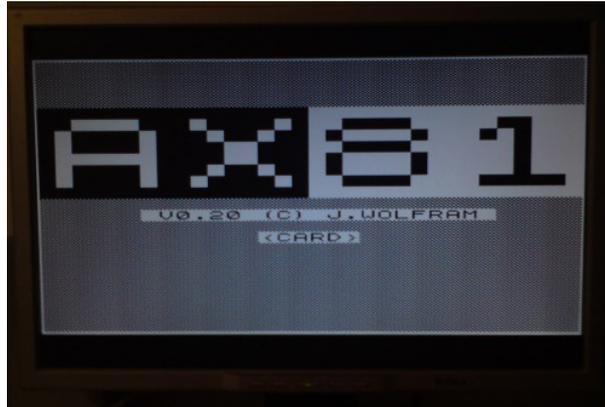


# AX81: Ein ZX81-Clone im ATmega

V0.21 (c) 2010-2011 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!

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
- frei verfügbarer I2C (z.Zt. nicht genutzt)
- Eigenes Tape-orientiertes Dateisystem auf SD-Karte oder Atmel Dataflash
- Bootloader zur seriellen Übertragung von Programmen

## 3 Konzept

Kernstück des Ganzen ist ein ATmega, die restliche Hardware besteht im Wesentlichen aus passiven Bauelementen und Steckverbindern. Timer 1 ist für das Videotiming zuständig. Kanal A arbeitet als PWM und erzeugt den Zeilen-Synchronimpuls. Im CSYNC-Mode (TV) wird während des vertikalen Synchronimpulses die PWM so umkonfiguriert, dass der Ausgang invertiert wird. Das Videosignal wird intern (VSYS=1) 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 Datenformat auf der SD-Karte

Die Programme liegen direkt in 16K Blöcken auf der Karte, dabei stehen am Anfang 10 Bytes Dateiname, (im ZX81 Zeichensatz) danach 2 Bytes Dateilänge und danach die Daten des P-Files. Im Tools-Ordner gibt es Perl-Scripte mit denen Dateien zwischen SD-Karte und PC einfach austauschbar sind. Die Scripte setzen neben einem Perl-Interpreter `dd` voraus und sind deswegen unter Windows wahrscheinlich nicht direkt nutzbar.

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 128MBytes. SDHC Karten werden momentan nicht unterstützt. **!!! ACHTUNG Beim Schreiben auf die SD-Karte werden dort bereits vorhandene Dateisysteme überschrieben und damit zerstört !!!**

#### 3.2.1 Dateien auf SD-Karte schreiben

Dazu dient das Script `write_tape.pl`. Es schreibt alle im aktuellen Verzeichnis befindlichen .P Files auf die freien Plätze eines Tapes. Sollte der verfügbare Platz nicht ausreichen, wird eine entsprechende Fehlermeldung ausgegeben. Der erste Parameter ist die Nummer des zu schreibenden Tapes (0...255), danach kann ein optionales `-f` dafür sorgen, dass das Tape vor dem Schreiben formatiert wird. Als letzten Parameter kann das zu beschreibende Device angegeben werden. Fehlt diese Angabe, wird `/dev/mmcblk0` verwendet.

```
./write_tape 1 -f
```

Schreibt die .P Dateien im aktuellen Verzeichnis auf die freien Programmplätze von Tape 1, dabei wird das Default-Device benutzt.

### 3.2.2 Dateien von SD-Karte lesen

Dazu dient das Script **read\_tape.pl**. Es liest alle Dateien des angegebenen Tapes als .P Dateien in das aktuelle Verzeichnis ein. Der erste Parameter ist die Nummer des zu schreibenden Tapes (0..255), danach kann optional das zu lesende Device angegeben werden. Fehlt diese Angabe, wird **/dev/mmcblk0** verwendet.

```
./read_tape 1 /dev/sdb
```

Liest alle -P Dateien von Tape 1 des Gerätes /dev/sdb1 in das aktuelle Verzeichnis ein.

## 3.3 config- und Hexfiles

Die Konfiguration für das Assemblieren erfolgt über mehrere Konfigurationsdateien, die über ".include" eingebunden werden. Im **config**-Ordner befinden sich dazu verschiedene Beispiele, die entsprechend den Anforderungen in Source-Hauptverzeichnis kopiert werden müssen.

Wenn nur das Hexfile benötigt wird, muss das zu der aktuellen Hardwarekonfiguration passende ausgewählt werden.

## **4 Changelog**

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

- Noch keine Mega328-Version