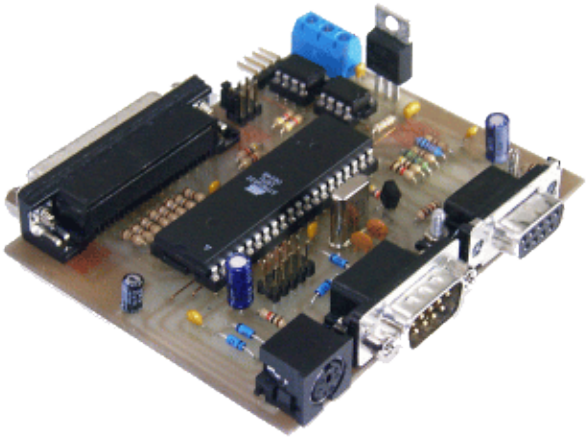
AVR-ChipBasic2: Ein BASIC-programmierbarer Einchip-Computer mit dem ATMega644

V0.89 (c) 2006-2009 Jörg Wolfram



1 Rechtliches

Das Programm unterliegt der GPL (GNU General Public Licence) Version 2 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 IRGENDEINE 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 Geschichte

Angefangen hat es mit der Idee, ein TV-Gerät zum Testen und Debuggen eines Formelparsers zu nutzen, da so mehrere Werte übersichtlich und gleichzeitig darstellbar waren.
Irgendwann im August 2006 kan dann die Idee, einen BASIC-Interpreter mit TV-Ausgabe in den Chip zu integrieren. Um das ganze einfach zu halten, habe ich mich auf einen TINY-BASIC ähnlichen Dialekt beschränkt. Und so kamen dann nacheinander Farbausgabe, ein Menü, ein integrierter Editor und weitere Funktionen dazu. Anfänglich konnte nur ein Programm im EEPROM-Speicher des ATMega16 Platz finden, beim Mega32 wurde dann der Flash benutzt. Nach den Versionen für die ATMega16 und ATMega32 Mikrocontroller stellt die Version für den ATMega644 die logische Weiterentwicklung bei gleicher Hardwarebasis dar.

3 Features

Als Eingabegerät dient eine normale PS2-Computertastatur, als Ausgabegerät ein Fernsehgerät mit Scart-Eingang (Farbe) oder BAS-Eingang (Graustufen) oder auch verschiedene PAL-/NTSC-taugliche TFT-Displays.

- 95 Programmzeilen mit maximal 32 nutzbaren Zeichen, Fullscreen-Editor
- 8 Programme im internen Flash
- Tiny-Basic-Programmiersprache mit Erweiterungen

- Eigene Fehlerbehandlung mit ONERR...
- 30x23 Zeichen mit maximal 8 Vorder- und Hintergrundfarben und Pseudografik
- 5 Videomodi, davon 4 mit Vollgrafik
- Sprites im Textmodus, Anzahl ist nur duch Array limitiert
- Schneller Bitblock-Transfer im Grafikmodus
- PAL/NTSC und Synchronsignale über Jumper einstellbar
- PS2-Tastatur zur Eingabe, Keyboard-Layout ist umschaltbar
- 1-Kanalige Audioausgabe (Noten in 2 Klangfarben, Rauschen) mit Hüllkurve
- Einfacher, im Hintergrund laufender Sequenzer
- serielle RS232-Schnittstelle mit 1200/2400 Bps und Ladungspumpe
- parallele Druckerschnittstelle, auch als I/O und Analogeingänge nutzbar
- optionales Daten-EEPROM (...24C512) für Datenlogger etc
- I2C-Anschluß für bis zu 8 LM75 Temperatursensoren oder andere I2C Devices
- Up/Download von Programmen über die serielle Schnittstelle
- Listingdruck über die Druckerschnittstelle
- automatischer Start des Programmes nach Einschalten über Jumper einstellbar
- Tastenkombinationen für Abbruch, Neustart, Screenshot
- Monitor mit Variablen- und Stackanzeige, Einzelschrittbetrieb
- Universelle I2C-Ansteuerung
- Einfaches Dateisystem auf ATMEL Dataflash, wahlweise direkter Page-Zugriff
- Clone-Funktion zum Kopieren der Software auf weitere Controller

4 Hard- und Softwarekonzept

Kernstück des Ganzen ist ein ATmega644, 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 mit Wavetable und Hüllkurve erzeugt.
Entgegen den meisten anderen Lösungen ist die PS2-Tastatur and den USART des ATMega angeschlossen. Eine Nutzung externer Interrupts verbietet sich schon wegen des Videotimings, man kann zwar das Clock-Signal während der Bilddarstellung auf LOW ziehen und damit die Tastatur zum Warten verdonnern, aber nicht jede Tastatur schafft es, ein Datenpaket während der Austastlücke zu senden und bei NTSC klappt das Ganze dann aber gar nicht mehr, weil dort die Austastlücke noch kürzer ist. Die zweite Möglichkeit wäre die SPI-Schnittstelle, aber die ist dann auch belegt. Und gerade während der Entwicklungsphase ist es lästig, wenn man ständig die Tastatur abziehen und anstecken muss. Die Nutzung des USART hat hingegen den grossen Vorteil, dass alles automatisch geht und man nur noch das fertige Zeichen abholen muss.
Wobei sich natürlich dann die Frage stellt „und was ist mit seriell?" Mit reduzierter Bitrate (1200 oder 2400 Bps) lässt sich die serielle Schnittstelle im Horizontal-Interrupt mit zwei Portpins realisieren. Timerausgang von Timer 0 treibt eine Ladungspumpe, die ca. -4,7V für die serielle Schnittstelle bereitstellt, dadurch sind keine weiteren Bausteine (MAX232 etc.) notwendig. Um Strom zu sparen, lässt sich diese Funktion auch per Software abschalten.
Die parallele Schnittstelle wird über Port A realisiert (+2 Steuerleitungen aus Port B). Wenn keine Druckerschnittstelle benötigt wird, können die 8 Pins als Ein-/ oder Ausgang konfiguriert oder auch als Analogeingang genutzt werden.

4.0.1 Systemvoraussetzung Host für das Assemblieren

Da fast alle I/O-Funktionen von verschiedenen Bibliotheken bereitgestellt werden, sind zum Assemblieren avr_libmake (läuft nur unter Linux und organisiert die Makros für die Bibliotheksfunktionen, es geht aber auch ohne) und der AVRA-Assembler notwendig. Das Hex-File sollte sich auch unter anderen Betriebssystemen brennen lassen.

5 Changelog

8.1.2009 Version 0.89

- Jede Menge Bugfixes (Vielen Dank an Wolfgang Günther fürs Testen)
- Array-Ansicht im Debugger
- Clone-Funktion sollte auch mit Mega644P funktionieren
- Problembeseitigung im Zusammenspiel mit Tastaturen (US Layout)
- Bugfix "IO DISABLED" beim Zugriff auf IO-Port
- Konvertierungsprogramm für im Textmode empfangene Screenshots

21.12.2008 Version 0.87

- Bugfixes, Redundanzen entfernt
- verbesserter Tokenizer, höhere Geschwindigkeit zur Laufzeit
- Daten-EEPROM Adresse kann eingestellt werden
- Seriell 1200 und 2400Bps einstellbar
- Konfiguration von I2C und SPI-Geschwindigkeit
- Datenübertragung auf Binärebene via X-Modem
- Bootloader für kommende Updates
- Native (Binär) Programme sind vorbereitet, API fehlt noch
- MENU-Funktion für eigene Menüs am unteren Bildschirmrand
- Array-Schreibzugriffe sind nur noch über DATA möglich
- EMIT und YEMIT sind entfallen (kann über PRINT realisiert werden)
- RREAD/RWRITE für direkten Zugriff auf das Dataflash
- überarbeitete Dokumentation
- Neuer Videomoddus (128x64 monochrom) kompatibel zum AVR Handheld Computer
- Zweiter Font (6x4) in den Grafikmodi

28.04.2008 Bugfix-Version (0.75)

- Überarbeitung der Dateistruktur im Projektverzeichnis
- ein paar Anpassungen an den AVRASM32
- bei 16Bit-Schreibzugriffen auf das Array wurde 2x das LOW-Byte geschrieben
- Dataflash-Delete zeigte keine Auswahlbox mehr an
- EEPROM-Schreibzugriffe schlugen manchmal fehl
- Palette wurde teilweise falsch gesetzt

- RND(n) erzeugte nur Zufallszahlen zwischen 0 und n/2
- geändertes Videotiming, dadurch ruhigeres Bild
- Intro-Screen
- Statt LPIX und LCHAR gibt es jetzt SCROLL im Textmodus
- Zugriff auf die SPI-Schnittstelle auch vom BASIC aus
- überarbeitete Dokumentation
- Ein paar neue Beispielprogramme

16.02.2008 Dritte öffentliche Mega644-Version (0.72)

- ein paar Bugfixes
- Clear und Copy für Programme
- SIN() und COS() arbeiten jetzt nur noch mit ganzen Graden als Argumente
- überarbeitete Doku und Webseite
- Ein paar Beispielprogramme

31.01.2008 Zweite öffentliche Mega644-Version (0.69)

- ein paar Bugfixes
- Neue Sprite-Routinen
- im Hintergrund arbeitender Noten-Sequenzer

14.01.2008 Erste öffentliche Mega644-Version (0.65)

- ein paar Funktionen sind noch nicht vollständig getestet
- Beispielprogramme fehlen noch