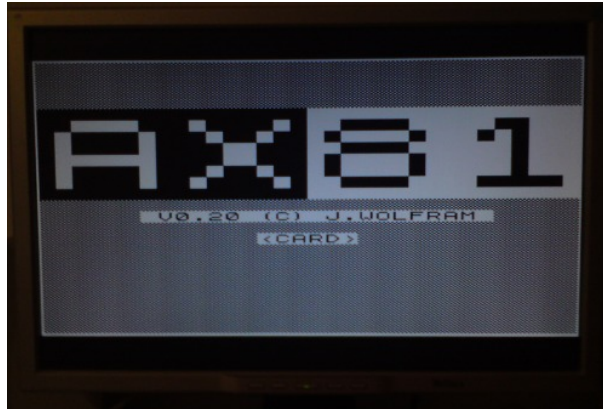


# AX81: Ein ZX81-Clone im ATmega

V0.26 (c) 2010-2012 Jörg Wolfram



## 1 Rechtliches

Das Programm unterliegt 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!

Eine Ausnahme bilden die fs\_install.exe Datei für die Installation des Dateisystems sowie die zum Lesen und Schreiben von Dateien in den Unterverzeichnissen generierten .exe Dateien. Diese liegen nur als Freeware vor.

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.

## 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.

- Z80 und System-Emulation bis zu 15,75K RAM in einem Mikrocontroller
- Geschwindigkeit entspricht in etwa einem 3,8-4,0 MHz Z80, abhängig vom Videomodus
- ZX81 Textmodus, Zeichensatz nicht änderbar
- VGA/LCD/PAL/NTSC einstellbar, VGA nur mit ATMega644P und ATMega1284P
- Unterstützt collapsed und normal DFILE sowie FAST und SLOW mode
- PS2-Tastatur zur Eingabe
- Eigenes Tape-orientiertes Dateisystem mit Image-Datei und Installationsprogramm

## 3 Konzept

Kernstück des Ganzen ist ein ATmega, die restliche Hardware besteht im Wesentlichen aus passiven Bauelementen und Steckverbindern. Das Videosignal wird intern über PORTC (TV) oder USART1 (nur 644P und 1284P) erzeugt, der LCD Anschluss erfolgt über PORT A. Über die config-Dateien lässt sich dies aber leicht anpassen.

Die PS2-Tastatur wird über den USART0 des ATmega angeschlossen. An die SPI-Schnittstelle wird eine SD-Karte angeschlossen.

### 3.1 Systemvoraussetzung Host für das Assemblieren

Das Ganze wurde unter Linux mit dem AVRA Assembler übersetzt. Eventuell muß der Pfad zu den Include-Dateien `/usr/local/include/avr` angepasst werden. Die Hex-Files sollte sich auch unter anderen Betriebssystemen brennen lassen.

### 3.2 Das AX81 Dateisystem auf SD-Karte

#### 3.2.1 Datenformat auf der SD-Karte

Die Programme liegen in einer 132 MBytes großen Image-Datei, die innerhalb der ersten 60 MB der SD-Karte beginnen muß. Jedes Tape hat darin eine eindeutige Kennung, mit der beim Systemstart der Beginn der Image-Datei bestimmt wird. 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.

Der Übersichtlichkeit halber sind je 32 Dateien zu (rein logisch) einem Tape zusammengefasst. Im ZX81 kann dann mittels **POKE 99,x** Befehl zwischen den einzelnen Tapes umgeschaltet werden. Rein rechnerisch können so maximal 8192 Dateien auf einer SD-Karte abgelegt werden und belegen dort ca. 132 MBytes. Ab Version 0.25 werden auch SDHC Karten unterstützt. Zusätzlich gibt es 256 Unterordner, die die Tapes repräsentieren und über die der "Datenaustausch" mit der Image-Datei erfolgt.

#### 3.2.2 das Dateisystem initialisieren

Dazu sollte die SD-Karte frisch formatiert sein. Zuerst werden entweder das Install-Programm oder die drei Install-Scripte in einen beliebigen Ordner auf der Karte kopiert. Dann wird einfach **install\_fs.exe** oder das Perl-script mit **perl install\_fs.pl** gestartet. Dann braucht es bis zu 5 Minuten, bis das Dateisystem angelegt ist. Neben der Image Datei gibt es einen Unterordner **TAPES** in dem sich die Verzeichnisse für die einzelnen Tapes befinden. Nach der Installation sollte das Installationsprogramm wieder gelöscht werden, da ein versehentlicher Aufruf eine neue, leere Image-Datei anlegt und dort bisher vorhandene Daten löscht.

### 3.2.3 Dateien auf SD-Karte schreiben

Dazu dient das Script **000\_write\_to\_tape.pl** oder das Programm **000\_write\_to\_tape.exe**. Es schreibt alle im aktuellen Verzeichnis befindlichen .P Files auf die freien Plätze des dem Unterverzeichnis entsprechenden Tapes. Sollte der verfügbare Platz nicht ausreichen, wird eine entsprechende Fehlermeldung ausgegeben. Vor dem Schreiben wird das Tape formatiert (alle Programme darin gelöscht). Die drei Nullen vor dem Dateinamen der programme / Scripte sollen dafür sorgen, dass diese immer am Anfang des Verzeichnisses stehen.

### 3.2.4 Dateien von SD-Karte lesen

Dazu dient das Script **000\_read\_from\_tape.pl** oder das Programm **000\_read\_from\_tape.exe**. Es liest alle im (dem Unterverzeichnis entsprechenden) Tape vorhandenen Programme aus. Bestehende Programme werden ohne Nachfrage überschrieben.

## 3.3 config- und Hexfiles

Die Konfiguration für das Assemblieren erfolgt über eine Konfigurationsdatei, die über ".include" eingebunden wird. Wenn nur das Hexfile benötigt wird, muss das zum verwendeten Controller passende ausgewählt werden.

## 4 Changelog

**17.1.2012** Version 0.26

- Bugfix: anstelle **OR A** wurde **OR L** ausgeführt

**21.11.2011** Version 0.25

- Unterstützung für SD-Karte mit FAT-Format überarbeitet
- Unterstützung Dataflash wieder eingefügt (automatische Erkennung)
- Bootloader startet nur bei DataFlash und Power On
- Speicherung im ZX81 Tape Format (experimentell)
- Unterstützung für ATmega328 wegen Codegröße entfernt
- Vereinfachte Konfiguration, nur noch 2 Hexfiles

**14.11.2011** Version 0.23 da die meisten Nutzer wohl aus dem Windows-Lager kommen und nur an für sie kompatiblen Lösungen interessiert sind, habe ich eine Trennung zwischen der "internen Entwicklung" und dem offiziellen Release vorgenommen.

- Unterstützung für SD-Karte mit FAT-Format
- Unterstützung für natives SD-Format und Dataflash entfernt
- Bootloader-Unterstützung entfernt

**26.10.2011** Erste öffentliche Version (v0.21)

- Noch keine Mega328-Version