”-Option	32
6	Benutzung der Flash Cartridge	34
6.1	Einleitung	34
6.2	-EXEPACKER Cartridges (Multi-cart images)	34
6.2.1	Systemvoraussetzungen	34
6.2.2	Das Modul umgehen (Cartridge Bypass)	35
6.2.3	Auswahl der Software (Software Selection)	35
6.3	Modul-Reset-Zeit	36
6.4	-DISKPACKER Images (Disk cartridges)	36
6.4.1	Das Modul umgehen (Cartridge Bypass)	36
6.4.2	Systemvoraussetzungen	36
6.4.3	Das Modul booten	36
6.4.4	Diskette umgehen (Disk Bypass)	37
6.4.5	Seiten-Reset (Side Reset)	37
6.4.6	Das Modul nach dem Booten abschalten (Post-boot Cartridge Disable)	37
6.5	-MEMPACKER (A8S Memory Unpacker Cartridge)	38

INHALTSVERZEICHNIS

6.5.1	Das Modul umgehen (Cartridge Bypass)	38
6.5.2	Systemvoraussetzungen	38
6.5.3	Das Modul booten (Booting the Cartridge)	38
6.6	Individuelle Cartridge Abbilder (Custom Cartridge Images)	38
7	Emulator Unterstützung	39
7.1	Einleitung	39
7.2	Änderungen am Atari800Win Plus 3.1	39
7.3	Herunterladen des Emulators	39
7.4	Starten des Emulators	40
7.5	Mehr zum Emulator	40
7.6	Testen eines -EXEPACKER (Multi-cart) oder -MEMPACKER (A8S) Flash Cartridge Image mit dem Emulator	40
7.6.1	Wahl der Moduldatei	40
7.6.2	Starten des Moduls	42
7.7	Testen eines -DISKPACKER (Disk-cart) Flash Cartridge Images mit dem Emulator	42
7.7.1	Einstellen der Optionen des Emulators	42
7.7.2	Wahl der Moduldatei	42
7.7.3	Starten des Moduls	43
8	Konvertierungsanleitung	45
8.1	Einleitung	45
8.2	Änderungen an Black Lamp	45
8.3	Änderungen an Zorro	50
8.4	Größere Änderungen an MULE	52

1 HARDWARE/CARTRIDGE SPEZIFIKATION

Im gesamten Text werden die Begriffe Modul oder Cartridge synonym verwendet, es wird immer die AtariMax Flash-Cartridge angesprochen.

1 Hardware/Cartridge Spezifikation

1.1 Physikalische Beschreibung

- Professional gefertigte Leiterplatte (Hergestellt in den USA)
- Neu angefertigtes schwarzes Modulgehäuse
- Farbiges Etikett
- Vollständig gesockelte Platine für den leichten Austausch von defekten Bauteilen
- Keine Batterien oder andere, schnell verschleißenden Komponenten

1.2 Kapazität

- 1MBit Flashspeicher angeordnet in 16 Bänken a 8kb (1Mbit Modul \Leftrightarrow 128kByte)
- 8Mbit Flashspeicher angeordnet in 128 Bänken a 8kb (8Mbit Modul \Leftrightarrow 1MByte)
- Maximum Kapazität von 1MB (pro Modul), davon durchschnittlich 5% Überschuss für das Dateisystem.

1.3 Programmierung

- Für die Programmierung wird eine spezielle, mit dem Toolkit gelieferte Software verwendet
- Die Dauer der Programmierung eines 1Mbit Moduls beträgt ca. 1-2 Minuten, abhängig vom Betriebssystem
- Minimum Lebenszeit des Flashspeichers: 100.000 Programmier-/Löschzyklen
- Lebenszeit der Daten ohne Stromzufuhr: 30 Jahre
- GNU Lizenz Softwarekit für die freie Verbreitung von Cartridge-Abbildern und zur Entwicklung neuer auf dem Modul basierender Software

2 Software/Toolkit Spezifikation

(Übersetzt von Matthias Lüdtko)

Information auf dem Stand des Betatester-Toolkit vom 07/10/2003.

Dies ist nur eine Übersicht die vorhandenen Merkmale, die "Readme" Datei des Toolkit beinhaltet eine vollständige funktionelle Beschreibung aller Möglichkeiten. Nur die Merkmale für derzeit vorhandene Hardware werden hier beschrieben. Ab dem 07/10/2003 sind das nur Merkmale für das 1 Mbit Modul. Obwohl das Modul selbst nicht Rechner- oder Betriebssystem-Spezifisch ist und jede Art von Programm oder Daten speichern kann, wird ein Open-Source-Toolkit mitgeliefert um Funktionen für Benutzer bereitzustellen, die keine eigene Software schreiben wollen.

2.1 Software-Übersicht

- Open-Source Design unter Verwendung von Perl-Skripten und TASM 6502 Assembler
- Vollständig Windows/Linux kompatibel
- Atari800-Emulator Unterstützung von Modulimages für einfache Fehlersuche und Tests
- Angepasste Atari800Win Unterstützung für die Fehlersuche in Cartridge-Abbilddateien
- Vollautomatische Generierung von direkt in das Flash-ROM schreibbaren Cartridge-Abbilddateien und Atari800 Emulator "CART"-Abbilddateien.

2.2 Hauptmerkmale des "multicard generator" Skripts (-EXEPACKER Option)

- Packt eine beliebige Anzahl von Atari-EXE-Files in den vorhandenen Flashspeicher, zum Erstellen einer Multicard
- Fähigkeit zur automatisches Einbindung eines Atari 800 Translators (Atari 800 OS-B)
- Option, den Modulstart beim Einschalten des Rechners zu umgehen, um eine leichte Neuprogrammierung der Cartridge zu ermöglichen
- Voll kompatibel zum Jellystone Park-EXE-Archive (1000+ Spiele wurden bisher auf Kompatibilität überprüft)
- Automatische Generierung eigener Cartridge-Abbilddateien, ohne Anpassung des Quellcodes
- Kompatibel mit allen Atari 8Bit-Rechnern (benötigt kein XL/XE-OS)
- Automatische EXE Packer- und Reparaturfunktion
- Tastatur- und Joystick-Programmauswahl im Modulmenü

2 SOFTWARE/TOOLKIT SPEZIFIKATION

- Option, unerwünschte Menümerkmale aus dem Modulmenü zu entfernen, die Programmanwahl zu Beschleunigen uvm.
- Platzsparende EXE-Unpacker Routine (weniger als 200 Bytes), frei im RAM verschiebbar für hohe Kompatibilität
- Der Unpacker kann an jede Stelle des Speichers entpacken, auch unterhalb des OS-Roms
- Volle Unterstützung der 8Mbit Cartridge, verifiziert von der 8Mbit Testerguppe (07/20/2003)
- Zeitaufwand zum Erstellen und Testen eines Cartridge-Abbildes mit EXE-Dateien auf dem Emulator: 5 Sekunden @ 500 Mhz i386 Intel PC.

2.3 Hauptmerkmale des "disk-card generator" Skripts (-DISKPACKER Option)

- Fähigkeit jedes Single-Density ATR-Disketten-Abbild in die Cartridge zu schreiben
- Beschränkte Fähigkeit, einige 2-seitigen Single-Density ATR-Disketten-Abbilder in ein 1 Mbit Cartridge zu packen, zum Beispiel die meisten Infocom-Spiele
- Möglichkeit zur Einbindung eines Atari 800 Translators (Atari 800 OS-B)
- Option zum Erstellen von Cartridge-Abbildern, die Atari-Basic benötigen (nur XL/XE)
- Option, den Modulstarts beim Einschalten zu umgehen um eine einfache Neuprogrammierung des Moduls zu ermöglichen
- Einfache Bedienung, nur Modul einstecken und die "Game-Disk" startet mit den eingestellten Optionen
- Unterstützung von Spielen/Utilities, die auf die Diskette zugreifen, zum Beispiel Infocom-Spiele
- Möglichkeit, nahezu alle Arten von Programmen in eine Cartridge zu speichern, z.B. ist es leicht ein DOS-Modul mit Hilfsprogrammen zu erstellen
- Vollständiger Zugriff auf das Diskettenlaufwerk auch bei eingesteckter Cartridge, somit ist das Speichern von Spielständen weiterhin ermöglicht
- Im Moment (07/10/2003) gibt es ca. 200 Spiele auf vorgefertigten Modulimages, einschließlich der meisten Infocom Spiele
- Automatisch generierte "Disk-Module" benötigen einen XL/XE Computer, manuell erstellte "Disk-Module" benötigen keine besonderen Voraussetzungen
- Automatischen oder manuelles wenden von 2 Diskettenseiten
- "Multi-Disk-Compacter" mit Menü wird derzeit von 8Mbit Testgruppe getestet
- Der Zeitaufwand für die Generierung einer Disketten-Cartridge von einem ATR File und Test im Emulator beträgt ca. 3 Sekunden auf einem 500 MHz i386 Intel PC.

2.4 Hauptmerkmale des "memory dump unpacker" Skripts (-MEMPACKER option)

- Fähigkeit Atari800-Emulator Speicherabbilder (A8S state saves) zu nutzen
- Der Speicher unterhalb des XL/XE Betriebssystems kann genutzt werden
- Diese Option wurde als Umsetzungskonzept programmiert, sie ist nützlich für einige schwer zu packende Spiele die nicht von Diskette laufen
- Diese Funktion funktioniert, wurde aber in nur einen Tag entwickelt. File- und Disketten-Pack Funktionen decken schon die meisten Programme ab

2.5 Andere Eigenschaften des "build scripts"

- Fähigkeit, ausführbare Atari Programme zu überprüfen und ggf. zu reparieren
- Konvertiert ein beliebiges Binär-Abbild in ein Atari800 Modul oder in eine flaschbare Diskette (erlaubt es jeden beliebigen Code in das Modul zu schreiben)
- In der Software Dokumentation befindet sich eine vollständige Liste aller Funktionen

3 Programmierung der Cartridge

(Übersetzt von Marc Brings)

3.1 Einleitung

Dieses Dokument setzt voraus daß bereits ein lauffähiges Flash-Cartridge-Programmier-Abbild erzeugt wurde, entweder im ATR-Format oder auf einer Diskette. Die Anleitung zum Erstellen eines eigenen Flash-Cartridge-Programmier-Image befindet sich in Kapitel 5: Anwendung der Maxflash-Software (Seite 20).

3.2 Programmierung der Cartridge

3.2.1 Schritt 1: Laden des Flash-Programmier-Image

APE/ SIO2PC-User Laden Sie das Flash-Programmier-Disk-Image in das virtuelle Laufwerk #1.

Diskettenlaufwerks-Benutzer Legen Sie Flash-Programmier-Disk in eine Atari 1050 oder kompatible Diskettenstation, welche als Laufwerk #1 eingestellt ist.

3.2.2 Schritt 2: Starten des Flash-Programmier-Images

Schalte Sie den ATARI-Rechner aus und stecken Sie das zu programmierende Flash-Modul ein. Halten Sie die OPTION-Taste gedrückt um den Inhalt der Flash-Cartridge zu umgehen und schalten Sie den ATARI ein, um die Flash-Programmier-Disk zu booten. Wenn die Flash-Programmier-Disk gebootet hat, sollten Sie mit einem Bildschirm ähnlich dem in Abbildung 1 gezeigtem begrüßt werden:

Folgen Sie den Anweisungen auf dem Bildschirm um das Flash-Modul zu löschen und neu zu programmieren. Den Namen, "Zork: The Great Underground Empire", den das Programm anzeigt, ist der Titel welcher beim Erzeugen des Flash-Programmier-Abbildes eingegeben wurde. Er beschreibt die Software die auf das Modul aufgespielt wird.

Sie werden aufgefordert zu bestätigen das der Inhalt der Cartridge gelöscht wird. Tritt kein Fehler auf, so läuft der vollständige Prozess, einmal gestartet, automatisch ab. Das Löschen und Programmieren der Cartridge geht schnell, aber es hängt stark davon ab wie schnell der ATARI die auf dem Modul zu plazierenden Daten von der Diskette oder vom PC laden kann.

Die typische Zeit zur Neuprogrammieren eines 1Mbit-Modules ist 1-2 Minuten mit Standard-SIO-Geschwindigkeit und APE/SIO2PC, deutlich weniger wenn ein High-speed-OS wie APE Warp+ OS benutzt wird, etwas länger wenn mit einer echten Diskette gearbeitet wird.

Wenn die Cartridge vollständig programmiert ist, schalten Sie den ATARI für mindestens fünf Sekunden aus und schalten Sie den Rechner dann wieder ein, um die Cartridge zu testen. Wurde das Flash-Cartridge-Abbild mit der "-diskpacker" (disk cart)

3 PROGRAMMIERUNG DER CARTRIDGE

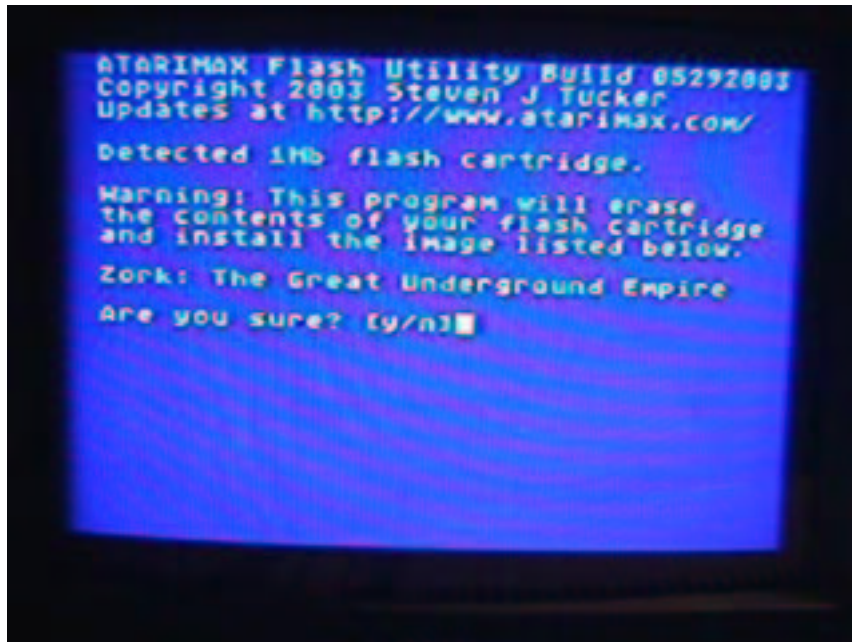


Abbildung 1: Programmieren der Cartridge



Abbildung 2: Programmierung der Flashcart abgeschlossen

3 PROGRAMMIERUNG DER CARTRIDGE

oder “-exepacker” (multicart) Option des Flash-Cart Software Paketes erstellt, werden alle benötigten Anpassungen beim Einschalten des Computers vorgenommen.

Es ist zum Beispiel nicht nötig, BASIC beim Booten eines XL/XE an- oder abzuschalten, die Cartridge-Software übernimmt diese Aufgabe.

3.3 Lösung von häufigen Programmierproblemen

Problem: "Ich habe die Flash-Cartridge in den Computer gesteckt, [OPTION] festgehalten, den Rechner eingeschaltet, aber der Bildschirm bleibt schwarz, oder die Cartridge-Software startet irgendwo. Was soll ich tun?"

Lösung: Die zum Flash-Modul gehörenden Hilfsprogramme zum automatischen Erzeugen von Multi-Carts und Disk-Carts plazieren eine Umgehungs-Funktion (Bypass) in der Cartridge, welche es erlaubt die Cartridge durch Festhalten der [OPTION]-Taste beim Booten zu ignorieren. Wenn die Cartridge mit einer Software programmiert wurde, die diesen speziellen Programmteil nicht beinhaltet, oder die Software wurde bei einem fehlgeschlagenem Programmierversuch beschädigt, wird der ATARI die Cartridge beim Starten nicht abschalten.

Um dieses mögliche Problem zu umgehen wurde die Flash-Cartridge-Hardware und -Software so konstruiert, daß die Cartridge mit der geringen Wahrscheinlichkeit eines Absturzes der Software bei eingeschaltetem Computer eingesteckt werden kann.

Folgen Sie der obigen Programmierprozedur, aber stecken Sie die Flash-Cartridge noch NICHT ein. Wenn die Flash-Programmier-Software startet, wird sie melden das keine Flash-Cartridge gefunden wurde (Siehe Abbildung 3).

3 PROGRAMMIERUNG DER CARTRIDGE



Abbildung 3: Keine Cartridge gefunden

Wenn dieses Bild auftaucht stecken Sie nun die Flash-Cartridge normal in den Computer. Drücken Sie die [Y]-Taste auf dem ATARI um den Erkennungsprozess neu zu starten. Wenn sich der Rechner während des Einsteckes des Modules aufhängt, drücken Sie RESET. In den meisten Fällen wird der Computer neu booten, das Flash-Modul ignorieren und die Programmierung kann fortfahren.

In der Regel dürfte das mehrmalige Wiederholen der obenstehenden Prozedur erfolgreich sein. Nachdem Sie Erfahrung mit der Programmierung der Flash-Cartridge gewonnen haben werden Sie lernen das Modul so einzustecken das der Computer nicht abstürzt.

Trotz der sorgfältigen Entwicklung der Hard- und Software um das Ein- und Ausstecken bei eingeschaltetem Computer zu ermöglichen, funktioniert die auf manchen Computer besser als auf anderen. Dies ist normalerweise darauf zurückzuführen, das die Modulschacht-Kontakte wenig (oder sehr) verschmutzt sind, oder auf die Gehäuseform, welche bestimmt wie schnell das Modul eingesteckt werden kann, wie z.B. beim 400, 800 und 1200XL.

Einige andere Tricks, die das Einstecken des Modules vereinfachen können, in keiner besonderen Reihenfolge:

- Reinigen Sie die Kontakte des Moduschachtes
- Neigen Sie das Modul beim Einstecken ein wenig nach rechts
- Booten Sie den Computer mit dem Omnimon OS (wenn vorhanden), starte den Debugger, und drücke [B] um von der Diskette zu booten

3 PROGRAMMIERUNG DER CARTRIDGE

- Wenn Sie eine Warp+ OS 32-in-1-Erweiterung benutzen, halten Sie [SELECT] gedrückt um in das OS-Auswahlmenü zu gelangen und wählen Sie ein OS zum Rebooten und Umgehen des Modules
- Wenn alles nichts hilft, nicht einmal die Tricks für den schlimmsten Fall, können Sie für weitere Unterstützung (auf englisch) Kontakt per E-Mail zu mir aufnehmen: "classics@atarimax.com".

Problem: "Wenn das Programmieren startet, schlägt das Löschen fehl."

Lösung: Versuchen Sie die Cartridge mehrmals zu löschen. Wenn sich die Cartridge nach mehrmaligen Versuchen nicht löschen lässt, aber erfolgreich erkannt wurde, versuchen Sie die Cartridge in einem anderem Computer zu programmieren.

Wenn die Programmierung auch auf einem anderen Computer fehlschlägt, sende bitte eine E-Mail mit Angaben zum Modell und TV-Standard des ATARI-Computers an "classics@atarimax.com".

Mögliche Gründe für ein Problem beim Löschen sind:

- Verschmutzte oder verbogene Modulschachtkontakte
- Verschmutzte Anschlüsse an der Cartridge (Die Kontakte mit einem nicht-schmirgelnden Kunstradierer reinigen)
- Ein Fehler im Flash-Speicher (Die Chips haben eine Lebensdauer von 100.000 Lösch-/Schreibzyklen)
- Ein Hardwarefehler (Die Cartridge kann kostenfrei ausgetauscht werden)
- Ein Fehler in der Software, der bisher unerkannt ist.

Problem: "Das Modul lässt sich programmieren, aber ich erhalte nicht das gewünschte Ergebniss"

Lösung: Wenn ein vorgefertigtes Cartridge-Abbild benutzt wird, bitte mit dem/der Ersteller/in des Abbildes klären ob spezielle Anforderungen an die Hardware oder den Speicher, ect. gestellt werden. Ein Flash-Cartridge-Abbild kann Software aller Art enthalten, es kann sein das versucht wird, ein Programm laufen zu lassen welches für eine ganz spezielle Hardware gedacht ist.

Durch fehlerhafte Downloads kann es zu beschädigten Flash-Abbildern kommen, oder die Quelldate war fehlerhaft.

Wenn das Flash-Abbild selbst erstellt wurde, konnte das Abbild erfolgreich im Atari800 Emulator getestet werden?

Überprüfen Sie ob die Einstellungen des Emulators den Standard entsprechen und mit der Hardware, auf der das Modul laufen soll, übereinstimmen.

Wenn mit dem "automatic conversion tool" ein Cartridge-Abbild erstellt wurde, welches im Emulator funktioniert, aber auf der realen Hardware versagt, senden Sie bitte eine E-Mail mit näheren Details an "classics@atarimax.com".

4 Installieren der Maxflash Software

(Übersetzt von Marc Brings)

4.1 Voraussetzungen

Das aktuelle Maxflash-Softwarepaket besteht aus einer Anzahl von Perl-Scripts und 6502-Assembler-Dateien. Um die Maxflash-Software zu benutzen wird ein funktions-tüchtiger Perlinterpretierer und der den TASM Crossassembler benötigt.

Es wird empfohlen mindestens Perl 5.x.x und TASM 3.0.1 zu verwenden.

4.1.1 Schritt 1: Die Installation von Perl

Wenn bereits eine lauffähige Installation von Perl 5 vorhanden ist, bitte bei Schritt 3 weiterlesen.

Perl unter Windows installieren Perl ist eine freie, mächtige Script-Sprache unter der GNU Public License (GPL). Es gibt viele Wege Perl zu beziehen und zu installieren.

Der bei weitem einfachste Weg, Perl unter Windows zu installieren, ist der Download und die Installation des Perl-Binärpaketes von ActiveState¹, einer Firma die Perl-Werkzeuge und Unterstützung anbietet.

Der Download von Perl ist kostenlos, es werden keine zusätzlichen kostenpflichtigen Tools benötigt um Perl zu benutzen.

Laden Sie das ActivePerl MSI-Paket von der Download Seite. Nach dem Download öffnen Sie das MSI-Paket mit dem Microsoft Windows Installationsprogramm.

Perl unter Linux installieren Bei den meisten Linux-Distributionen ist Perl bereits installiert oder es kann direkt benutzt werden.

Es gibt sehr viele unterschiedliche GNU/Linux-Distributionen, so daß eine Anleitung zur Installation von Perl unter Linux in diesem Dokument nicht beschrieben werden kann. Schauen Sie in das Nachbuch Ihrer Linux-Installation.

Weitere Informationen zur Installation von Perl finden Sie unter <http://www.perl.com/>, <http://www.perl.org/> oder unter <http://www.cpan.org/ports/index.html>.

4.1.2 Schritt 2: Den Perl-Interpreter testen.

Zum Test, ob der Perl-Interpreter korrekt installiert ist und funktioniert, öffnen Sie die Eingabeaufforderung (unter Windows fälschlicherweise auch MS-DOS Fenster genannt) oder ein Terminalfenster und geben Sie das Commando "perl -V" ein.

Die Ausgabe sollte ähnlich der folgenden erscheinen:

¹<http://www.activestate.com/>

4 INSTALLIEREN DER MAXFLASH SOFTWARE

```
C:\>perl -v
This is perl, v5.8.0 built for cygwin-multi-64int
Copyright 1987-2002, Larry Wall
Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.
Complete documentation for Perl, including FAQ lists, should be found on
this system using 'man perl' or 'perldoc perl'. If you have access to the
Internet, point your browser at http://www.perl.com/, the Perl Home Page.
```

Wir eine Fehlermeldung ausgegeben, so wiederholen Sie bitte die Installation. Überprüfen Sie unter Windows ob der Pfad zum Perl-Interpreter (perl.exe) in den Windows Pfad-Variablen eingetragen ist.

4.1.3 Schritt 3: Die Installation von TASM

Wenn bereits eine lauffähige TASM 3.x.x Installation vorhanden ist, lesen Sie bitte ab Schritt 5 weiter.

TASM ist ein tabellenbasierender Crossassembler, welcher vom Maxflash-Script benutzt wird um den Quelltext der Maxflash-Software bei der Erzeugung eines jeden neuen Modul-Images zu übersetzen.

TASM ist als Shareware für DOS und Windows erhältlich.²

Das TASM Archiv kann unter der URL <http://www.atarimax.com/flashcart/download/tasm32.zip> bezogen werden.

Entpacken Sie das TASM Archiv in ein neues Verzeichniss auf der Festplatte. Die Software hat kein eigenes Installationsprogramm.

Notieren Sie das Verzeichnis, unter dem Sie TASM installieren, zum Beispiel "C:\TASM" oder "/home/user/tasm"; Sie benötigt die Information später bei der Konfiguration des Maxflash-Build-Utilities.

4.1.4 Schritt 4: TASM testen

Öffnen Sie eine Eingabeaufforderung oder ein Terminalfenster und wechseln Sie in das Verzeichnis, in dem TASM installiert ist.

Starten Sie das Kommando "tasm" ohne Parameter. Wenn TASM korrekt installiert ist und funktioniert solltest eine Meldung ähnlich der Folgenden ausgegeben werden:

```
E:\tasm32>tasm
TASM Assembler. Version 3.2 September, 2001.
Copyright (C) 2001 Squak Valley Software
tasm: No files specified.
tasm -[nn] [-options] src_file [obj_file [lst_file [exp_file [sym_file]]]]
Option Flags defined as follows:
-[nn] Table (48=8048 65=6502 51=8051 85=8085 80=z80)
(68=6800 05=6805 70=TMS7000 96=8096)
(3210=TMS32010 3225=TMS32025)
-t[tab] Table (alternate form of above)
-a Assembly control (strict error checking)
-b Produce object in binary format
-c Object file written as a contiguous block
-d[macro] Define macro
-e Show source lines with macros expanded
-f[xx] Fill entire memory space with 'xx' (hex)
-g[x] Obj format (0=Intel,1=MOSTech,2=Motorola,3=bin,4=IntelWord)
-h Produce hex table of the assembled code
-i Ignore case in labels
-l[al] Produce a label table in the listing[l=long,a=all]
-m Produce object in MOS Technology format
-o[xx] Define number of bytes per obj record = [xx]
-p[lines] Page the listing file
-q Quiet, disable the listing file
-s Write a symbol table file
-x[xx] Enable extended instruction set (if any)
E:\tasm32>
```

²Alternativ gibt es im Internet auch Anpassungen der MaxFlash Software, um Open Source Assembler wie XA65 oder ca65 benutzen zu können.

4 INSTALLIEREN DER MAXFLASH SOFTWARE

Werden Fehlermeldungen ausgegeben, prüfen Sie bitte die Installation oder wiederholen Sie die Installation.

4.1.5 Schritt 5: Installation des Maxflash-Software-Paketes

Das gesammte Maxflash-Software-Paket, mit der zeitweisen Ausnahme einer Datei, wird als freie Software unter Version 2 der GNU Public License vertrieben.

Eine Kopie der GNU Public License liegt der Software bei und kann ebenfalls auf der gnu.org-Homepage³ gelesen werden.

Die Datei "flash.com", die sich in dem Maxflash-Download befindet, ist separat für nicht-kommerzielle Nutzung lizenziert, bis der Quellcode dieses Programmes veröffentlicht wird. Wenn das Paket unter den Bedingungen der GPL weitergegeben wird, muss das "flash.com"-Utility Programm entfernt werden.

Das aktuelle Maxflash-Utility und der Quellcode kann unter der URL http://www.atarimax.com/flashcart/download/maxflash_10152003.zip geladen werden.

Entpacken Sie die Datei in ein Verzeichniss Ihrer Wahl. Nach der Installation sollten dort die folgenden Dateien vorhanden sein:

```
07/10/2003 03:55 PM [DIR] flash
05/17/2003 02:31 PM [DIR] osb
07/10/2003 03:30 PM 26,107 exeptions.a65
07/10/2003 03:30 PM 7,158 exeunpak.a65
07/10/2003 03:14 PM 19,966 maxdisk.a65
07/10/2003 03:14 PM 6,236 mempack.a65
07/10/2003 03:14 PM 7,562 softdisk.a65
07/14/2003 03:04 PM 37,650 maxflash.pl
07/10/2003 07:25 PM 836 maxflash.cfg
06/29/2003 04:39 PM 18,349 license.txt
```

Die Größe der Dateien und das Datum können variieren.

4.1.6 Schritt 6: Konfiguration der Maxflash-Software.

Im Gegensatz zu Perl trägt sich der TASM Crossassembler nicht selbst in den System-Pfad ein, deswegen muß dem Maxflash-Utility mitgeteilt werden, wo es das Programm finden kann.

In dem Verzeichniss, indem die Maxflash-Software installiert wurde, befindet sich eine Datei mit dem Namen "maxflash.cfg". Öffne diese mit einem beliebigen Texteditor.

So sollte die Datei aussehen:

³<http://www.gnu.org/>

4 INSTALLIEREN DER MAXFLASH SOFTWARE

```
##
## MAXFLASH TASM CONFIGURATION FILE
## Names and values in this file must be seperated by a colon and must not span more than one line.
##
##
## Define the location of the TASM assembler (version 3.2 recommended) here.
##
TASM LOCATION : E:\tasm32\tasm.exe
TASMTABS DIRECTORY : E:\tasm32
##
## Changing this will break things.
##
TASM OPTIONS : -65 -b -f00
##
## If your using a version of TASM that returns useless exit codes, set this to ON to ignore them.
##
IGNORE EXIT CODE : off
```

Ändere den Eintrag in der Zeile "TASM LOCATION" in das Verzeichnis, in dem sich die Datei TASM . EXE auf dem System befindet.

Ändern Sie den Eintrag in der Zeile "TASMTABS DIRECTORY" in den Verzeichnisnamen, in dem sich die TASM Tabellendateien (Table Files) befinden. Diese Dateien befinden sich normalerweise in dem selben Verzeichnis wie TASM, könnten aber auch verschoben worden sein.

Wenn unter Windows Perl aus dem Cygwin Paket⁴ anstelle von ActiveState Perl benutzt wird, kann es sein, daß die Dateierweiterung ".EXE" vom TASM. EXE-Dateinamen entfernt werden muß, um TASM korrekt aufzurufen. Normalerweise ist dies aber nicht nötig.

4.1.7 Schritt 7: Test des Maxflash-Utility

Jetzt, nachdem alle Software installiert ist, kannst die Installation überprüft werden.

Um die TASM-Konfiguration zu testen, führen Sie das "maxflash.pl" Perl-Script aus, indem Sie in der Eingabeaufforderung, im TASM-Verzeichnis, das Kommando "perl maxflash.pl" eingeben.

Hierdurch wird der Perl Interpreter gestartet und das "maxflash.pl"-Script abgearbeitet. Wenn das Script mit den korrekten Pfadangaben zu TASM eingerichtet wurde, sollte eine Liste mit Programm-Optionen erscheinen.

Sind Angaben falsch, wird das Script mitteilen welche Angaben zu korrigieren sind.

Wenn alles stimmt, sollte eine Liste mit den Optionen des Scripts, ähnlich der nachfolgend abgebildeten, erscheinen:

⁴<http://www.cygwin.org/>

4 INSTALLIEREN DER MAXFLASH SOFTWARE

```
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
usage: perl maxflash.pl [switches] [hacks] [disk.atr] [cartridge.bin]
[dump.a8s]
switches:
-4MB (*) Create image for 4Mb cartridge. (default is 1Mb)
-8MB (*) Create image for 8Mb cartridge. (default is 1Mb)
-ALTPTR (E) Use alternate address for Atari EXE loader.
-BASIC (D) Enable BASIC. (For packing disks that use BASIC)
-BATCH (*) Never prompt for optional information.
-BIN2ALL (U) Make existing BIN file into CART/ATR images.
-BIN2ATR (U) Make existing BIN file into flash programming image.
-BIN2CART (U) Make existing BIN file into Atari800 CART image.
-CART (*) Write file with Atari800 cartridge header.
-CHEESE (E) Enable DLIs, sounds and colors in EXE menu.
-CLEAN (U) Remove intermediate files. (lst,bak,atr,com,bin,car)
-COLORS (*) Fill color registers while working. (debug)
-DISKPACKER (K) Pack ATR image into cartridge using MAXFLASH driver.
-EXEPACKER (K) Create multi-cart image using directory of EXE files.
-FASTKEYS (E) Paster selection in EXE loader menu.
-FIXSEG (U) Combine with -SEGCHECK to try and fix an EXE file.
-FLASHER (*) Make bootable ATR image for programming MAXFLASH cart
-HOLMES (D) Try to follow archive directives (-bas, -osb, etc).
-LOFLICKER (D) Reduce flickering in some modes.
-LOMEM (D) Force HIMEM to A000 after installation.
-MEMPACKER (K) Pack Atari800 A8S image into cartridge.
-NOBIN (*) Delete raw binary after processing.
-NODISK (D) Create image with blank disk area.
-NOEXE (E) Erase EXE files after packing with -EXEPACKER option.
-OSB (*) Build automatic OS-B translator into cartridge.
-PALHACK (D) Force OS to detect GITA as PAL.
-RECURSIVE (*) Process sub-directories in directory mode.
-REINIT (E) Force EXE loader to enter RUN vector via warmstart
-SEGCHECK (U) Display segment information in an Atari EXE file.
-TRANSLATOR (D) Replace OS but not not install any drivers.
-TRIGGERS (D) Enable sector triggers.
-UNDERS (M) Unpack memory under OS.
-VERBOSE (*) Display useless information.
```

Wenn du die oben gezeigte Nachricht erscheint, dann "Herzlichen Glückwunsch!" - Sie haben die Perl, TASM und die Maxflash-Software richtig installiert!

Wenn eine Fehlermeldung erscheint, das Perl oder TASM nicht gefunden wurden, gehe Sie zurück und überprüfen Sie die Installation nach den vorhergehenden Schritten.

5 Benutzung der Maxflash Software

(Übersetzt von Carsten Strotmann)

5.1 Einleitung

Die Maxflash Software besitzt drei Hauptfunktionen für die Erstellung der Cartridge-Abbilder, und eine Reihe von Unterstützungs-Funktionen.

Wird das "maxflash.pl" Script mit dem Perl Interpreter ausgeführt ("perl maxflash.pl"), erscheint die folgende Hilfmeldung:

```
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
usage: perl maxflash.pl [switches] [hacks] [disk.atr] [cartridge.bin] [dump.a8s]
switches: -4MB (*) Create image for 4Mb cartridge. (default is 1Mb)
-8MB (*) Create image for 8Mb cartridge. (default is 1Mb)
-ALTPTR (E) Use alternate address for Atari EXE loader.
-BASIC (D) Enable BASIC. (For packing disks that use BASIC)
-BATCH (*) Never prompt for optional information.
-BIN2ALL (U) Make existing BIN file into CART/ATR images.
-BIN2ATR (U) Make existing BIN file into flash programming image.
-BIN2CART (U) Make existing BIN file into Atari800 CART image.
-CART (*) Write file with Atari800 cartridge header.
-CHEESE (E) Enable DLIs, sounds and colors in EXE menu.
-COLORS (*) Fill color registers while working. (debug)
-DISKPACKER (K) Pack ATR image into cartridge using MAXFLASH driver.
-EXEPACKER (K) Create multi-cart image using directory of EXE files.
-FASTKEYS (E) Faster selection in EXE loader menu.
-FIXSEG (U) Combine with -SEGCHECK to try and fix an EXE file.
-FLASHER (*) Make bootable ATR image for programming MAXFLASH cart
-HOLMES (D) Try to follow archive directives (-bas, -osb, etc).
-LOFLICKER (D) Reduce flickering in some modes.
-LOMEM (D) Force HIMEM to A000 after installation.
-MEMPACKER (K) Pack Atari800 A8S image into cartridge.
-NOBIN (*) Delete raw binary after processing.
-NODISK (D) Create image with blank disk area.
-NOEXE (E) Erase EXE files after packing with -EXEPACKER option.
-OSB (*) Build automatic OS-B translator into cartridge.
-PALHACK (D) Force OS to detect GITA as PAL.
-RECURSIVE (*) Process sub-directories in directory mode.
-REINIT (E) Force EXE loader to enter RUN vector via warmstart
-SEGCHECK (U) Display segment information in an Atari EXE file.
-TRANSLATOR (D) Replace OS but not not install any drivers.
-TRIGGERS (D) Enable sector triggers.
-UNDEROS (M) Unpack memory under OS.
-VERBOSE (*) Display useless information.
hacks: @PORTB (*) Delete writes to PORT B in image.
Each switch is designated a function class, these are:
* (K) Key Function (One of the three main script functions)
* (U) Utility Function (Invokes a special stand-alone script function)
* (E) Exepacker Switch (Modified -EXEPACKER behavior)
* (D) Diskpacker Switch (Modifies -DISKPACKER behavior)
* (M) Mempack Switch (Modifies -MEMPACK behavior)
* (*) Global Switch (Modifies behavior of all functions)
```

Bei der Benutzung des "maxflash.pl" Programmes muß immer mindestens eine Schlüsseloption ([K]ey option) oder eine Hilfsfunktion ([U]tility option) gewählt werden. Die anderen Schalter beeinflussen die Ausführung dieser Optionen.

Die Schalter, Dateinamen und Verzeichnisse können in beliebiger Reihenfolge angegeben werden.

5.2 Schlüsseloptionen

Die Schlüsselfunktionen sind im Hilfetext mit dem (K)=Key markiert. Diese Optionen sind:

```
-EXEPACKER (K) Create multi-cart image using directory of EXE files.
-DISKPACKER (K) Pack ATR image into cartridge using MAXFLASH driver.
-MEMPACKER (K) Pack Atari800 A8S image into cartridge.
```

Die "-EXEPACKER"-Option wird zur Erstellung einer "Multicart" benutzt.

5 BENUTZUNG DER MAXFLASH SOFTWARE

Diese Option des maxflash.pl Scripts erzeugt automatisch aus ausführbaren Atari Programmen (COM oder EXE) ein voll funktionsfähiges Atari 8-bit Multicart-Abbild.

Das Packen der Programme, die Erstellung des Menüs, Anpassen an die Cartridge und die Erstellung des Flash-Abbildes wird automatisch durchgeführt.

Die mit der -EXEPACKER erstellten MultiCarts funktionieren auf jedem Atari 8-Bit Computer und benötigen keine Betriebssystem (OS) Funktionen.

Hier nun ein Beispiel der -EXEPACKER Option zur Erstellung eines 1MBit Flash-Cartridge Abbildes:

```
>perl maxflash.pl -exepacker -cheese -altptr -cart "Z:\Jellystone Park Archive\games\exe\A-C" MyMultiCart.bin
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
Looking over EXE files in Z:\Jellystone Park Archive\games\exe\A-C
Using ARENA.exe (7999 bytes), 114881 bytes remaining.
Using ATLANTIS.exe (8334 bytes), 106547 bytes remaining.
Using AceyDuecy.exe (15370 bytes), 91177 bytes remaining.
Using ActionQuest.exe (10440 bytes), 80737 bytes remaining.
Using Adax.exe (33224 bytes), 47513 bytes remaining.
Using Adventure 7 Mystery Fun.exe (19983 bytes), 27530 bytes remaining.
Using AdventureCreator.exe (16491 bytes), 11039 bytes remaining.
Using AdventuresOfMrFish.exe (8797 bytes), 2242 bytes remaining.
Using Amazed.exe (1743 bytes), 499 bytes remaining.
Skipped Anteater.exe, may be damaged, run -SEGCHECK on it.
Assembled 9 files with 499 bytes of slack.
By default the names of the images that appear on the menu of the
flash cartridge will be the same as the files they were taken
from.
You can choose new menu descriptions here, or just
press Enter to keep what is shown (34 characters maximum).
Title [ARENA] :
Title [ATLANTIS] :
Title [AceyDuecy] :
Title [ActionQuest] :
Title [Adax] :
Title [Adventure 7 Mystery Fun] :
Title [AdventureCreator] :
Title [AdventuresOfMrFish] :
Title [Amazed] :
Please enter a title for this cartridge. This text will
appear at the top-most line of the multicart menu. Just
press enter to keep the default description
Cartridge Title: [ATARIMAX Flash Multicart EXE Loader] :
Saved new binary flash image to MyMultiCart.bin
Saved new Atari800 "CART" image to MyMultiCart.car
E:\maxflash>
```

Bei der "-EXEPACKER"-Option müssen sie zusätzlich angeben:

- Das Quellverzeichnis mit den ausführbaren Dateien, welche in die Cartridge gepackt werden sollen
- Der Ziel-Dateinamen für das .BIN Cartridge-Abbild.

Im oben gezeigten Beispiel werden noch folgende Optionen benutzt:

- -CHEESE – Aktiviert DLI-Farben und Töne im MultiCart Menü
- -ALTPTR – Verschiebt die Entpackroutine an eine alternative Adresse
- -CART – Erzeugt ein Cartridge-Abbild für den Test im Atari800 Emulator

5 BENUTZUNG DER MAXFLASH SOFTWARE

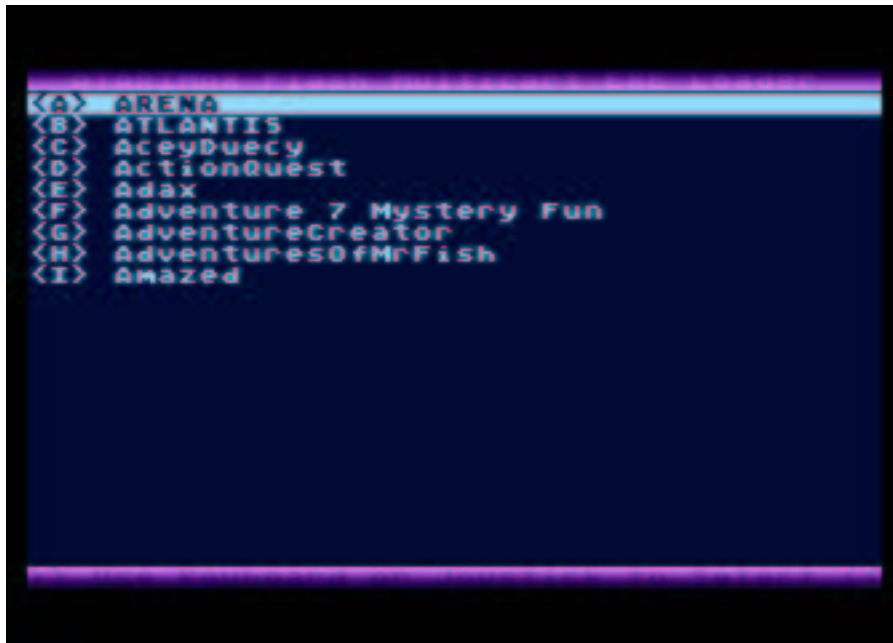


Abbildung 4: Die fertiggestellte Cartridge

Weitere Optionen, die bei der Erstellung von MultiCart Abbildern verwendet werden können:

- -4MB/-8MB – Erstellen eines 4 Mbit oder 8Mbit Cartridge-Abbildes
- -OSB – zur Benutzung von "400/800"-Software auf XL/XE Maschinen (Translator/Old-Runner)
- -FASTKEYS – Schnelles Scrolling/Auswählen im Menü
- -COLORS – Fehlersuch-Option, Erzeugt ein Farbflackern beim Entpacken
- -FLASHER – Erzeuge ein abschliessendes Flash-Cartridge ATR Programmier-Abbild
- -NOBIN – Lösche die ".BIN"-Datei nach der Abbild-Erstellung
- -NOEXE – Lösche die EXE/COM Dateien nach dem Einfügen in das Cartridge Abbild
- -REINIT – Starte das MultiCart Menü über den Warmstart-Vektor
- -VERBOSE – Ausgabe von Debug-Informationen während der Scriptausführung

5.3 Packen der EXE/COM Dateien

Wenn mit der “-EXEPACKER”-Option ein Verzeichnis mit ATARI EXE Programmen (ausführbare COM-Dateien) angegeben wird, so versucht das Script so viele Dateien wie möglich in die Cartridge zu packen.

Hier ist als Beispiel die Ausgabe des Scripts beim Packen der Dateien:

```
Looking over EXE files in Z:/Jellystone Park Archive/games/exe/A-C
Using ARENA.exe (7999 bytes), 114881 bytes remaining.
Using ATLANTIS.exe (8334 bytes), 106547 bytes remaining.
Using AceyDucey.exe (15370 bytes), 91177 bytes remaining.
Using ActionQuest.exe (10440 bytes), 80737 bytes remaining.
Using Adax.exe (33224 bytes), 47513 bytes remaining.
Using Adventure 7 Mystery Fun.exe (19983 bytes), 27530 bytes remaining.
Using AdventureCreator.exe (16491 bytes), 11039 bytes remaining.
Using AdventuresOfMrFish.exe (8797 bytes), 2242 bytes remaining.
Using Amazed.exe (1743 bytes), 499 bytes remaining.
Skipped Anteater.exe, may be damaged, run -SEGCHECK on it.
Assembled 9 files with 499 bytes of slack.
```



Abbildung 5: Menü einer 8MBit Cartridge

Für jede Datei, die hinzugefügt wird, gibt das Script die Information aus, wie viel Platz das Programm in der Cartridge belegt und wieviel Platz in der Cartridge noch frei ist.

Wird eine Datei oder ein Programm gefunden, welches defekt oder nicht ladbar ist, so wird eine Meldung ausgegeben mit der Information das die Option “-SEGCHECK” manuell auf diese Datei angewendet werden sollte um die Datei oder das Programm zu reparieren.

Im obigen Beispiel bleiben, nachdem alle EXE Dateien bearbeitet sind, noch 499 Byte Restplatz (Slack) frei. Das ist der ungenutzte Platz in der Cartridge. Es ist normal das ein wenig Platz frei bleibt, da es nicht viele Programme gibt die z.B. in 499 Byte passen.

5 BENUTZUNG DER MAXFLASH SOFTWARE

Wenn defekte Dateien gemeldet werden, sollten diese Dateien mit der Option “-SEG-CHECK” und “-FIXSEG” des “maxflash.pl”-Scriptes behandelt werden, um die Fehler ggf. automatisch zu beheben.

5.3.1 Eingabe der Menü-Texte

Beim Packen von Programmen in ein Multicart-Abbild benutzt das Script die Dateinamen als Standardwert für die Menü-Texte. Diese Menütexte werden angezeigt, wenn die Cartridge eingesteckt und gestartet wird. Wenn zum Beispiel das Programm “ArchonII.EXE” gepackt wird, so wird “ArchonII” als Standard-Menütext verwendet.

Da aber meist schönere Menü-Texte gewünscht sind, fragt das Script für jede Datei nach einem alternativen Menütext.

```
By default the names of the images that appear on the menu of the
flash cartridge will be the same as the files they were taken
from.
You can choose new menu descriptions here, or just
press Enter to keep what is shown (34 characters maximum).
Title [ARENA] :
Title [ATLANTIS] :
Title [AceyDuecy] :
Title [ActionQuest] :
Title [Adax] :
Title [Adventure 7 Mystery Fun] :
Title [AdventureCreator] :
Title [AdventuresOfMrFish] :
Title [Amazed] :
```

Der Text in den eckigen Klammern ist der Standardtext, identisch mit dem Dateinamen des Programms. Dieser kann einfach mit der [ENTER]-Taste übernommen werden. Alternativ kann ein neuer Text eingegeben und mit [ENTER] übernommen werden.

Es ist zu beachten das die Bildschirmbreite des ATARI beschränkt ist und die Länge der Texte daher bis 34 Zeichen betragen sollte.

5.3.2 Eingabe des Cartridge Titels

Nachdem die Texte für die MultiCart-Auswahl eingegeben sind, kann der Text über dem Auswahlmenü geändert werden.

```
Please enter a title for this cartridge. This text will
appear at the top-most line of the multicart menu. Just
press enter to keep the default description
Cartridge Title: [ATARIMAX Flash Multicart EXE Loader] :
```

Dieser Text erscheint am oberen Rand der Menüauswahl. Die maximale Länge dieses Textes ist 40 Zeichen.

5.3.3 Erstellen eines Disketten-Abbildes

Nachdem der Cartridgetitel eingegeben wurde, schließt das Script die Zusammenstellung ab und erzeugt die gewünschten Abbilder (Images).

```
Saved new binary flash image to MyMultiCart.bin
Saved new Atari800 *CART* image to MyMultiCart.car
```

Abhängig von den gewählten Optionen werden die folgenden Abbilder erstellt:

- *YourTitle.bin* - Die “rohen” Cartridge Daten ohne Kopf-Daten (Header)

5 BENUTZUNG DER MAXFLASH SOFTWARE

- *YourTitle.car* - Ein Atari800 Cart-Abbild zum Test im Atari800 Emulator
- *YourTitle.atr* - Ein ATR-Diskettenabbild, wird benutzt um die Flashcart zu programmieren

Ein “.BIN”-Abbild wird immer erzeugt. Mit der “-NOBIN” Option kann das Script angewiesen werden, das BIN-Abbild nach Beendigung automatisch zu löschen. Das “.CAR”-Abbild wird nur erzeugt, wenn die “-CART”-Option angegeben wurde. Das “.ATR”-Abbild wird erzeugt wenn die Option “-FLASHER” beim Start des Scriptes angegeben wurde.

Das “.CAR”-Abbild macht es einfach, die Cartridge im Atari800 Emulator zu testen. Das “.BIN”-Abbild kann auch im Emulator benutzt werden. Wenn ein BIN-Abbild in den Atari800 Emulator geladen wird, muss der korrekte Cartridge-Typ manuell gewählt werden.

Ein Test der Cartridge im Emulator vor der Programmierung einer echten Flash-Cartridge spart Zeit und Aufwand wenn noch Änderungen notwendig sind.

Siehe Kapitel 3 zur Benutzung des “.ATR”-Abbildes, um eine Cartridge zu programmieren.

5.4 Erstellen einer Disketten-Cartridge

Information: Diese Anleitung bezieht sich nur auf 1MBit Cartridges

5.4.1 Erstellen des Abbildes

Die “-DISKPACKER” Option des “maxflash.pl”-Scripts kann benutzt werden, um ein Atari 8-Bit Diskettenspiel in eine AtariMax Cartridge zu konvertieren.

Die Arbeit des Packen des Programmes, Einpassen in die Cartridge, Anpassen des Atari Betriebssystems und die Erstellung des Brenn-Images wird automatisch durchgeführt.

Disketten-Cartridges die mit der automatischen “-DISKPACKER” Option erstellt wurden benötigen einen Atari XL/XE Computer.

In folgenden Beispiel wird das Infocom Spiel “Zork 1” mit der “-DISKPACKER” Option in eine Cartridge gebracht:

```
>perl maxflash.pl -diskpacker "Z:\Jellystone\atr\T-Z\Zork I.atr" zork_i.bin -cart -flasher
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
New binary image [zork_i.bin] created.
Saved new Atari800 "CART" image to zork_i.car
When you load the flash cartridge programming disk on the Atari Computer
it will display the title of the software to be programmed. You can change
what is displayed, or just press ENTER to keep what is shown. Note: you
may use a maximum of 34 characters in this field.
Title [zork_i] : Zork I The Great Underground Empire
Saved new cartridge programming image to "zork_i (maxflash image).atr"
```

Bei der Benutzung der “-DISKPACKER”-Option muss angegeben werden:

- Das Quell-ATR-Disketten-Abbild, welches in die Cartridge gepackt werden soll
- Den Zielnamen für das “.BIN”-Abbild

5 BENUTZUNG DER MAXFLASH SOFTWARE

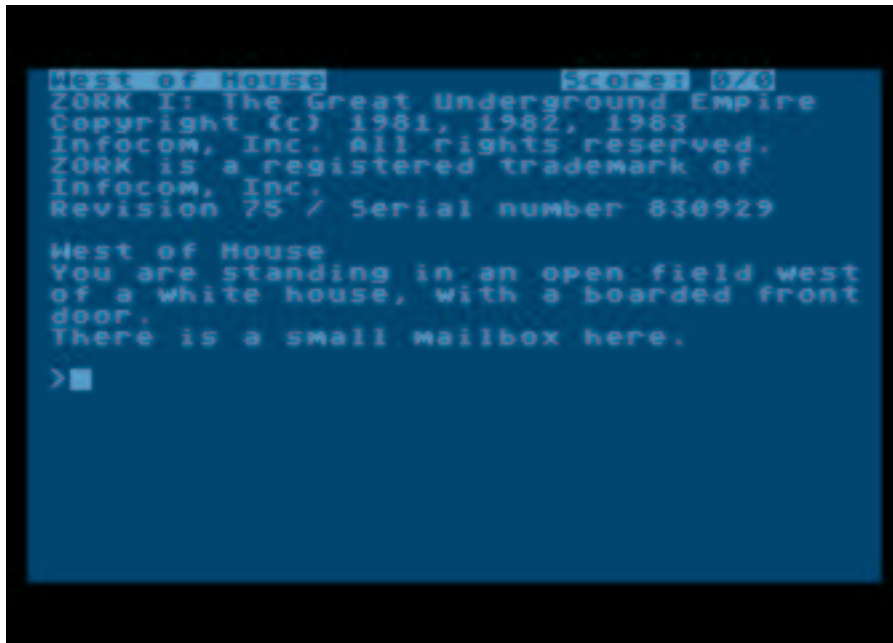


Abbildung 6: Beispiel einer Disketten Cartridge

Im obigen Beispiel wurden zwei zusätzliche Optionen benutzt:

- -FLASHER - Erzeuge ein abschliessendes Flash-Cartridge ATR Programmier Abbild
- -CART - Erzeugt ein Cartridge-Abbild für den Test im Atari800 Emulator

Weitere Optionen, die bei der Erstellung einer Disketten-Cartridge benutzt werden können, sind:

- -4MB/-8MB – Erstellen eines 4 Mbit oder 8Mbit Cartridge-Abbildes
- -OSB – zur Benutzung von "400/800"-Software auf XL/XE Maschinen (Translator/Old-Runner)
- -BASIC – Aktiviere BASIC für die Software auf der Cartridge
- -FLASHER – Erzeuge ein abschliessendes Flash-Cartridge ATR Programmier-Abbild
- -HOLMES – Versuche die Benötigten Options-Informationen aus dem Dateinamen zu ermitteln
- LOFLICKER – Reduzierung/Verhinderung von Bildschirmflackern durch zeitgesteuerten Zugriff auf die Cartridge



Abbildung 7: Ali-Baba mit Translator (-OSB)

- -LOMEM – Verhindere Zugriff auf die oberen 16K des RAM (erzwingt Cartridge Erkennung)
- -VERBOSE – Ausgabe von Debug-Informationen während der Scriptausführung
- -NODISK – Erstelle eine leere Disk Cartridge ohne Disketten Image
- -PALHACK – Erzwingt die Erkennung eines PAL GTIA durch das OS
- -TRANSLATOR – Erzeuge eine Translator Cartridge (kein Patch des OS)
- -TRIGGERS – Erzeuge Code zum Diskettenwechsel (siehe erweiterte Optionen)

Nicht alle Diskettenabbilder können automatisch in eine Cartridge gewandelt werden. Die meisten Utility-Disketten funktionieren, zum Beispiel ist die Erstellung einer DOS-Cartridge so einfach wie das Zork-Beispiel.

Cartridges die nicht mit der automatischen Umwandlung funktionieren sind Programme welche die OS-SIO-Routinen umgehen, OS-Bereiche überschreiben oder PORT B ändern.

Allgemein kann gesagt werden, das mit ein wenig Aufwand die meisten Spiele funktionsfähig gemacht werden können. Meist ist eine kleine Änderung notwendig, um eine Re-Initialisierung des PORT B zu verhindern.

5 BENUTZUNG DER MAXFLASH SOFTWARE

Hunderte von Spielen aus dem Holmes und Jellystone Park Archiv konnten von dem Beta-Test-Team umgewandelt werden. Prüfe das Forum⁵ und das Archiv ob schon jemand das gewünschte Spiel/die Anwendung umgewandelt hat, um Zeit zu sparen.

Im Kaptiel 8 finden sich weitere Informationen über die Anpassung von Programmen über den modifizierten Atari800Win Emulator, und wie man diese Programme auch für nicht XL/XE Computern anpassen kann.

5.4.2 Eingabe des Cartridge Titels

Bei der Erstellung des Diskettenabbildes der Cartridge wird der Titel der Cartridge angezeigt. Der vorgeschlagene Titel kann mit [ENTER] übernommen werden oder es kann ein neuer, bis zu 34 Zeichen langer, Titel eingegeben werden:

```
Title [zork_i] : Zork I The Great Underground Empire
Saved new cartridge programming image to "zork_i (maxflash image).atr"
```

Dieser Text wird bei der Programmierung der Flash-Cartridge angezeigt.

5.4.3 Erstellen eines Disketten-Abbildes

Nachdem der Cartridgetitel eingegeben wurde schliesst das Script die Zusammenstellung ab und erzeugt die gewünschten Abbilder (Images).

```
Saved new binary flash image to MyMultiCart.bin
Saved new Atari800 "CART" image to MyMultiCart.car
```

Abhängig von den gewählten Optionen werden die folgenden Abbilder erstellt:

- *YourTitle.bin* - Die "rohen" Cartridge Daten ohne Kopf-Daten (Header)
- *YourTitle.car* - Ein Atari800 Cart-Abbild zum Test im Atari800 Emulator
- *YourTitle.atr* - Ein ATR-Diskettenabbild, wird benutzt um die Flashcart zu programmieren

Ein ".BIN"-Abbild wird immer erzeugt. Mit der "-NOBIN" Option kann das Script angewiesen werden, das BIN-Abbild nach Beendigung automatisch zu löschen. Das ".CAR"-Abbild wird nur erzeugt, wenn die "-CART"-Option angegeben wurde. Das ".ATR"-Abbild wird erzeugt wenn die Option "-FLASHER" beim Start des Scriptes angegeben wurde.

Das "CAR"-Abbild macht es einfach, die Cartridge im Atari800 Emulator zu testen. Das "BIN"-Abbild kann auch im Emulator benutzt werden. Wenn ein BIN-Abbild in den Atari800 Emulator geladen wird, muss der korrekte Cartridge-Typ manuell gewählt werden.

Ein Test der Cartridge im Emulator vor der Programmierung einer echten Flash-Cartridge spart Zeit und Aufwand wenn noch Änderungen notwendig sind.

Siehe Kapitel 3 zur Benutzung des "ATR"-Abbildes um eine Cartridge zu programmieren.

⁵<http://www.atarimax.com/>

5.5 Erstellen eines Atari800 A8S Speicher-Abbildes

Die Unterstützung für A8S-Speicherabbilder mit der "-MEMPACKER"-Option wurde hinzugefügt um das Entpacken von Speicherbereichen aus der Cartridge zu demonstrieren. Die Benutzbarkeit dieser Option ist beschränkt. Da 99% der Software, welche mit "-MEMPACKER" bearbeitet werden kann auch mit den anderen Optionen funktioniert, wird diese Option wahrscheinlich nicht weiterentwickelt.

5.5.1 MEMPACKER

Die "-MEMPACKER" Option des maxflash.pl Scriptes kann benutzt werden um Atari800 A8S Speicher-Abbilder automatisch in eine Cartridge zu konvertieren.

Die Arbeit des Packen des Programmes, Einpassen in die Cartridge, Anpassen des Atari Betriebssystems und die Erstellung des Brenn-Images wird automatisch durchgeführt.

Mit der "-MEMPACKER"-Option erstellte Cartridges unterliegen keiner Hardwarebeschränkung. Wenn die "-UNDEROS" Option angegeben wird, so versucht die Cartridge-Software den Speicher unter dem OS zu benutzen. Dieser ist nur auf Atari XL/XE Modellen verfügbar.

Das folgende Beispiel benutzt ein Speicher-Abbild des Spiels "Bounty Bob Strikes Back":

```
>perl maxflash.pl -mempacker -underos T:\bounty_bob.a8s bbsb.bin -cart -flasher -colors
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
New image [bbsb.bin] created.
Saved new Atari800 "CART" image to bbsb.car
When you load the flash cartridge programming disk on the Atari Computer
it will display the title of the software to be programmed. You can change
what is displayed, or just press ENTER to keep what is shown. Note: you
may use a maximum of 34 characters in this field.
Title [bbsb] : Bounty Bob Strikes Back!
Saved new cartridge programming image to "bbsb (maxflash image).atr"
```

Bei der Benutzung der "-MEMPACKER" Optionen müssen die folgenden Parameter angegeben werden:

- Die Quelldatei mit dem A8S-Speicherabbild
- Der Zielfilename für das ".BIN" Cartridge-Abbild

Im vorhergehenden Beispiel wurden zusätzlich diese Optionen verwendet:

- FLASHER – Erzeuge ein abschliessendes Flash-Cartridge ATR Programmier Abbild
- -CART – Erzeugt ein Cartridge-Abbild für den Test im Atari800 Emulator
- -UNDEROS – Benutze den Speicher unter dem OS bei XL/XE Geräten
- -COLORS – Zeige Farbeflickern beim Enpacken in den Speicher (Fehlersuch-Option)

Weitere Optionen die Benutzt werden können:

- -OSB – Benutze OSB (Atari800) für nicht XL/XE Software (Translator)
- -VERBOSE – Zeige Fehlersuch-Informationen bei Ausführen des Scriptes



Abbildung 8: Bounty Bob Strikes Back

- -PALHACK – Erzwingt die Erkennung eines PAL GTIA durch das OS

Die “-COLOR”-Option ist zur Fehlersuche gedacht. Während des Entpackens in den Speicher werden zufällige Farben auf dem Bildschirm dargestellt, um den Entpackprozess anzuzeigen.

Derzeitige Beschränkungen der “-MEMPACKER” Funktion:

- Keine Unterstützung um den Hardware-Status eines A8S-Abbildes wiederherzustellen (einige Stati sind auf echter Hardware auch nicht möglich)
- die Prozessorregister (mit dem Prozesszähler, IP) werden nicht wiederhergestellt. Stattdessen wird ein Warmstart nach dem Entpacken ausgelöst. Die Software muss resetfest sein
- Der Prozessorstack (\$100) wird nicht wiederhergestellt
- Keine Unterstützung für mehr als ein A8S-Abbild pro Cartridge

5.5.2 Erstellen eines Abbildes

Nachdem der Cartridgetitel eingegeben wurde (wenn die “-FLASHER”-Option angegeben wurde) schliesst das Script die Zusammenstellung ab und erzeugt die gewünschten Abbilder (Images).

```
New image [bbsb.bin] created.
```

Abhängig von den gewählten Optionen werden die folgenden Abbilder erstellt:

5 BENUTZUNG DER MAXFLASH SOFTWARE

- *YourTitle.bin* - Die "rohen" Cartridge Daten ohne Kopf-Daten (Header)
- *YourTitle.car* - Ein Atari800 Cart-Abbild zum Test im Atari800 Emulator
- *YourTitle.atr* - Ein ATR-Diskettenabbild, wird benutzt um die Flashcart zu programmieren

Ein ".BIN"-Abbild wird immer erzeugt. Mit der "-NOBIN" Option kann das Script angewiesen werden, das BIN-Abbild nach Beendigung automatisch zu löschen. Das ".CAR"-Abbild wird nur erzeugt, wenn die "-CART"-Option angegeben wurde. Das ".ATR"-Abbild wird erzeugt wenn die Option "-FLASHER" beim Start des Scriptes angegeben wurde.

Das "CAR"-Abbild macht es einfach, die Cartridge im Atari800 Emulator zu testen. Das "BIN"-Abbild kann auch im Emulator benutzt werden. Wenn ein BIN-Abbild in den Atari800 Emulator geladen wird, muss der korrekte Cartridge-Typ manuell gewählt werden.

Ein Test der Cartridge im Emulator vor der Programmierung einer echten Flash-Cartridge spart Zeit und Aufwand wenn noch Änderungen notwendig sind.

Siehe Kapitel 3 zur Benutzung des "ATR"-Abbildes um eine Cartridge zu programmieren.

5.6 Hilfsprogramm-Optionen

Das maxflash.pl Script beinhaltet einige separate Hilfsprogrammfunktionen. Diese sind mit einem (U) in der Auflistung der Optionen gekennzeichnet:

```
-BIN2ALL (U) Make existing BIN file into CART/ATR images.  
-BIN2ATR (U) Make existing BIN file into flash programming image.  
-BIN2CART (U) Make existing BIN file into Atari800 CART image.  
-SEGCHECK (U) Display segment information in an Atari EXE file.  
-FIXSEG (U) Combine with -SEGCHECK to try and fix an EXE file.
```

5.6.1 "-BIN2ALL"-Option

Syntax:

```
perl maxflash.pl -BIN2ALL filename.bin
```

Die BIN2ALL-Funktion nimmt eine bestehende "BIN"-Datei und wandelt diese in je ein Atari CART-Abbild und ein AtariMax Flash Programmierabbild, "ATR"-Datei.

Dies erlaubt es beliebige Daten, selbst wenn diese nicht vom Script erzeugt wurden, in ein Flash-Programmier-Abbild zu verwandeln.

Beliebige Daten in der "BIN"-Datei werden in die Cartridge programmiert.

5.6.2 "-BIN2ATR"-Option

Syntax:

```
perl maxflash.pl -BIN2ATR filename.bin
```

Diese Option ist identisch zur "BIN2ALL"-Option, mit dem Unterschied das nur die "ATR"-Datei erzeugt wird.

5 BENUTZUNG DER MAXFLASH SOFTWARE

5.6.3 “-BIN2CAR”-Option

Syntax:

```
perl maxflash.pl -BIN2CAR filename.bin
```

Diese Option ist identisch zur “BIN2ALL”-Option, mit dem Unterschied das nur die “CAR”-Datei erzeugt wird.

“-SEGCHECK”-Option

Syntax:

```
perl maxflash.pl -SEGCHECK [-FIXSEG] atarigame.com
```

Diese Option analysiert die Segmente einer Atari-Programmdatei und korrigiert optional gefundene Fehler.

```
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
Checking segments in Z:\Jellystone\A-C\Archon II Adept.exe
Header marker 0xFFFF at 0x0000
Valid segment [0x2D80 - 0x2FFC] at 0x0002
Valid segment [0xD01F - 0xD01F] at 0x0283
Valid segment [0x02E2 - 0x02E3] at 0x0288
Segment sets immediate INIT address of 0x2D80
Valid segment [0x02E0 - 0x02E1] at 0x028E
Segment sets RUN address of 0x2DB6
Header marker 0xFFFF at 0x0294
Warning: Ignoring extraneous header embedded in file.
Valid segment [0x3000 - 0x49FF] at 0x0296
Header marker 0xFFFF at 0x1C9A
Warning: Ignoring extraneous header embedded in file.
Valid segment [0x49DE - 0xBA5D] at 0x1C9C
Valid segment [0x1A1A - 0x1A1A] at 0x8D20
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D25
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D2A
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D2F
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D34
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D39
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D3E
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D43
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D48
Warning: Ignoring junk segment. (transfer padding)
Valid segment [0x1A1A - 0x1A1A] at 0x8D4D
Warning: Ignoring junk segment. (transfer padding)
Warning: Ignoring junk segment. (transfer padding)
Warning: Ignoring 1 bytes of extra data at end of file.
Process completed with 22 warnings and 0 errors.
New filename: [Archon II Adept_fixseg.exe] : archon_ii_fixed.exe
Wrote archon_ii_fixed.exe, 36124 bytes.
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
Checking segments in archon_ii_fixed.exe
Header marker 0xFFFF at 0x0000
Valid segment [0x2D80 - 0x2FFC] at 0x0002
Valid segment [0xD01F - 0xD01F] at 0x0283
Valid segment [0x02E2 - 0x02E3] at 0x0288
Segment sets immediate INIT address of 0x2D80
Valid segment [0x02E0 - 0x02E1] at 0x028E
Segment sets RUN address of 0x2DB6
Valid segment [0x3000 - 0x49FF] at 0x0294
Valid segment [0x49DE - 0xBA5D] at 0x1C98
Process completed with 0 warnings and 0 errors.
```

Wenn die “-FIXSEG” -Option nicht angegeben wird, werden die Fehler nur ausgegeben, aber nicht berichtigt. Um eine neue, berichtigte Programmdatei zu sichern, muss die Option “-SEGCHECK” zusammen mit der Option “-FIXSEG” benutzt werden.

5 BENUTZUNG DER MAXFLASH SOFTWARE

Wird die “-FIXSEG” Option angegeben, so wird nach dem Dateinamen für die neue, berichtigte Datei erfragt.

Nachdem die Berichtigung der Datei abgeschlossen ist, wird die Funktion “-SEGCHECK” automatisch noch einmal ausgeführt um die neue Datei zu überprüfen.

Solange keine Fehlermeldungen während des Packvorganges mit der “-EXEPACKER”-Funktion ausgegeben werden, müssen die Programmdateien nicht mit den “-SEGCHECK”- und “-FIXSEG”-Funktionen bearbeitet werden.

6 Benutzung der Flash Cartridge

Übersetzt von Marc Brings

6.1 Einleitung

Abhängig davon, mit Hilfe welcher Software die Cartridge programmiert wurde, hat die Cartridge einige spezielle Funktionen, welche die Benutzung vereinfachen.

6.2 -EXEPACKER Cartridges (Multi-cart images)



Abbildung 9: Bildschirmfoto einer typischen mit der -EXEPACKER Option erzeugten Multicart

6.2.1 Systemvoraussetzungen

Bei der Benutzung eines Multicard-Image, welches mit der -EXEPACKER-Option erstellt wurde, gibt es keine speziellen Hardwareanforderungen.

Die -EXEPACKER-Menüsoftware arbeitet auf jedem ATARI, auch auf einem ATARI 400 mit nur 16k.

6.2.2 Das Modul umgehen (Cartridge Bypass)

Wenn der Computer mit gedrückter [OPTION]-Taste eingeschaltet wird, deaktiviert sich das Modul selbstständig und ermöglicht so einen ganz normalen Bootvorgang von Diskette.

Dieses Feature wurde eingefügt um eine einfache Neuprogrammierung ohne Tricks wie das Einstecken des Modules bei eingeschaltetem Computer zu ermöglichen.

Nur Module, welche mit den Standard-Konvertierungsprogrammen (standard conversion tools) erstellt wurden, enthalten diese Option. Wenn die Cartridge mit einem Programm erzeugt wurde, welches die Bypass-Funktion nicht enthält, findest du im Kapitel "Programmieren des Modules" weitere Informationen zur Neuprogrammierung.

6.2.3 Auswahl der Software (Software Selection)

Multi-Carts mit dieser Option haben ein eingebautes Text-Menü für die Auswahl der Software per Joystick oder Tastatur.

Auswahl mit dem Joystick Es kann ein Joystick in Port #1 benutzt werden um einen Titel von der Multicart auszuwählen.

[Auf] - vorherigen Titel anwählen

[Ab] - nächsten Titel anwählen

[Links] - vorherige Seite

[Rechts] - nächste Seite

[Feuer] - Programm starten

Auswahl über die Tastatur Die ATARI-Pfeiltasten (ohne [CONTROL]!) können zum Navigieren im Menü benutzt werden.

[Auf] - vorherigen Titel anwählen

[Ab] - nächsten Titel anwählen

[Links] - vorherige Seite

[Rechts] - nächste Seite

[Return] oder [Leertaste] - Programm starten

Zusätzlich befindet sich links von jedem Titel ein Buchstabe in Klammern. Es kann jedes Programm direkt durch Druck auf den entsprechenden Buchstaben gestartet werden.

Das Programm muß nicht auf dem Bildschirm zu sehen sein wenn es ausgewählt wird; wenn ein Spiel Auswahl [X] ist kann es einfach gestartet werden, indem die Taste [X] gedrückt wird, sobald das Menü erscheint.

6.3 Modul-Reset-Zeit

Im Gegensatz zu Standard-ROM-Modulen enthält das Flash-Modul ein eigenes Bankswitching, ein eigenes Programm und einen eigenen Zustandsspeicher.

Wenn der Computer aus- und wieder einschaltet wird, um ihn neu zu starten, lassen Sie ihn einige Sekunden ausgeschaltet, um sicher zu sein, daß das Modul seine vorhergehenden Einstellungen vergißt und normal startet.

6.4 -DISKPACKER Images (Disk cartridges)

6.4.1 Das Modul umgehen (Cartridge Bypass)

Wenn der Computer mit gedrückter [OPTION]-Taste einschaltet wird, deaktiviert sich das Modul selbst und ermöglicht so einen ganz normalen Bootvorgang von Diskette.

Diese Funktion wurde eingefügt um eine einfache Neuprogrammierung ohne Tricks wie das Einstecken des Modules bei eingeschaltetem Computer zu ermöglichen.

Nur Module, welche mit den Standard-Konvertierungsprogrammen (standard conversion tools) erstellt wurden, enthalten diese Option. Wenn die Cartridge mit einem Programm erzeugt wurde, welches die Bypass-Funktion nicht enthält, findest du im Kapitel "Programmieren des Modules" weitere Informationen zur Neuprogrammierung.

6.4.2 Systemvoraussetzungen

"Disk conversion cartridges", welche automatisch mit der "-DISKPACKER"-Option erzeugt wurden, benötigen einen Computer der XL/XE-Serie.

Ein mit einem "DISPACKER"-Abbild programmiertes Modul wird auf einem 400 oder 800 (nicht XL/XE) Computer als nicht gültig erkannt. Du kannst das Modul in diesen Maschinen aber löschen und neu programmieren.

Wenn die Cartridge mit einem nicht-standard Abbild programmiert wurde, hängen die Systemanforderungen davon ab, wie das Abbild erzeugt wurde.

6.4.3 Das Modul booten

Cartridges die mit der "-DISKPACKER"-Option erstellt wurden booten vollautomatisch.

Stecken Sie die Cartridge einfach in einen XL/XE und schalte den Rechner ein. Halten Sie nicht die [OPTION]-Taste gedrückt um das BASIC abzuschalten; dies würde die im Modul eingebaute Bypass-Funktion auslösen.

Sämtliche Konfigurations-Einstellungen, die bei der Erstellung des Cartridge-Abbildes vorgenommen wurden, zum Beispiel BASIC an-/abschalten, Atari 800 OSB-Translator an-/abschalten, etc, werden von der Cartridgesoftware automatisch vorgenommen.

6.4.4 Diskette umgehen (Disk Bypass)

Der eingebaute Treiber für "-DISKPACKER"-Module ersetzt den normalen OS-Zugriff auf Laufwerk #1 durch Zugriff auf das Modul-ROM

Es ist immer noch möglich auf das normale ATARI-Laufwerk #1 zuzugreifen, zum Beispiel um einen Spielstand zu laden oder zu speichern, indem während des Diskettenzugriffes [OPTION] und [SELECT] zusammen gehalten werden.

Wenn zum Beispiel das Spiel "Enchanter" von einem Flash-Cartridge geladen wird, und der Spielstand soll gespeichert werden, gehen Sie ganz normal vor, aber wenn die "Press Return to Begin"-Meldung kommt halten Sie [OPTION] + [SELECT] gedrückt und drücken dann [RETURN].

Halten Sie [OPTION] + [SELECT] solange gedrückt bis die Aufforderung kommt die Spieldiskette wieder einzulegen!

6.4.5 Seiten-Reset (Side Reset)

Eines der fortschrittlichen Merkmale der "-DISKPACKER"-Option ist es ein zweiseitiges Diskettenspiel, welches weniger als die Gesamtkapazität des Modules benötigt, auf ein 1Mb Flash-Modul zu packen.

Das Spiel "Enchanter" von Infocom zum Beispiel ist mit dieser Methode gepackt. Beide Seiten der Spieldisk liegen auf dem Modul, und der Wechsel von Seite 1 zu Seite 2 wird automatisch bei der Modulerstellung durchgeführt.

Manchmal ist es wünschenswert das Modul zurück auf "Seite 1" zu schalten. Zum Beispiel wenn man aufgefordert wird Seite 1 einzulegen wenn "RESTART" eingegeben wird, um "Enchanter" neu zu starten.

Es ist möglich den Diskettentreiber zu zwingen das Modul auf "Seite 1" zurückzuschalten indem [SELECT] und [START] während des Diskettenzugriffes gedrückt wird. (Hier ist der virtuelle Diskettenzugriff innerhalb des Modules gemeint.)

Wenn zum Beispiel in Enchanter "RESTART" eingegeben wird und die Aufforderung "Re-insert disk Side 1 and press RETURN" ausgegeben wird, halten Sie [SELECT] + [START] zusammen gedrückt und drücken Sie [RETURN].

Anders als beim [OPTION] + [SELECT]-Bypass muss [SELECT] + [START] nicht mehr festgehalten werden, sobald der Diskettenzugriff begonnen hat. Halten Sie die Tasten einfach nur gedrückt bis [RETURN] gedrückt wurde, dann lassen sie los.

6.4.6 Das Modul nach dem Booten abschalten (Post-boot Cartridge Disable)

Wenn der Flash-Cartridge-Romdisktreiber aus irgendwelchen Gründen vollständig abgeschaltet werden soll, kann dies durch gleichzeitiges Drücken von [START], [SELECT] und [OPTION] während irgendeines Diskettenzugriffes erreicht werden.

Wenn das Modul mit Hilfe von [START], [SELECT] und [OPTION] deaktiviert wurde, kann es nur durch Aus- und Wiedereinschalten des Computers aktiviert werden.

6.5 -MEMPACKER (A8S Memory Unpacker Cartridge)

6.5.1 Das Modul umgehen (Cartridge Bypass)

Wenn der Computer mit gedrückter [OPTION]-Taste einschaltet wird, deaktiviert sich das Modul selbst und ermöglicht so einen ganz normalen Bootvorgang von Diskette.

Diese Funktion wurde eingefügt um eine einfache Neuprogrammierung ohne Tricks wie das Einstecken des Modules bei eingeschaltetem Computer zu ermöglichen.

Nur Module, welche mit den Standard-Konvertierungsprogrammen (standard conversion tools) erstellt wurden, enthalten diese Option. Wenn die Cartridge mit einem Programm erzeugt wurde, welches die Bypass-Funktion nicht enthält, findest du im Kapitel "Programmieren des Modules" weitere Informationen zur Neuprogrammierung.

6.5.2 Systemvoraussetzungen

Ein "A8S unpacker Modul" wird auf jedem Computer laufen.

Wenn die "-UNDEROS"-Option bei der Erstellung angegeben wurde um in den Speicher unter dem XL/XE-OS zu unpacken, wird ein XL/XE-Computer benötigt um das Modul benutzen zu können.

6.5.3 Das Modul booten (Booting the Cartridge)

Module, die mit der "-MEMPACKER"-Funktion erstellt wurden, booten vollautomatisch.

Stecken Sie das Modul einfach in einen XL/XE und schalten Sie den Rechner ein. Halten Sie nicht die [OPTION]-Taste gedrückt um das BASIC abzuschalten; das würde die im Modul eingebaute Bypass-Funktion auslösen.

Sämtliche Konfigurations-Einstellungen, die bei der Erstellung des Modul-Abbildes vorgenommen wurden, zum Beispiel BASIC an-/abschalten, 800 OSB-Translator an-/abschalten, ect. werden von der Cartridgesoftware automatisch vorgenommen.

6.6 Individuelle Cartridge Abbilder (Custom Cartridge Images)

Das Flash-Modul ist flexibel. Diese Flexibilität beinhaltet auch die Möglichkeit mit einem Programm programmiert zu werden, den jemand anders erstellt hat. (Hier dürfte das Programm mit dem das Modul programmiert wird gemeint sein; nicht das XL-Programm das auf das Modul programmiert wird. M.B.)

Das Modul welches zum Beispiel in dem "Mule Cartridge Tutorial" erstellt wurde, ist eine individuelle Anpassung (custom conversion). In diesem Fall wurde ein Diskettenspiel von Hand mit einem speziellen Programm ohne den "disk-cartridge-driver" erstellt und arbeitet deswegen auch Atari800-Rechnern.

Wenn ein Modul mit einem individuellen Flash-Abbild programmiert wurde, dürfte keiner der oben angegebenen Bypass-Tricks funktionieren.

Kläre mit der Person die das Abbild erstellt hat, wie es korrekt benutzt wird.

7 Emulator Unterstützung

Übersetzt von Ralf Patschke

Anmerkung des Übersetzers:

Ich habe mich bemüht, den Text soweit, wie möglich in einem verständlichen Deutsch zu formulieren. Leider ist es mir nicht überall gelungen. Ich verwende nicht die "neue deutsche Rechtschreibung". Also bitte keine Hinweise diesbezüglich :)

Viel Spaß mit dem AtariMaxFlash Modul, Ralf Patschke.

7.1 Einleitung

MaxFlash Cartridges Abbilder werden voll vom Atari800Win Plus Emulator unterstützt.

Es wird empfohlen jedes eigene Abbild zuerst mit dem Emulator zu testen, bevor es auf die ATARI-MaxFlash-Cartridge gebrannt wird.

Man kann mit dem Emulator selbst erstellte Abbilder sehr schnell auf Lauffähigkeit testen. Dadurch kann man sehr viel Zeit sparen. Obwohl keiner der ATARI Emulatoren 100% perfekt sind, kann man trotzdem schnell Fehler finden, die die Funktion des Moduls auf dem echten ATARI beeinträchtigen könnten.

7.2 Änderungen am Atari800Win Plus 3.1

Die hier angebotene Version des ATARI800WinPlus ist nicht identisch mit der Version 3.1. Folgende Änderungen sind vorgenommen worden:

- Unterstützung von 1Mbit, 4Mbit und 8Mbit Flash Cartridge Abbildern, im ".bin" und ".car" Format
- Anzeige der Adresse einer Routine, die in ROM-Bereiche schreibt
- Anzeige der Adresse einer Routine, die das OS vom RAM zum ROM bzw. umgekehrt verschiebt
- kleine Änderungen am Kaltstartprogramm

Sämtliche Änderungen am Programm unterliegen den gleichen Lizenzbestimmungen der GNU Public Lizenz, wie die übrigen Teile des AtariMax Programmpaketes. Siehe auch gpl.txt im Archiv für eine Kopie der GNU Public License.

7.3 Herunterladen des Emulators

Der Atari800Win Plus Emulator ist ein Programm, das unter der GNU Lizenz veröffentlicht wurde.

- Programmdatei für Windows
- Quellcode mit den Änderungen

7.4 Starten des Emulators

Zum Betrieb des Emulators ist eine Kopie des OS-Roms des ATARI nötig. Da diese Roms dem Copyright unterstehen, sind sie nicht im Programmpaket enthalten.

Auf der Internetseite des Atari800 Emulators⁶ kann man Kopien finden, falls man selbst keine Kopien erstellen kann.

Suche bei Google nach 800XL ROM Images. Es ist wahrscheinlich nur legal diese Kopien zu nutzen, wenn man selbst einen entsprechenden ATARI Rechner besitzt.

7.5 Mehr zum Emulator

Steve Tucker hat den Emulator nicht geschrieben und bietet deshalb auch keine Unterstützung dafür. Auf diesen Seiten sind Informationen zum Betrieb zu finden.

- Hauptseite des Atari800 Win Plus - http://atariarea.histeria.pl/PLus/index_us.htm
- Hauptseite des Atari800 Win - <http://www.concentric.net/~Twist/atari800win/>

7.6 Testen eines -EXEPACKER (Multi-cart) oder -MEMPACKER (A8S) Flash Cartridge Image mit dem Emulator

Den modifizierten Emulator am besten mit folgenden Einstellungen starten:

- MachineType XL/XE – REQUIRED
- MemoryType 64k – REQUIRED
- DisableBASIC – ENABLED
- SIO Patch – DISABLED
- Video System – n/a

7.6.1 Wahl der Moduldatei

Bei Nutzung des "maxflash.pl"-Scripts wird entweder eine ".bin" oder ".car" Abbild-Datei erstellt.

- .bin Dateien sind Rohdaten einer Cartridge.
- .car Dateien sind Atari800Win 'CART' Abbilder, die die Rohdaten, eine Identifikation (welcher Cartridge Typ), sowie eine Kontrollsumme enthalten.

⁶<http://atari800.sourceforge.net/>

7 EMULATOR UNTERSTÜTZUNG

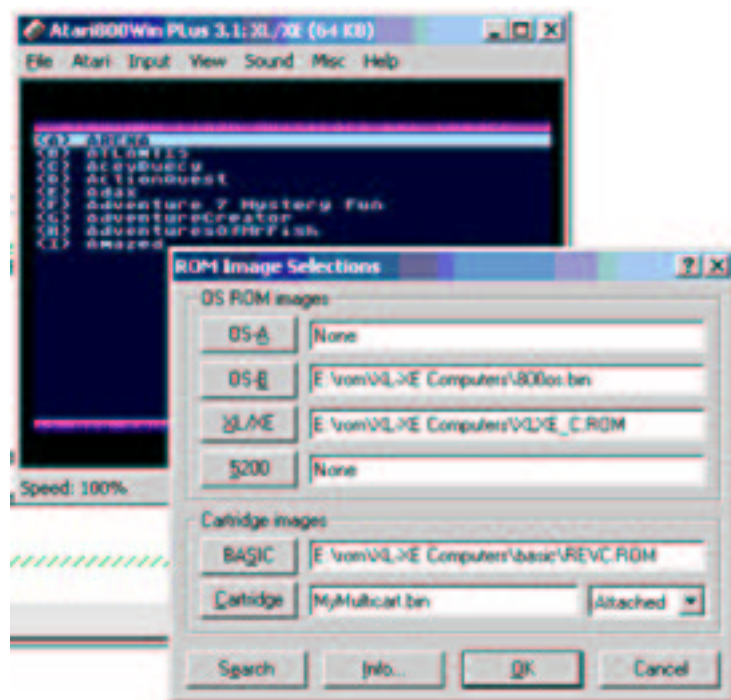


Abbildung 10: Einstellen der Optionen des Emulators

7 EMULATOR UNTERSTÜTZUNG

Zum Laden eines “.bin” oder “.car” Abbildes wird im Menüpunkt 'Atari' das Untermenü 'ROM Images' gewählt. Im erscheinenden Dialog muß auf 'Cartridge' gedrückt werden und anschließend die gewünschte Datei im Dateiauswahldialog gewählt werden.

Falls ein “.car” Abbild gewählt wurde, erkennt Atari800Win die Korrekte Größe und den Typ des Moduls alleine. Bei einem “.bin” Abbild wird man nach dem korrekten Typ des Moduls gefragt.

Rechts in der Auswahlbox für die Cartridges ist ein Klappmenü, in dem man die Cartridge de-/aktivieren kann. Um eine Cartridge in den Emulator "einzustecken", muß man hier auf "attached" stellen. "Detached" steht hier für ein nicht eingestecktes Modul (Cartridge).

Um ein “.BIN”-Abbild einer 1Mb Flash Cartridge einzulegen, muß im Klappmenü "Atarimax 1Mb Flash Cartridge" ausgewählt werden.

Nach Drücken auf "Ok" gelangt man in das Emulatorfenster zurück.

7.6.2 Starten des Moduls

In der Standardkonfiguration startet der Atari800WinPlus sofort, wenn ein Modul eingelegt wurde, neu.

Es ist möglich, daß ein Kaltstart von Nöten ist, um ein Modul zu starten. Falls nach dem Einlegen eines Moduls nichts passiert, ist er daher einfach mittels Shift+F5 her- vorzurufen.

7.7 Testen eines -DISKPACKER (Disk-cart) Flash Cartridge Images mit dem Emulator

7.7.1 Einstellen der Optionen des Emulators

Den modifizierten Emulator am besten mit folgenden Einstellungen starten:

- MachineType – XL/XE REQUIRED
- MemoryType – 64k REQUIRED
- DisableBASIC – ENABLED
- SIO Patch – DISABLED
- Video System – n/a

7.7.2 Wahl der Moduldatei

Wenn man eine Moduldatei mittels des “maxflash.pl”-Skriptes erzeugt erhält man eine “.bin” und/oder eine “.car” Datei.

- “.bin” Dateien sind Rohdaten einer Cartridge.

7 EMULATOR UNTERSTÜTZUNG

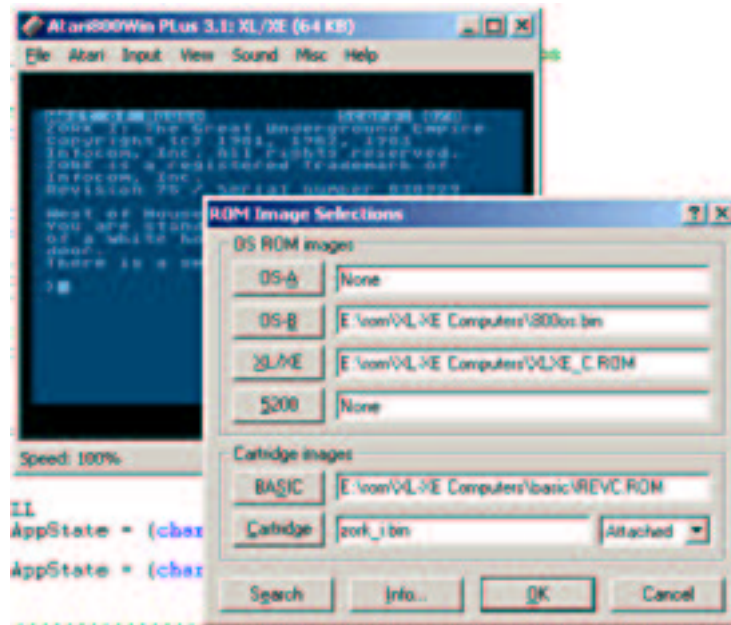


Abbildung 11: Atari800Win Einstellungen

- “.car” Dateien sind Atari800Win 'CART' Abbilder, die die Rohdaten, eine Identifikation (was für ein Cartridge Typ), sowie eine Kontrollsumme enthalten.

Zum Laden eines “.bin” oder “.car” Abbildes wird im Menüpunkt 'Atari' das Untermenü 'ROM Images' gewählt. Im erscheinenden Dialog muß auf 'Cartridge' gedrückt werden und anschließend die gewünschte Datei im Dateiauswahldialog gewählt werden.

Falls ein “.car” Abbild gewählt wurde, erkennt Atari800Win die Korrekte Größe und den Typ des Moduls alleine. Bei einem “.bin” Abbild wird man nach dem korrekten Typ des Moduls gefragt.

Rechts in der Auswahlbox für die Cartridges ist ein Klappmenü, in dem man die Cartridge de-/aktivieren kann. Um eine Cartridge in den Emulator "einzustecken", muß man hier auf "attached" stellen. "Detached" steht hier für ein nicht eingestecktes Modul (Cartridge).

Um ein “.bin” Abbild einer 1Mbit Flash Cartridge einzulegen, muß im Klappmenü "Atarimax 1Mb Flash Cartridge" ausgewählt werden. Nach Drücken auf "Ok" gelangt man ins Emulatorfenster zurück.

7.7.3 Starten des Moduls

In der Standardkonfiguration startet der Atari800WinPlus sofort, wenn ein Modul eingelegt wurde, neu.

7 EMULATOR UNTERSTÜTZUNG

Es ist möglich, daß ein Kaltstart von Nöten ist, um ein Modul zu starten. Falls nach dem Einlagen eines Moduls nichts passiert, ist er daher einfach mittels Shift+F5 hervorzurufen.

Falls nichts, außer einem weißen Cursor oder einem Ton, nach Starten des Emulators kommt, sollte man überprüfen, ob man auch den SIO-Patch ausgeschaltet hat.

8 Konvertierungsanleitung

Übersetzt von Marc Brings

Anmerkung des Übersetzers:

Ich habe die im folgenden Kapitel beschriebenen Arbeiten noch nicht selbst durchgeführt. Bei einigen Absätzen war ich mir nicht ganz sicher was Steve gemeint hat, deshalb befindet sich an einigen Stellen der englische Originaltext hinter der Übersetzung.

Marc Brings

8.1 Einleitung

Dieses Kapitel setzt Erfahrung im Umgang mit 6502-Assembler, Emulatoren und Debuggern voraus.

Da es schon eine große Anzahl von auf Modul umgesetzten ATARI-Spielen gibt, sollte vor der aufwendigen Arbeit einer individuellen Anpassung erst geforscht werden, ob diese Software schon an die AtariMax Falsh Cartridge angepasst wurde.

Während viele, wenn nicht die meisten, Spiele mit der -DISKPACKER-Funktion zusammenarbeiten, müssen einige Spiele modifiziert werden, die PORT B reinitialisieren oder andere ungewöhnliche Abfragen durchführen.

Dies ist keine umfassende Anweisung zur Erstellung einer individuellen Anpassung, sondern nur ein paar Beispiele um Anhaltspunkte zu geben, wie eine individuelle Anpassung vorgenommen werden kann.

Wenn Sie nicht vorhaben eine individuelle Anpassung zu erstellen und eine Menge 6502-Assembler zu schreiben, und das nur aus Spaß an der Freude,so können Sie dieses Kapitel überspringen.

8.2 Änderungen an Black Lamp

Dieses Beispiel benutzt das Blacklamp2.atr-Disk-Abbild des Jellystone Park Archive. Konvertieren Sie die Disk mit der "-DISKPACKER"-Funktion des "maxflash.pl"-Scriptes zu einem Modul-Abbild.

```
>perl maxflash.pl -diskpacker BlackLamp2.atr test.bin
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atariimax.com/
New binary image [test.bin] created.
```

Eine normale automatische Übersetzung, für dieses Spiel werden keine speziellen Funktionen benötigt.

Starten Sie den Emulator und laden Sie das gerade erzeugte "test.bin"-Abbild. Wenn Sie Hilfe beim Erstellen oder Laden des Abbildes benötigen, siehe Kapitel 5: Benutzung der Maxflash-Software, oder Kapitel 7: Emulator-Unterstützung, bevor Sie weiter vorgehen. Wenn der Emulator läuft, wird ein schwarz/gelb gestreifter Bildschirm angezeigt; das Spiel bootet nicht.

Da es sich bei dem Emulator nicht um einen echten ATARI handelt, können wir nachsehen was schiefgelaufen ist in dem wir den Emulator stoppen und den eingebauten Monitor/Debugger starten.

8 KONVERTIERUNGSANLEITUNG

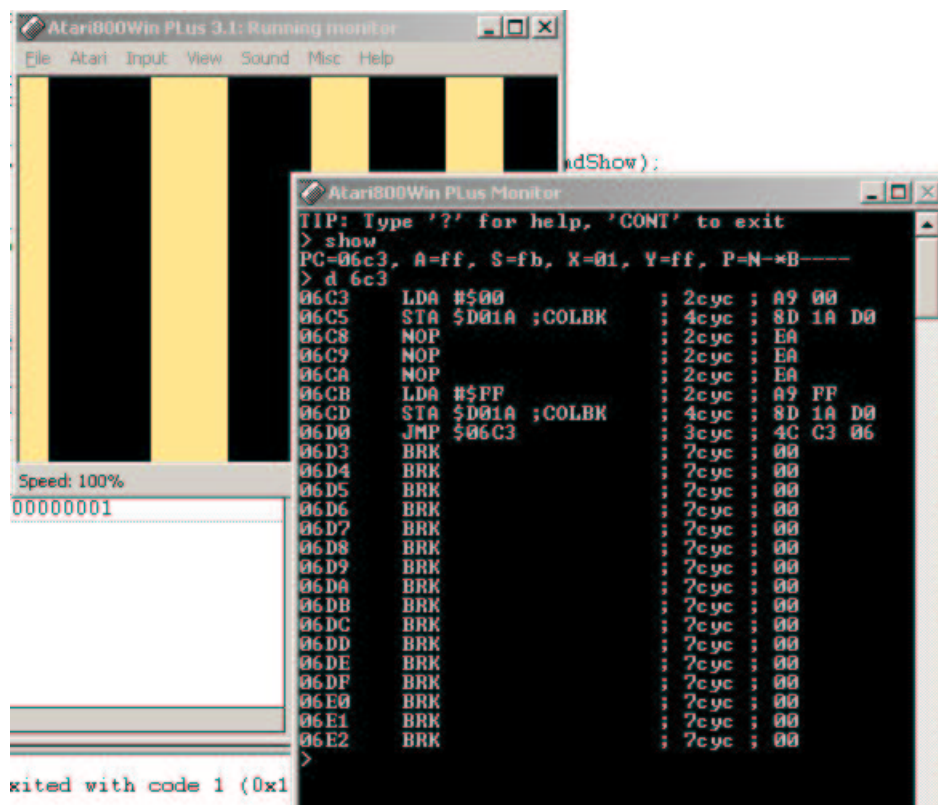


Abbildung 12: Black Lamp Absturz

8 KONVERTIERUNGSANLEITUNG

Starten Sie den Monitor mit [F8], geben Sie *show* ein um den aktuellem Instruktions-Zeiger (Instruction Pointer, IP) anzuzeigen zu lassen, und letztendlich *d 6c3* um den Code anzuzeigen, an dem die Emulation beendet wurde. Diese Vorgänge werden in der Abbildung 12 gezeigt.

Es sieht aus, als wenn das Programm in einer Schleife, die die farbigen Balken erzeugt, festhängt.

An 0x600 ist üblicherweise das Boot-Programm zu finden; geben Sie *d 600* ein und überprüfen Sie einige der folgenden Seiten Programm um zu sehen, was uns hierher gebracht haben könnte.

```
0635 CLC ; 2cyc ; 18
0636 ADC #S80 ; 2cyc ; 69 80
0638 STA $0304 ;DBUFLO ; 4cyc ; 8D 04 03
063B LDA $0305 ;DBUFHI ; 4cyc ; AD 05 03
063E ADC #S00 ; 2cyc ; 69 00
0640 STA $0305 ;DBUFHI ; 4cyc ; 8D 05 03
0643 LDA $030A ;DAUX1 ; 4cyc ; AD 0A 03
0646 CLC ; 2cyc ; 18
0647 ADC #S01 ; 2cyc ; 69 01
0649 STA $030A ;DAUX1 ; 4cyc ; 8D 0A 03
064C LDA $030B ;DAUX2 ; 4cyc ; AD 0B 03
064F ADC #S00 ; 2cyc ; 69 00
0651 STA $030B ;DAUX2 ; 4cyc ; 8D 0B 03
0654 DEC $F0 ;FCHRFLG ; 5cyc ; C6 F0
0656 LDA $F0 ;FCHRFLG ; 3cyc ; A5 F0
0658 BNE $062F ; 2cyc ; D0 D5
065A RTS ; 6cyc ; 60
065B LDA #S00 ; 2cyc ; A9 00
065D STA $022F ;SDMCTL ; 4cyc ; 8D 2F 02
0660 LDA $D301 ;PORTB ; 4cyc ; AD 01 D3 <----- Hier wird überprüft ob sich das OS im ROM befindet
0663 CMP #SFF ; 2cyc ; C9 FF <----- Das ist nicht der Fall wenn von Modul geladen wird, dieses
0665 BNE $06C3 ; 2cyc ; D0 5C <----- LDA wird FE und nicht FF zurückmelden.
0667 JSR $0608 ; 6cyc ; 20 08 06
066A JSR $A020 ; 6cyc ; 20 20 A0
```

In Adresse 0x665 ist der Sprung zu den farbigen Streifen zu finden.

Wenn Sie die automatische Anpassung der “-DISKPACKER”-Funktion benutzten, wird das Lesen von PORT B durch ein ATARI-Programm anstelle des 0xFF ein 0xFE zurückmelden, weil das Flash-Modul das OS im RAM plazierte hat.

Jedes Programm, das PORT B überprüft oder beschreibt, kann ein solches Problem verursachen. Einige Spiele re-initialisieren PORT B einfach um sich abzusichern, andere vergewissern sich, wie oben zu sehen ist, daß kein ROM-Monitor installiert ist. Wir können diesen Test oder das Schreiben einfach umgehen nachdem PORT B bereits initialisiert ist.

Umgehen wir den Test und sehen, ob das Programm booten wird.

Um dies zu tun, müssen wir den Emulator stoppen BEVOR das Programm den Test ausführt und das Ergebniss des PORT B-Test ändern.

Zuerst setzen wir einen Break-Punkt an 0x660, einige Befehle vor dem Test. Geben Sie *break 660* ein und drücken Sie [ENTER].

Wenn der Break-Punkt gesetzt wurde, muß der Emulator neu gestartet werden. Geben Sie *cont* ein um zum Emulator zurückzukehren und drücken Sie [Shift][F5] um den Emulator neu zu starten.

Wenn der Emulator den Break-Punkt erreicht, wird er einen Dialog ausgeben in dem er mitteilt, daß sich der ATARI aufgehängt hat. Klicke auf YES um den Debugger zu starten.

Sie sollten nun den oben abgebildete Schirmbild sehen, zumindest aber einen teilweise schwarzen Schirm.

8 KONVERTIERUNGSANLEITUNG

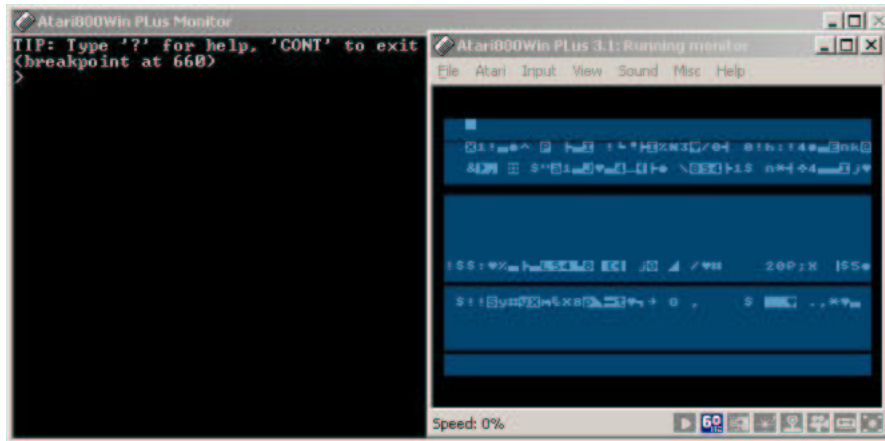


Abbildung 13: Breakpoint an Speicherstelle \$660

Wir haben unmittelbar vor dem PORT B-Test angehalten, wir schauen uns das Programm an der aktuellen IP an. Geben Sie zum Disassemblieren *d 660* ein.

```
TIP: Type '?' for help, 'CONT' to exit
(breakpoint at 660)
> d 660
0660 LDA $D301 ;PORTB ; 4cyc ; AD 01 D3
0663 CMP #$FF ; 2cyc ; C9 FF
0665 BNE $06C3 ; 2cyc ; D0 5C --- Hier ist es wieder.
0667 JSR $0608 ; 6cyc ; 20 08 06
066A JSR $A020 ; 6cyc ; 20 20 A0
066D LDA $14 ;RTCLOCK+2 ; 3cyc ; A5 14
066F CMP $14 ;RTCLOCK+2 ; 3cyc ; C5 14
0671 BEQ $066F ; 2cyc ; F0 FC
0673 LDA #$2A ; 2cyc ; A9 2A
0675 STA $F0 ;FCHRF LG ; 3cyc ; 85 F0
0677 LDA #$07 ; 2cyc ; A9 07
0679 STA $0621 ; 4cyc ; 8D 21 06
067C LDA #$49 ; 2cyc ; A9 49
067E STA $0626 ; 4cyc ; 8D 26 06
0681 LDA #$00 ; 2cyc ; A9 00
0683 STA $062B ; 4cyc ; 8D 2B 06
0686 JSR $060C ; 6cyc ; 20 0C 06
0689 LDA #$FE ; 2cyc ; A9 FE
068B STA $F0 ;FCHRF LG ; 3cyc ; 85 F0
068D LDA #$1C ; 2cyc ; A9 1C
068F STA $0621 ; 4cyc ; 8D 21 06
0692 LDA #$73 ; 2cyc ; A9 73
0694 STA $0626 ; 4cyc ; 8D 26 06
0697 LDA #$00 ; 2cyc ; A9 00
> a 665
Simple assembler (enter empty line to exit)
665 : nop
666 : nop
667 :
> cont
```

Wir können den Test in 0x665 einfach umgehen indem wir den Sprungbefehl (BNE) entfernen.

Geben Sie *a 665* ein um den Befehl zu überschreiben, dann fügen Sie zwei NOP-Befehle, welche die zwei von der BNE-Anweisung benötigten Byte überschreiben, ein.

Geben Sie *cont* ein um zu sehen ob die Änderungen nun laufen.

Nach einigen Sekunden Verzögerung beim ATARI-Logo-Screen lädt das Programm weiter. Ein kleines Testspiel wird uns bestätigen daß das Spiel nun auf dem Modul läuft.

Jetzt, wo wir wissen was zu ändern ist, müssen wir die "test.bin"-Datei so modifizieren,

8 KONVERTIERUNGSANLEITUNG

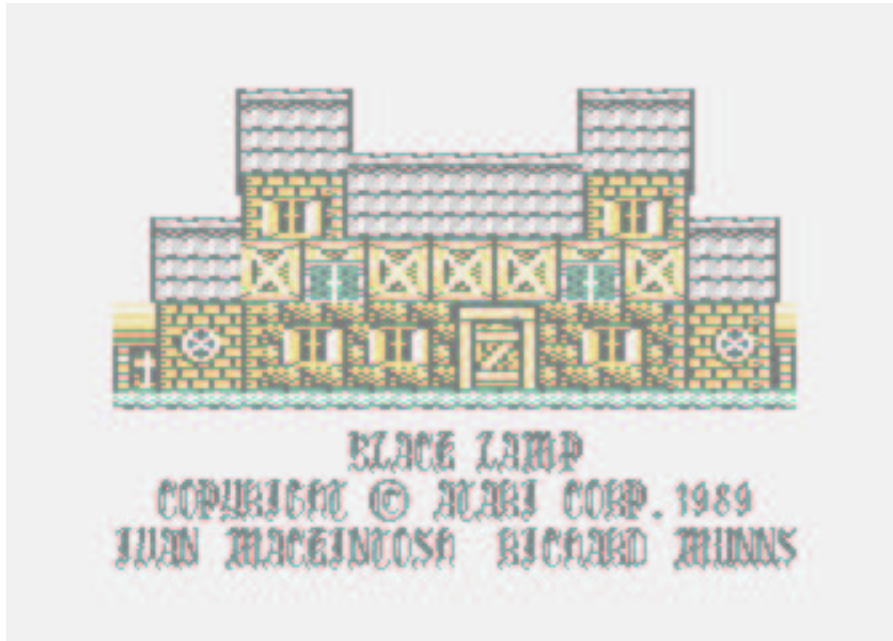


Abbildung 14: Gepatches Black Lamp

daß die Änderung ein Teil des Images ist und auf das Modul geschrieben werden kann.

So sieht das Programm vor der Änderung aus:

```
0660 LDA $D301 ;PORTB ; 4cyc ; AD 01 D3
0663 CMP #$FF ; 2cyc ; C9 FF
0665 BNE $06C3 ; 2cyc ; D0 5C <---Hier ist es wieder.
0667 JSR $0608 ; 6cyc ; 20 08 06
```

Und nach der Änderung:

```
0660 LDA $D301 ;PORTB ; 4cyc ; AD 01 D3
0663 CMP #$FF ; 2cyc ; C9 FF
0665 NOP ; 2cyc ; EA <--- Unsere Aenderungen.
0666 NOP ; 2cyc ; EA
0667 JSR $0608 ; 6cyc ; 20 08 06
```

Im Disassembler des Emulators sieht man in der Spalte ganz rechts für jeden Befehl den aktuellen Assembler-OP-Code in Hex-Format. Letztendlich wurden nur zwei Byte geändert. Die Bytes 0xD0 0X5C wurden zu 0xEA 0xEA, NOP-Befehlen.

Starten Sie einen Hex-Editor und öffnen Sie das "test.bin"-Modul-Abbild. Suche den Bereich im Original-Programm in der Nähe der zu ändernden Stelle, 0xC9 0xFF 0xD0 0x5C 0x20. Überschreiben Sie einfach die Bytes 0xD0 0x5C, den oben gezeigten BNE-Befehl, mit 0xEA 0xEA und speichern Sie das Abbild wieder.

Die Datei ist nun geändert und kann auf dem Modul gespeichert werden. Benutzen Sie hierfür die "-BIN2ALL"-Funktion des "maxflash.pl"-Scriptes um ein ".car"-Abbild zum Testen und ein ".atr"-Disk-Abbild zum Programmieren des Modules zu erzeugen.

8 KONVERTIERUNGSANLEITUNG

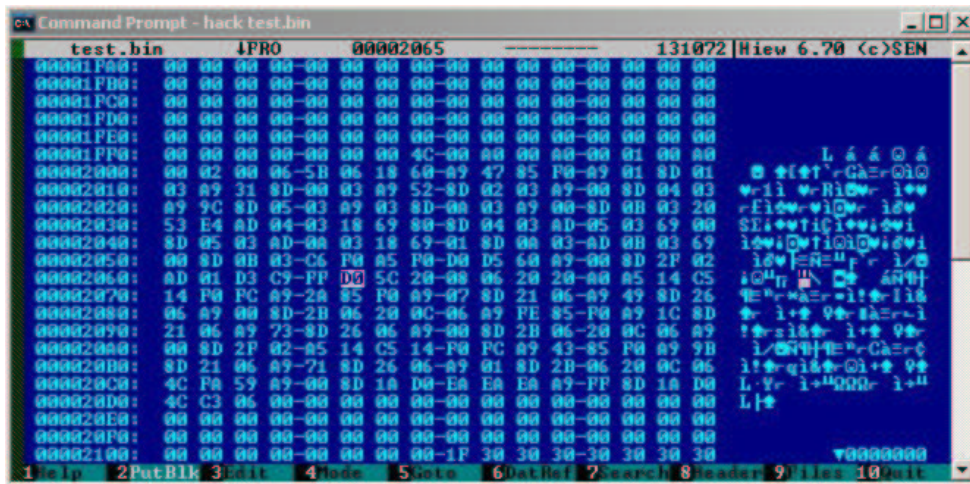


Abbildung 15: Black Lamp Abbild im Hex-Editor

8.3 Änderungen an Zorro

Konvertieren Sie Zorro mit dem “maxflash.pl”-Script zu einem Modul-Abbild ohne besondere Funktionen zu benutzen.

```
>perl maxflash.pl -diskpacker zorro.atr zorro.bin
MAXFLASH Multicart Build/Utility Script - c2003 Steven J Tucker
Information and updates at http://www.atarimax.com/
New binary image [zorro.bin] created.
```

Testen Sie das Abbild mit dem ATARI800Win Emulator.

Ups, er hängt sich auf. Beachten Sie das unten im Debugger-Fenster zu lesen ist daß das OS zurück zum ROM gewechselt wurde. Da ein “-DISKPACKER”-Modul das OS bearbeitet, muß es im RAM verbleiben. Dieses Problem müssen wir beheben.

Das OS wird durch Schreiben einer 1 nach Bit 0 von PORT B zurück in das ROM gelegt. Wir können nach einem Schreibzugriff auf PORT B suchen um herauszufinden wodurch das Problem verursacht wird. Starten Sie den Monitor mit YES und suche mit *s 0 c000 8d 01 d3* nach einem Schreibzugriff auf PORT B.

Normalerweise wird man so den störenden Programmteil finden, und das Ändern ist kein Problem. Aber in diesem Fall wird man nichts finden. Da das Problem unmittelbar nach dem Start des Bootvorganges auftritt, überprüfen wir einfach das Bootprogramm und sehen nach, ob dort der Übeltäter zu finden ist. Setzen Sie einen Break-Punkt auf den SIO-Vektor 0xE459 und drücken Sie [Shift][F5], um den Emulator neu zu starten.

Wenn der Bootvorgang startet, wird ATARI800Win am Break-Punkt anhalten und Sie auffordern den Monitor zu öffnen. Klicken Sie YES an. Im Monitor können wir uns die SIO-Register ansehen um zu sehen, was geladen wurde. Geben Sie *d 300* ein um dir die Register anzeigen zu lassen.

Der erste Aktion ist lediglich ein Statusabfrage (0x53 an Speicherstelle 0x0302), geben Sie also *cont* ein um weiterzumachen. Der Break-Punkt wird sofort wieder angefahren. Klicken Sie YES um in den Monitor zu gelangen.

8 KONVERTIERUNGSANLEITUNG

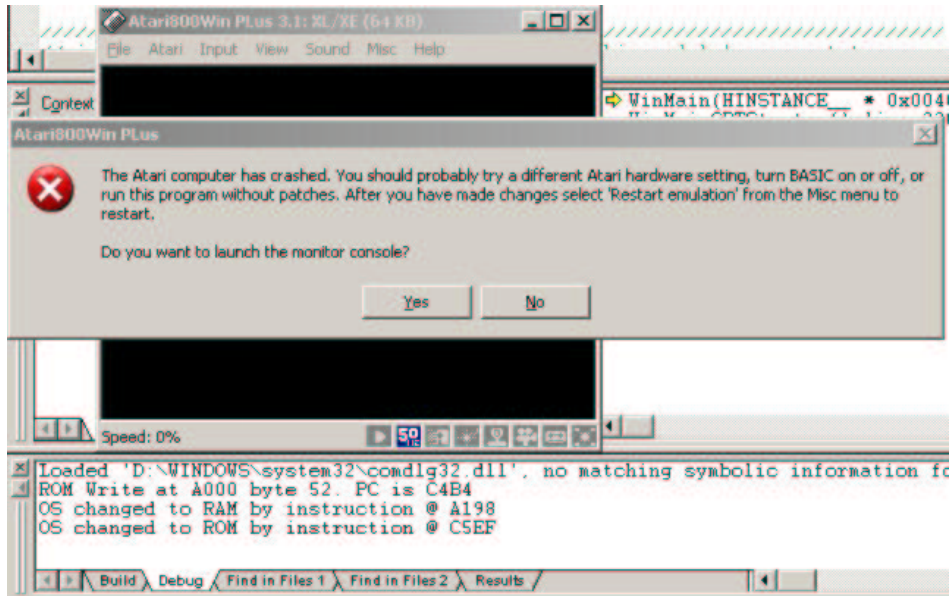


Abbildung 16: Zorro Programmabsturz

Zeigen Sie den Speicherinhalt an 0x300 an.

```
TIP: Type '?' for help, 'CONT' to exit
(breakpoint at CC00)
> d 300
0300 AND ($01),Y ;NGFLAG ; 5cyc ; 31 01
0302 CIM ; 2cyc ; 52
0303 RTI ; 6cyc ; 40
0304 BRK ; 7cyc ; 00 <-- Zielbuffer für den
0305 NOP $07 ;CMCMD ; 3cyc ; 04 07 <-- Diskettenzugriff, 0x0400
0307 BRK ; 7cyc ; 00
0308 NOP #$00 ; 2cyc ; 80 00
030A ORA ($00,X) ; 6cyc ; 01 00
030C BRK ; 7cyc ; 00
030D BRK ; 7cyc ; 00
```

Das XL/XE-OS lädt das Bootprogramm immer nach 0x0400. Wir haben das OS den Sektor bisher noch nicht laden lassen, also geben Sie *cont* ein und gehen Sie, wenn der Break-Punkt erneut erreicht wird, wieder in den Monitor.

Zeigen Sie den Speicher ab 0x0400 an, das ist der erste Bootsektor.

```
> d 400
0400 BRK ; 7cyc ; 00
0401 ASO ($00,X) ; 8cyc ; 03 00 <-- Zu ladender Sektor (Offset +1)
0403 BCS $0405 ; 2cyc ; B0 00 <-- Zieladresse (Offset +2/+3)
0405 BMI $03B0 ;B7-ICHID ; 2cyc ; 30 A9
0407 BRK ; 7cyc ; 00
0408 STA $0A ;DOSVEC ; 3cyc ; 85 0A
040A LDA #$30 ; 2cyc ; A9 30
```

Der erste Sektor sagt dem OS daß es 3 Sektoren an Adresse 0xB000 laden soll. Setzen Sie den Break-Punkt an Adresse 0xB008 und starte den Emulator mit [Shift][F5] neu. Wurde der Break-Punkt erreicht, geben Sie *d d008* ein um das Programm an der aktuellen IP zu disassemblieren.

8 KONVERTIERUNGSANLEITUNG

```
TIP: Type '?' for help, 'CONT' to exit
(breakpoint at B008)
> d b008
B008 STA $0A ;DOSVEC ; 3cyc ; 85 0A
B00A LDA #$30 ; 2cyc ; A9 30
B00C STA $0B ;DOSVEC+1 ; 3cyc ; 85 0B
B00E LDA #$FF ; 2cyc ; A9 FF
B010 STA $D301 ;PORTB ; 4cyc ; 8D 01 D3 <--Setzt das OS in das ROM
B013 LDX #$01 ; 2cyc ; A2 01
B015 STX $D508 ; 4cyc ; 8E 08 D5 <--Ups
B018 DEX ; 2cyc ; CA
B019 STX $0244 ;COLDST ; 4cyc ; 8E 44 02
B01C LDA #$29 ; 2cyc ; A9 29
B01E STA $BFFA ; 4cyc ; 8D FA BF
B021 LDA #$B0 ; 2cyc ; A9 B0
```

Da ist es, hier wird PORT B auf 0xFF gesetzt, was das OS zurück in das ROM legt. Wir ändern einfach den Wert der vorhergehenden Anweisung zu 0xFE um das OS im RAM zu belassen.

Ebenfalls interessant ist die Anweisung STA \$D508. Sie würde ein Spartados X-Modul abschalten, ein Flash-Modul jedoch einschalten, so daß wir diese Anweisung durch Überschreiben mit NOP-Befehlen ebenfalls entfernen.

Führen Sie die folgenden Anweisungen aus um die benötigten Änderungen durchzuführen.

```
> a b00e
Simple assembler (enter empty line to exit)
B00E : lda #$FE
B010 :
> a b015
Simple assembler (enter empty line to exit)
B015 : nop
B016 : nop
B017 : nop
B018 :
>
```

Das Ergebnis sieht wie folgt aus:

```
B008 STA $0A ;DOSVEC ; 3cyc ; 85 0A
B00A LDA #$30 ; 2cyc ; A9 30
B00C STA $0B ;DOSVEC+1 ; 3cyc ; 85 0B
B00E LDA #$FE ; 2cyc ; A9 FE
B010 STA $D301 ;PORTB ; 4cyc ; 8D 01 D3
B013 LDX #$01 ; 2cyc ; A2 01
B015 NOP ; 2cyc ; 1A
B016 NOP ; 2cyc ; 1A
B017 NOP ; 2cyc ; 1A
B018 DEX ; 2cyc ; CA
B019 STX $0244 ;COLDST ; 4cyc ; 8E 44 02
B01C LDA #$29 ; 2cyc ; A9 29
B01E STA $BFFA ; 4cyc ; 8D FA BF
B021 LDA #$B0 ; 2cyc ; A9 B0
```

Geben Sie *cont* ein und testen Sie die Änderungen. Das Ergebnis ist ein perfekt laufendes Zorro-Modul, und alles was wir jetzt noch zu tun haben ist die *zorro.bin*-Datei in einen HEX-Editor zu laden, die Änderungen durchzuführen, und wir sind fertig.

8.4 Größere Änderungen an MULE

Die “-DISPACKER”-Funktion arbeitet wunderbar mit MULE zusammen, aber wenn das MULE-Modul auf einem ATARI 800 genau so laufen soll wie auf einem XL/XE, sind einige größere Eingriffe nötig.

MULE ist ein gutes Spiel zum Üben einer Umsetzung. Es ist ein mehrteiliges, diskettenbasierendes Spiel mit genügend Komplexität um eine kleine Herausforderung zu sein, aber noch einfach genug um es in ein paar Stunden zu schaffen.

Es gibt einige Wege eine individuelle Anpassung (custom conversion) wie diese zu erstellen. Der hier vorgestellte ist nur ein Weg es zu tun.



Abbildung 17: M.U.L.E.

Wenn Sie den Diskettenzugriff und die Sektornummeranzeige des Emulators nicht eingeschaltet haben, aktivieren Sie diese nun - wir benötigen diese Informationen um den Ladevorgang von MULE in den Speicher zu beobachten. Sie findest diese Einstellungen im Atari800Win-Emulator im Menü `view*->*show`. Deaktivieren Sie den SIO-Patch. Das Spiel soll langsam laden damit wir den Sektornummerzähler beobachten können. Stellen Sie den Computertyp auf XL/XE ein und den Speicher auf 64k.

Es sind eine Menge verschiedener Versionen von MULE im Umlauf; manche booten nicht auf einem XL/XE, einige booten nur auf einem XL/XE, etc. Die hier verwendete Version von Mule kann als ATR-Abbild geladen werden.

Um anzufangen, laden Sie das `mule.atr`-Disk-Abbild in den Emulator und starten Sie den Emulator wie Sie normalerweise ein Diskettenspiel starten. Nachdem das Spiel geladen wurde drücken Sie [RESET]. Das Spiel startet neu ohne zu laden. Da das erste Segment des Spieles resetfest ist wird uns die Arbeit enorm vereinfacht.

Alles was wir zu tun haben ist herauszufinden wo sich das Programm im Speicher befindet und diesen Bereich auf Disk zu speichern. Starten Sie den Monitor mit [F8]. Der Warmstartvektor befindet sich bei XL/XE-Modellen in Adresse 0x000A. Betrachten Sie also diese Speicherstelle um herauszufinden ab welcher Adresse Mule sich selber startet.

```
TIP: Type '?' for help, 'CONT' to exit
> d a
000A NOP ; 2cyc ; 5A
000B SAX $40 ;FREQ ; 3cyc ; 87
```

Geben Sie `d a` ein um den Inhalt der Speicherstelle 0x000A anzeigen zu lassen. Die 16-Bit-Adresse ist im LSB/MSB-Format in zwei 8-Bit-Speicherstellen gespeichert. Der

8 KONVERTIERUNGSANLEITUNG

Warmstartvektor zeigt bei Mule nach 0x875A. Notiere Sie diese Information das; später brauchen wir es. Jetzt müssen wir herausfinden wie lang das Programm ist das Mule beim Start lädt und wo es gespeichert wird.

Für ein Spiel ist eine Menge Platz in einem Flash-Modul. Wenn wir faul sind können wir einfach alles hineipacken, aber wir wollen ein wenig selektieren.

Booten Sie MULE neu und beobachte die Sektoranzeige während das Programm lädt. Sie werden sehen daß die Sektoren 1-132 geladen werden bevor das Titelbild erscheint. Ein wenig Mathematik sagt uns das 132 Sektoren zu je 128 Byte, also ungefähr 16k oder 0x4200 Bytes von der Disk geladen werden.

Jetzt können wir durch disassemblieren des Bootsektors herausfinden wo diese Informationen im Speicher abgelegt werden. Wahrscheinlich lädt das Spiel lediglich einen großen Block in den Speicher und startet ihn, so daß wir nach dem Erscheinen des Spielmenüs einfach etwas im Speicher herumstochern können um zu sehen, was geschrieben wurde.

Starten Sie den Emulator neu und wenn der MULE-Titelbildschirm erscheint, starten Sie den Monitor. Geben Sie *d 0* ein um den Bereich ab 0x0000 zu disassemblieren und drücken Sie [Enter]. Halten Sie [Enter] gedrückt und es wird der gesamte Inhalt des Speichers bis 0xFFFF, der letzten Adresse, langsam angezeigt.

Da das OS die meisten Adressen unter 0x0500 selbst benutzt, und alles über 0xC000 zum OS-ROM gehört, müßte jeder große Block von 6502-Befehlen, den wir außerhalb dieses Bereiches finden, das MULE-Menü sein.

Nachdem Sie etwas herumgestöbert haben, werden Sie feststellen daß der Speicher größtenteils leer ist, abgesehen von einem großen Block von 0x7000 bis 0xBFFF, welcher 0x4FFF Bytes groß ist. Ziemlich nah an den 0x4200 die ursprünglich geladen wurden.

Laß uns unsere Theorie wie MULE lädt testen indem wir diesen Teil von Hand in den Speicher laden ohne MULE auf die Disk zugreifen zu lassen.

Schreibe diesen Bereich des Speichers zuerst mit Hilfe des Monitor-Befehles WRITE in ein Binärfile. *WRITE 7000 BFFF C:\test.bin* zum Beispiel würde den Inhalt des ATARI-Speichers von 0x7000 bis 0xBFFF in eine Datei namens C:\test.bin schreiben.

Jetzt, wo wir den Bereich, der uns interessiert, gespeichert haben, entfernen Sie das mule.atr-Disk-Abbild aus Laufwerk #1 und starten die Emulation mit [Shift][F5] neu.

Sobald der ATARI800Win startet, drücken Sie [F8]. Machen Sie das so schnell wie möglich, so daß wir den Rechner unter Kontrolle bekommen bevor er in den Selbstest-modus geht.

Wenn der Monitor auftaucht, laden wir den Speicherbereich, den wir eben gesichert haben, mit Hilfe des READ-Befehl zurück in den ATARI. *READ C:\test.bin 7000 FFFF* lädt den gesamten Inhalt von C:\test.bin in den Speicher des ATARIs, beginnend an Adresse 0x7000.

Nachdem wir den MULE-Speicher-Dump neu geladen haben, müssen wir den ATARI dazu bringen, daß er denkt er habe bereits gebootet, deshalb geben wir *c 244 0* und *c 8 0* ein um einen Warmstart anstelle eines Kaltstartes zu erzeugen.

Jetzt müssen wir den Warmstartvektor, den wir uns vorhin notiert haben, von Hand setzen: geben Sie *c a 5a 87* ein. Dadurch wird aus 0x000A 0x5A, und 0x000B wird zu 0x87.

8 KONVERTIERUNGSANLEITUNG

Jetzt testen wir das ganze indem wir einen Warmstart auslösen. Wir können dies einfach durch Setzen des Programmzählers auf die Warmstartroutine des OS, 0xE474 tun. Gebe im Monitor *setpc e474* ein und dann *cont* um den Warmstart ausführen zu lassen.

Es klappt, MULE erscheint ohne von Disk zu laden. Mit dem “generic memory unpacker”-Programm im “mempack.a65” können wir beginnen ein Modul zu erstellen.

Zuerst müssen wir berechnen wie viele Bänke der erste Teil von Mule im Modul belegt. Wir haben 0x4FFF Bytes Programm, und jede Bank im Modul fasst 0x2000 Bytes, also benötigen wir drei Bänke. Ein wenig Platz wird verschwendet, aber das ist nicht wirklich ein Problem da außer wenn es später eng werden sollte.

Mit dem generischen Speicher-Entpacker (generic memory unpacker) Programm des *mempack.a65* können wir einen individuellen Entpacker (custom memory unpacker) für unseren MULE Teil 1 Dump erstellen. Wir benutzen Bank 0 des Modules für unseren Memory Entpacker, und die Bänke 1 bis 3 für unseren MULE Speicher Dump von Teil 1.

Die Quelldateien, die in diesem Beispiel verwendet werden, können auf Der AtariMax Internetseite⁷ heruntergeladen werden. Das Teil 1-Entpacker-Programm nennt sich *mule.a65*.

Nach einer Anpassung das nun 3 Bänke anstatt 8 entpackt werden, erstellen wir aus *mempack.a65* das angepasste *mule.a65*. Mit dem Assembler (TASM) erzeugen wir das Unpacker Programm für Bank 0. Da der Unpacker eine volle Bank benutzt (das macht uns die Sache leichter), füllen wir das ausführbare Programm bis auf eine Größe von 8192 Bytes auf.

Vergrößern Sie den MULE Speicherabzug auf die Größe von exakt 3 Bänken, $8192 \times 3 = 24576$ Bytes. Das ist eine einfach zu verarbeitende Größe (bezogen auf die Größe der Modulbänke). Kopieren Sie die zwei Dateien mit Hilfe des DOS-Copy-Befehles zusammen, zum Beispiel mit *copy /b mule.com+test.bin testcart.bin*.

Vergrößern Sie das resultierende Modul-Image auf die Größe eines unbearbeitetem 1Mb Modul, 131.072 Bytes. Jetzt können Sie das Entpackerprogramm und den Speicherdump einfach testen indem Sie die “testcart.bin”-Datei als Modul-Image in den Emulator laden.

Es funktioniert wunderbar, Stufe 1 ist erledigt!

Entfernen Sie nun das Modul-Image und führen Sie einen Reset des Emulators durch um *mule.atr* von Diskette zu laden. Laden Sie MULE und starten Sie ein Spiel im Anfänger-Modus.

Wenn Sie MULE eine Weile spielen, werden Sie feststellen das es zweimal von Diskette lädt. Einmal wenn es das Titelbild geladen wird, und ein zweites mal wenn es das eigentliche Spiel lädt. Alles was wir jetzt noch zu tun haben ist einen Dump vom zweiten Teil zu erstellen und wir haben ein funktionierendes MULE-Modul!

Das ist etwas trickreicher; wir können uns die Arbeit nicht mit dem RESET-Trick vereinfachen. Laden Sie MULE erneut und bereiten Sie den Start des Spieles vor. Wenn Sie das Spiel gestartet haben, beobachten Sie die Sektornummeranzeige während der zweite Teil lädt.

⁷http://www.atarimax.com/flashcart/documentation/mulecart_tutorial_source.zip

8 KONVERTIERUNGSANLEITUNG

Es sieht aus als wenn die Sektoren 370-690 geladen werden um das eigentliche Spiel zu starten. Das sind 0xA000 Byte, das paßt locker in den restlichen freien Platz im Modul. Was wir jetzt natürlich wissen müssen ist, wohin das Programm geladen wird. Das erfordert ein wenig mehr Aufwand als den Trick mit dem RESET, wird aber nicht zu mühsam.

Wir wissen das Mule von Disk lädt, also ist der einfachste Weg herauszufinden wohin es geladen wird sich das Ladeprogramm anzusehen. Starte im MULE-Titelbild den Monitor und suche nach einem JSR zum OS SIO-Vektor. Der OS SIO-Vektor ist 0xE459 und der JSR-OP-Code ist 0x20, also suche im Speicher nach 0x20 0x59 0xE4.

```
> s 0 ffff 20 53 e4
Found at babb
Found at bb14
Found at c5a2
```

Der Treffer über 0xC00 liegt im OS, also ignorieren wir diese. Wir haben zwei Treffer in dem Bereich, indem das Programm vorhin entpackt wurde, beide so um die Adresse 0xbab0. Wenn Sie etwas in diesem Bereich herumstöberst, werden Sie den Anfang einer Diskettenleseroutine finden.

```
BAF0 STA $0A ;DOSVEC ; 3cyc ; 85 0A
BAF2 LDA #$E4 ; 2cyc ; A9 E4
BAF4 STA $0B ;DOSVEC+1 ; 3cyc ; 85 0B
BAF6 LDA #$01 ; 2cyc ; A9 01 <-- Vorberitung eines SIO Lesebefehls 0xBAF6
BAF8 STA $0301 ;DUNIT ; 4cyc ; 8D 01 03
BAFB LDA #$52 ; 2cyc ; A9 52
BAFD STA $0302 ;DCOMND ; 4cyc ; 8D 02 03
BB00 LDA #$00 ; 2cyc ; A9 00 <-- Zieladresse des Ladebefehls.
BB02 STA $0304 ;DBUFLO ; 4cyc ; 8D 04 03
BB05 LDA #$10 ; 2cyc ; A9 10
BB07 STA $0305 ;DBUFHI ; 4cyc ; 8D 05 03
BB0A LDA #$90 ; 2cyc ; A9 90
BB0C STA $030A ;DAUX1 ; 4cyc ; 8D 0A 03
BB0F LDA #$01 ; 2cyc ; A9 01
BB11 STA $030B ;DAUX2 ; 4cyc ; 8D 0B 03
BB14 JSR $E453 ;DSKINV ; 6cyc ; 20 53 E4 <-- OS Diskettenroutine starten
```

Das sieht schon gut aus. Wir müssen uns jedoch sicher sein daß dies das Programm ist, welches aufgerufen wird, wenn das Laden des Spieles beginnt. Setzen Sie einen Break-Punkt an die Adresse 0xBAF6, starten Sie die Emulation mit [Shift][F5] neu und beginnen Sie ein Spiel.

Wenn wir Recht haben, erreicht der Emulator den Break-Punkt unmittelbar bevor MULE versucht den zweiten Teil von Disk zu laden. Er tut es, also ist das definitiv die Stelle an der der Lesevorgang für Stufe zwei erfolgt. Wenn Sie sich das disassemblierte Programm oben anschauen, werden Sie feststellen daß das Pogramm den Speicherzeiger für den Diskettenzugriff, 0x0304, auf die Adresse 0x1000 legt.

Das könnte bedeuten daß das Programm das Spiel ab Adresse 0x1000 in den Speicher lädt. Vom Beobachten des Ladevorganges her wissen wir noch daß 0xA000 Bytes von Disk geladen werden. Wir können also versuchen einen Dump des zweiten Teiles von 0x1000-0xB000 (insgesamt 0xA000 Bytes), auf Disk zu erstellen.

Starten Sie den Emulator neu und beginnen Sie eine Partie MULE im Anfängermodus. Wenn Teil zwei einmal vollständig geladen ist und die Nachricht "Now landing on Planet Irata" erscheint drücken Sie [F8] um den Monitor zu starten. Speichern Sie den Speichebereich von Teil 2 mit dem Befehl *WRITE 1000 B000 C:\stage2.bin*.

Jetzt müssen wir sehen ob das Laden aus dem Speicher anstelle von Disk läuft und ob wir das richtige Diskettenladeprogramm gefunden haben. Setzen Sie einen Break-Punkt auf 0xBAF6, den Anfang des oben gezeigten Ladeprogrammes für den zweiten Teil, und starten Sie ein Spiel im Anfängermodus bis der Break-Punkt erreicht wird. Starten Sie den Monitor.

8 KONVERTIERUNGSANLEITUNG

Laden Sie im Monitor den Speicherbereich, den wir eben als `stage2.bin` gespeichert haben. Wir simulieren den Ladevorgang hier von Hand - geben Sie `READ C:\stage2.bin 1000 ffff` ein um den Speicherdump zu laden. Jetzt haben wir den Speicher geladen, aber wir sind noch am Anfang der Disklade-Routine. Wir müssen diese Routine umgehen da wir den benötigten Teil bereits von dem Speicherdump geladen haben. Wenn wir das disassemblierte Programm betrachten, direkt hinter dem Ende der Disklade-Routine, sehen wir dieses Programm:

```
BB43 LDA #$09 <--- Ende der Disklade-Routine, bereite Teil 2 vor.
BB45 STA $0A ;DOSVEC
BB47 LDA #$10
BB49 STA $0B ;DOSVEC+1
BB4B LDA #$03
BB4D STA $0232 ;SSKCTL
BB50 STA $D20F ;SKCTL
BB53 LDA #$00
BB55 STA $D208 ;AUDCTL
BB58 LDA $14 ;RTCLOCK+2
BB5A CLC
BB5B ADC #$01
BB5D CMP $14 ;RTCLOCK+2
BB5F BNE $BB5D
BB61 LDA #$00
BB63 STA $02CB ;COLOR4
BB66 LDA $D20A ;RANDOM
BB69 STA $092F
BB6C JMP $E474 ;WARMSV
```

Das Programm ist geladen, dann wird ihm die Kontrolle via Warmstart an Adresse `0x1009` übergeben.

Nach dem Laden von `stage2.bin` in den Speicher müssen wir mit `setpc bb43` die aktuelle IP setzen, dann mit `cont` das Programm weiterlaufen lassen um unsere Arbeit zu testen. Wenn alles gut gelaufen ist werden wir sehen das MULE ganz normal ein Spiel startet.

Nachdem wir einen funktionstüchtigen Lader für Teil 1 haben, und jetzt ein funktionierendes Beispiel wie ein Lader für den zweiten Teil erstellt haben, müssen wir den Dump vom zweiten Teil nur noch auf das Modul packen und das Diskettenladeprogramm durch eine modifizierte Kopie unseres Memory-Unpacker-Programmes ersetzen.

Hier ist die Übung zu Ende. Wenn Sie Ihr Wissen vertiefen möchten können Sie das vollständige Programm, die Binärdateien für beide Stufen sowie die Scripte, die zur Erstellung eines Modules benötigt werden auf der AtariMax Seite⁸ herunterladen und studieren.

Benutzen Sie die `-BIN2ATR`-Funktion des `“maxflash.pl”`-Scriptes um aus der Binär-Datei ein Flash-Modul zu erzeugen.

⁸<http://www.atarimax.com/flashcart/documentation/chapter8.html>

Abbildungsverzeichnis

1	Programmieren der Cartridge	11
2	Programmierung der Flashcart abgeschlossen	11
3	Keine Cartridge gefunden	13
4	Die fertiggestellte Cartridge	22
5	Menü einer 8MBit Cartridge	23
6	Beispiel einer Disketten Cartridge	26
7	Ali-Baba mit Translator (-OSB)	27
8	Bounty Bob Strikes Back	30
9	Bildschirmfoto einer typischen mit der -EXEPACKER Option erzeugten Multicart	34
10	Einstellen der Optionen des Emulators	41
11	Atari800Win Einstellungen	43
12	Black Lamp Absturz	46
13	Breakpoint an Speicherstelle \$660	48
14	Gepatches Black Lamp	49
15	Black Lamp Abbild im Hex-Editor	50
16	Zorro Programmabsturz	51
17	M.U.L.E.	53