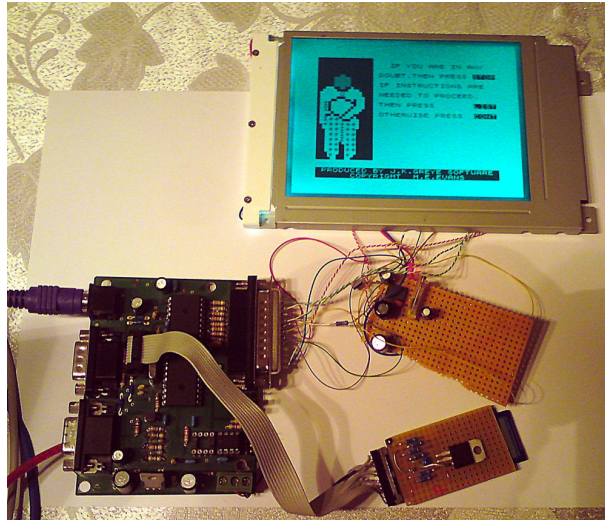


# AX81b: Ein ZX81++ Emulator im ATmega1284P

V1.33 (c) 2011-2012 Jörg Wolfram



## 1 Rechtliches

Die Programme unterliegen der GPL (GNU General Public Licence) Version 3 oder höher, jede Nutzung der Software/Informationen nonkonform zur GPL oder ausserhalb des Geltungsbereiches der GPL ist untersagt!

Die Veröffentlichung dieses Projekts erfolgt in der Hoffnung, daß es Ihnen von Nutzen sein wird, aber OHNE IRGEND-EINE GARANTIE, auch ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK.

Alle im Text genannten Marken sind Eigentum des entsprechenden Inhabers.

**HINWEIS:** Seit 2012 unterstütze ich nur noch freie Software, für die Tools zum Bearbeiten der Image-Dateien gibt es daher nur noch Linux-Binaries und den C-Quellcode, der sich zumindest auch unter BSD problemlos kompilieren lassen sollte. Ebenso beziehen sich die Anleitungen zum Schreiben und Lesen von Image-Dateien auf Linux.

## 2 Features

Als Eingabegerät dient eine normale PS2-Computertastatur, als Ausgabegeräte können Fernsehgeräte mit Scart- oder BAS-Eingang (S/W), VGA-Monitore oder auch grafische LCDs (320x240, ohne Controller) verwendet werden.

- Universalvariante mit ATmega1284P, auf 24/25MHz "übertaktet", ChipBasic Hardware kann (leicht modifiziert) weiter genutzt werden
- VGA / LCD / PAL / NTSC einstellbar
- Varianten mit 12 / 14,7456MHz und TV / LCD
- Variante mit 20MHz und VGA / LCD
- Z80 und System-Emulation bis zu 15K RAM in einem Mikrocontroller
- Geschwindigkeit auf "Original" einstellbar
- ZX81 Textmodus, UDG und CHR\$128, ZX80 Emulation
- ZX Spectrum und Jupiter ACE Emulation (ca. 60% Originalgeschwindigkeit)
- PS2-Tastatur zur Eingabe, derzeit werden DE und UK Layout unterstützt (nicht umschaltbar)
- Eigenes Tape-orientiertes Dateisystem mit Image-Dateien und Installationsprogramm
- Snapshotfunktion und serielle Übertragung von Snapshots zwischen zwei AX81b
- (abschaltbarer) Zugriff auf alle I/O Register des AVR

## 3 Konzept

Kernstück des Ganzen ist ein ATmega 1284P, die restliche Hardware besteht im Wesentlichen aus passiven Bauelementen und Steckverbindern. Das Videosignal wird intern über USART1 erzeugt, der (optionale) LCD Anschluss erfolgt über PORT A. Alternativ stehen bis zu 16 IO-Leitungen zur Verfügung. Die PS2-Tastatur wird über den USART0 des ATmega angeschlossen, an die SPI-Schnittstelle kann eine SD-Karte als Datenspeicher angeschlossen werden. Über Config-Dateien lässt sich das System (in Grenzen) an andere Hardwarekonfigurationen anpassen.

### 3.1 Systemvoraussetzung Host für das Assemblieren

Das Programm wurde unter Linux mit dem AVRA Assembler übersetzt. Eventuell muß der Pfad zu den Include-Dateien (bei mir `/usr/local/include/avr`) angepasst und eine der Zielhardware entsprechende `config.inc` im Hauptverzeichnis erstellt werden. Die Hex-Files sollte sich aber auch unter anderen Betriebssystemen brennen lassen.

### 3.2 Systemvoraussetzung Host für das Übersetzen der Tool-Sourcen

Normalerweise reicht ein installierter GCC und das Starten des `compile.sh` Scripts im Source-Ordner aus.

### 3.3 Tools zur Verwaltung der Images

Die im `tool`-Verzeichnis des Binärpaketes befindlichen Programme sind zum Erstellen des Dateisystems sowie zum Lesen und Schreiben von und auf die Images notwendig. Sinnvollerweise sollten sie nach `/usr/local/bin` kopiert werden. Alternativ befinden sich im `package` Verzeichnis ein rpm- und ein deb-Paket, über die sich die Programme ebenfalls installieren lassen.

### 3.4 Das AX81 Dateisystem auf SD-Karte

Zumächt ein Hinweis. Die Image-Dateien und Transfer-Programme der alten Versionen (<V1.0) sind **NICHT** kompatibel mit der neuen Version (V1.24 und aufwärts). Grund ist vor allem die variable Länge der Images und eine geänderte Methode zu Lokalisierung derselben auf dem Medium.

#### 3.4.1 Datenformat auf der SD-Karte

Für das Dateisystem ist zwingend mindestens eine FAT16 Partition auf dem Medium notwendig, Medien ohne Partitionstabelle oder fehlender FAT16 Partition werden nicht unterstützt.

Die Programme liegen in jeweils einer Image-Datei im Hauptverzeichnis, die Größe kann bei der Installation festgelegt werden. Für jeden Rechnertyp, der emuliert werden kann (ZX81, ZX80, Jupiter ACE und ZX Spectrum) existiert ein eigenes Image.

Jedes Image hat einen 512 Bytes großen Header-Sektor, in dem z.B. die Anzahl der Tapes vermerkt ist. Danach folgen die Programme der einzelnen Tapes lückenlos hintereinander. Ein Programm besteht aus einem Header-Block (512 Bytes) und dem Datenblock von 16 KBytes. Im Headerblock stehen der Programmname (10 Zeichen im ZX81 Zeichensatz) und danach 2 Bytes Dateilänge (bei ZX80 / ZX81). Beim Jupiter ACE und ZX Spectrum stehen ab Byte 256 die für den Snapshot relevanten Daten (Registerinhalte etc.).

Der Übersichtlichkeit halber sind je 32 Dateien zu (rein logisch) einem Tape zusammengefasst. Rein rechnerisch können so maximal 8192 Programme auf einer SD-Karte abgelegt werden und belegen dort ca.132 MBytes. Es werden SD, SDHC und MMC-Karten (nicht getestet) unterstützt. Zusätzlich gibt es je Emulationstyp einen Unterordner in dem sich weitere Unterordner entsprechend der Anzahl der Tapes befinden. Über diese Ordner erfolgt der "Datenaustausch" mit PC-Systemen.

#### 3.4.2 das Dateisystem initialisieren

Dazu sollte die SD-Karte frisch formatiert sein. Für meine Methode der Vorbereitung einer SD-Karte wird eine Root-Shell benötigt. Ich habe versucht, die folgenden Anweisungen möglichst allgemein zu halten, benutze aber selbst nur OpenSuSE. Erfahrungsberichte oder Anleitungen unter anderen Distributionen werde ich aber gerne mit einarbeiten.

Über den Desktop wird ein Terminal geöffnet und anschließend zum User "Root" mit dem Kommando **su** und nachfolgender Passwordeingabe gewechselt.

Zunächst hängt man das SD-Karten Laufwerk ein, indem man es über den Desktop öffnet wie wenn man z.B. Daten von dort kopieren will. Danach sucht man nach dem zuletzt eingebundenen Laufwerk, indem man den Befehl **cat /etc/mtab** in der Root-Shell eingibt. Dort müsste dann als letzter Eintrag etwas so ähnlich wie:

```
/dev/mmcblk0p1 /media/E945-1FF8 vfat ...
```

stehen. Wichtig ist in dem Fall das "/dev/mmcblk0", das p1 zeigt nur an, dass es sich um die erste Partition auf dem Medium handelt. Nun sollte das Laufwerk vom Desktop aus wieder ausgeworfen werden, zur Kontrolle kann /etc/mtab nochmals angesehen werden, die entsprechenden Einträge sollten nicht mehr existieren.

Der nächste Schritt ist die Erstellung einer passenden Partitionstabelle. Dazu geben wir den Befehl **fdisk /dev/mmcblk0** ein, fdisk sollte sich mit

Befehl (m für Hilfe):

melden. Als Befehl geben wir einfach ein **"p"** (print partition table) ein. In der nachfolgenden Ausgabe interessiert nur die Partitionstabelle am Ende, die z.B. so aussehen kann

Gerät	boot.	Anfang	Ende	Blöcke	Id	System
/dev/mmcblk0p1		2048	985087	491520	b	W95 FAT32

Ist unter den Partitionen mindestens eine mit der ID 6 (FAT16), so können wir die Root-Shell schließen und die vorhandene Partition einfach nutzen.

Im dargestellten Fall ist dies nicht so, aber auch nicht weiter kompliziert. Wenn bereits mindestens eine Partition angelegt ist, kann der nächste Schritt übersprungen und einfach der Partitionstyp geändert werden. Wenn sich keine Partition auf dem Medium befindet, muss mindestens eine angelegt werden. Das geschieht dann mit dem Befehl **"n"**, danach wählt man **"p"** für primäre Partition. Danach werden noch die Partitionsnummer (**"1"**), Startsektor (nur mit ENTER bestätigen) und Partitionsgröße (z.B. **+256M**) abgefragt. Mit dem Befehl **"t"** muss noch der Typ der Partition geändert werden, als Hexcode wird für FAT16 eine **"6"** eingegeben werden. Freier Speicher auf der Karte lässt sich mit weiteren, nicht für den AX81 genutzten Partitionen füllen.

Nun kann mittels des Kommandos **"w"** die neue Partitionstabelle auf die Karte geschrieben werden, danach beendet sich fdisk automatisch. Als letzter Schritt muß nun die Partition noch formatiert werden. Dazu gibt man **mkdosfs -F 16** gefolgt von der zu formatierenden Partition (hier: **/dev/mmcbk0p1**) ein, im Beispiel wäre das

**mkdosfs -F 16 /dev/mmcbk0p1**

Das **"-F 16"** sorgt für eine FAT16-Struktur, wenn der Vorgang abgeschlossen ist, sollte die Karte in den AX81 gesteckt und das System gestartet werden. Es sollte nun der Kartentyp und **"NO IMAGE FOUND"** angezeigt werden. Danach die Karte wieder zurück in den PC, denn es fehlen ja noch die Image-Dateien und Verzeichnisse.

Dazu dient das Programm **ax81-fsinstall** welches einfach im Hauptverzeichnis der Karte gestartet wird. Als Parameter müssen die Anzahl der Tapes für die verschiedenen Emulationen angegeben werden. Wenn das Programm ohne Parameter gestartet wird, wird eine Kurzbeschreibung ausgegeben.

**ax81-fsinstall 10 10 5 5**

erzeugt z.B. je 10 Tapes für ZX81, ZX80 und je 5 Tapes für Jupiter ACE und ZX Spectrum. Für jeden der 4 Parameter sind Werte zwischen 0 und 255 zulässig, wobei beim Wert 0 das betreffende Image nicht angelegt wird. Weiterhin sollte die Summe aller Tapes nicht größer als die Hälfte der Partitionsgröße (in Megabytes) sein, da dann etwa die Hälfte der Partition durch die Image-Dateien belegt wird und der Rest für die zu kopierenden Programme frei bleibt.

### 3.4.3 Alternative bei Problemen

Da es scheinbar teilweise Probleme mit dem **"ax81-fsinstall"** Programm gibt, kann auch die Struktur zuerst in einem Verzeichnis auf der Festplatte erzeugt und danach auf die frisch formatierte SD-Karte kopiert werden. Außerdem kann es bei manchen SD-Karten hilfreich sein, die Größe der genutzten Partition auf 256MBytes zu begrenzen.

### 3.4.4 Dateien auf SD-Karte schreiben

Dazu dient das Programm **ax81-write** aus dem **"tool"** Verzeichnis. Für die einzelnen Emulationen gibt es nur noch ein Programm, der Betriebsmodus wird über den Verzeichnisname beim Aufruf vorgegeben. Es schreibt alle (maximal 32) im aktuellen Verzeichnis befindlichen Files auf die freien Plätze des dem Unterverzeichnisname entsprechenden Tapes. Vor dem Schreiben wird das Tape formatiert (alle Programme darin gelöscht). Je nach Emulationstyp wird nach entsprechende Dateierweiterungen gesucht:

- für die ZX81-Modi **.P** oder **.p**
- für den ZX80-Modus **.O** oder **.o**
- für den ACE-Modus **.ACE** oder **.ace**
- für den Spectrum-Modus **.Z80** oder **.z80**

Da der emulierte Spectrum nur über 15K RAM verfügt, werden die meisten im Internet veröffentlichten **.z80** Dateien zu einem Absturz führen, da der Stack nach dem Laden in einem nicht beschreibbaren Bereich liegt.

Als Vorgehensweise empfehle ich, die zu schreibenden Dateien in das zu wählende **TAPExxx** Verzeichnis zu kopieren, dann in diesem Verzeichnis ein Terminal zu starten und dort den Befehl **ax81-write** einzugeben.

### 3.4.5 Dateien von SD-Karte lesen

Dazu dient das Programm **ax81-read** aus dem **"tool"** Verzeichnis. Es liest alle im (dem Unterverzeichnis entsprechenden) Tape vorhandenen Programme aus. Bestehende Programme werden ohne Nachfrage überschrieben. Auch hier ist es am einfachsten, in das **TAPExxx** Verzeichnis zu wechseln, hier ein Terminal zu starten und dort den Befehl **ax81-read** einzugeben. Zur Zeit ist das Programm noch unvollständig, ACE und Z80 Dateien können nicht zurückgelesen werden.

### **3.4.6 Nutzer anderer Betriebssysteme**

können sich 4 vorbereitete Images herunterladen und in das Wurzelverzeichnis der ersten FAT16-Partition auf der SD-Karte kopieren. Ausserdem sollte sich der Quellcode der Tools nach kleineren Modifikationen auch mit dem MinGW Compiler übersetzen lassen.

## 4 Was noch fehlt

Das Projekt ist nur bedingt vollständig. Neben sicher fälligen Bugfixes werden auch Teile der Funktionalität und der Dokumentation erst im Laufe der Zeit (oder niemals) fertig werden.

- Zurücklesen von Jupiter ACE und ZX Spectrum Snapshots am PC fehlt noch.
- Liste der lauffähigen Programme
- Beispielprogramme für I/O Ansteuerung

## 5 Changelog

Die Versionen ab V1.20 sind der Hauptzweig des Projekts. Die "alte" Version ( < V1.00 ) wird nicht mehr weiterentwickelt.

### 19.8.2012 Version 1.33

- Bugfix: Dokumentation der I/O Befehle war noch von Vorgängerversion
- Bugfix beim LCD M-Signal
- Bugfix: I/O Umschaltung bei ACE und Spectrum hatte Fehler
- Sourcen (vsys-Ordner) waren in der letzten Version unvollständig

### 7.8.2012 Version 1.32

- Varianten für 12/14,7456/20MHz hinzugefügt
- I/O Zugriff geändert, statt IN / OUT jetzt memory mapped

### 9.6.2012 Version 1.28

- Fehlermeldung bei ungültigen Snapshots (ACE + Spectrum)

### 9.6.2012 Version 1.27

- Detailliertere Fehlermeldungen bei der Karten-Erkennung
- Checksumme für FLASH Überprüfung
- SPI-Geschwindigkeit kann auf 1/4 reduziert werden

### 23.5.2012 Version 1.26

- erste öffentliche Version