

# AVR-ChipBasic8: Hardware

V0.50 (c) 2006-2009 Jörg Wolfram

## 1 Allgemeines

### 1.1 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 Das Konzept und die Realisierung

Kernstück des Ganzen ist ein ATmega8 oder ATmega88, 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 wird während des vertikalen Synchronimpulses die PWM so umkonfiguriert, dass der Ausgang invertiert wird.

Kanal B von Timer 1 erzeugt den (Horizontal-) Videointerrupt. Dieser enthält eine Synchronisation auf den Timer, dadurch erzeugen auch EEPROM-Lesezugriffe mit 5 Takten CPU-Stop keine Bildstörungen. Timer 2 erzeugt PWM mit dem Audiosignal. Dieses wird entweder mit dem (Pseudo-) Zufallsgenerator oder per DDS (Sägezahn) und Hüllkurve erzeugt. Die PS2-Tastatur ist an den USART des ATmega angeschlossen. Wobei sich natürlich dann die Frage stellt „und was ist mit seriell?“ Mit reduzierter Bitrate (1200Baud) lässt sich die serielle Schnittstelle im Horizontal-Interrupt mit zwei Portpins realisieren. Ein dritter treibt eine Ladungspumpe, die ca. -4,7V für die serielle Schnittstelle bereitstellt, dadurch sind keine Spezialbausteine (MAX232 etc.) notwendig. Alternativ kann aber auch ein Standard-UART simuliert werden. Mit PC0-PC3 stehen 4 frei benutzbare Ein/Ausgänge zur Verfügung. Diese können wahlweise als Digitaleingang, Digitalausgang oder Analogeingang benutzt werden. Das BASIC-Programm selbst ist im Flash gespeichert und belegt dort 512 Bytes. Vor jedem Programmstart (Taste F4) wird überprüft, ob sich das im Editor befindliche Programm von dem im Flash unterscheidet. Ist dies der Fall, wird das Programm vor dem Start vom Editor (RAM) in den Flash-Speicher gesichert.

### 2.1 Assemblieren

Da fast alle I/O-Funktionen von verschiedenen Bibliotheken bereitgestellt werden und sehr stark von Makros Gebrauch gemacht wird, ist zum Assemblieren der AVRA-Assembler notwendig. Die bereits mitgelieferten Hex-Files sollten sich auch ohne diese Voraussetzung brennen lassen. Eine Besonderheit gibt es für den ATmega88, wenn man **avr\_libmake** benutzen will. Für den **avr\_libmake** Lauf muss der Controller als Mega8 definiert sein, hinterher muß der Controller auf Mega88 undefiniert werden und noch einmal zusätzlich AVRA gestartet werden (**avra main.asm**). Das hängt damit zusammen, dass der vor dem ersten Assemblerlauf erzeugte Quelltext für den Mega88 zu einem Abbruch des Assembliervorganges wegen überlappender Codesegmente führt.

### 2.2 Konfiguration

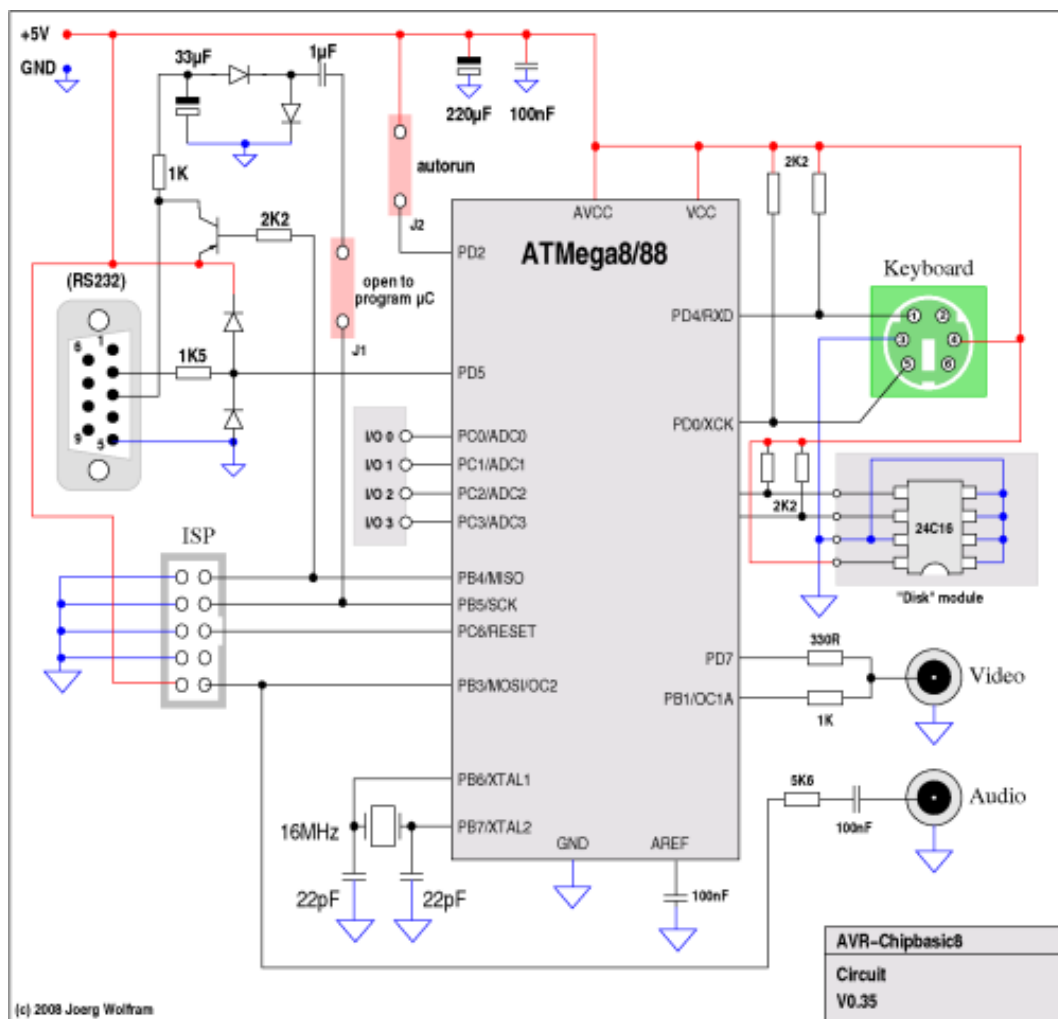
Im **src**-Verzeichnis befindet sich eine Datei namens **config**. Hier können einige Optionen für das Assemblieren ausgewählt werden, die sich hauptsächlich auf die verwendete Hardware beziehen:

- **ctype** wird der Controller ausgewählt. Gültige Werte sind **atmega8**, **atmega16** und **atmega88**
- **video\_norm** kann zwischen **pal** und **ntsc** gewählt werden

- **kbd\_type** wählt das Keyboard-Layout, **kbd\_de** und **kbd\_en** sind möglich
- mit **std\_uart** kann gewählt werden, ob der serielle Port mit der angegebenen Schaltung arbeiten (Wert=0) oder sich wie ein Standard-Uart (Wert=1) verhalten soll.

### 3 Der Aufbau

Für die Beschaltung reicht schon ein Steckbrett, eine Streifenleiter- oder eine Lochrasterplatte aus. Lediglich die Steckverbinder für Tastatur (Mini-DIN 6) und Video/Audio (Cinch) passen in den meisten Fällen nicht direkt in das 2,54mm Raster der Lochrasterplatten. Ab Softwareversion 0.50 wird der Autorun-Jumper nicht mehr benötigt.



## 4 Die Schnittstellen

### 4.1 Video und Audio

Ausgegeben werden ein S/W Videosignal (an 75 Ohm) und ein Audiosignal welches beim getesteten TV ungefähr gleiche Lautstärke wie die restlichen Sender produzierte. Evtl. muss der 5K6 Widerstand angepasst werden.

### 4.2 Tastatur

Es kann jede PS2-Tastatur mit deutschem oder US Tastaturlayout angeschlossen werden.

### 4.3 Die serielle Schnittstelle

Da der USART bereits für den Tastaturanschluss verwendet wird, wird die serielle Schnittstelle in Software realisiert. Systembedingt lässt sich die TX(Sende)-Leitung aber nur 1 mal pro Bildschirmzeile aktualisieren. Daraus ergibt sich die recht niedrige Geschwindigkeit von nur 1200Bps. Im Gegensatz zu anderen Lösungen wird dafür kein extra Schnittstellen-IC benötigt. Ein Ausgang bildet mit zwei Schottky-Dioden und zwei Kondensatoren eine Ladungspumpe, die ca. -3 V erzeugt. Diese liegt über einen Widerstand am TX-Anschluss an. Soll jetzt eine „0“ ausgegeben werden, schaltet der Ausgang des Mikrocontrollers einen PNP-Transistor durch. Dadurch liegen am TX-Anschluss jetzt ca. 4,5V. Empfangsseitig wird die Eingangsspannung mittels Vorwiderstand und zweier Dioden auf 0..5V begrenzt. Das Signal am Portpin ist dadurch zwar negiert, da die Auswertung in Software erfolgt ist das aber kein Problem. Soll ein RS232 Schnittstellenkonverter (MAX232 etc.) eingesetzt werden, muß das Flag **std\_uart** in der Datei main.asm vor dem Assemblieren auf 1 gesetzt werden. Dann wird die Negierung aufgehoben und die Ladungspumpe nicht aktiviert. Im **hexfiles** Verzeichnis gibt es auch entsprechend vorbereitete Dateien.

### 4.4 Die I/O Leitungen

An den 4 I/O-Leitungen können digitale Signale ausgegeben und abgefragt werden. Ausserdem können auch analoge Spannungen im Bereich 0-5V gemessen werden.

### 4.5 Die I2C-Schnittstelle

Der Signalausgleich findet über zwei Leitungen, **SCL** (Serial Clock) und **SDA** (Serial Data) statt. Darüber können serielle EEPROMs vom Typ 24C16 zum Speichern von Programmen angeschlossen werden. Damit die angeschlossene Peripherie nicht in jedem Fall eine extra Stromversorgung braucht, sind am I2C Anschluss auf der Platine auch +5V und GND verfügbar.

### 4.6 Die ISP-Schnittstelle

Diese Schnittstelle dient nur zur Programmierung des Mikrocontrollers. Dazu muß der Jumper J1 geöffnet werden, da ansonsten die Ladungspumpe den MISO-Pin belastet. Außerdem ist es sinnvoll, den Audio-Anschluß abzuziehen.