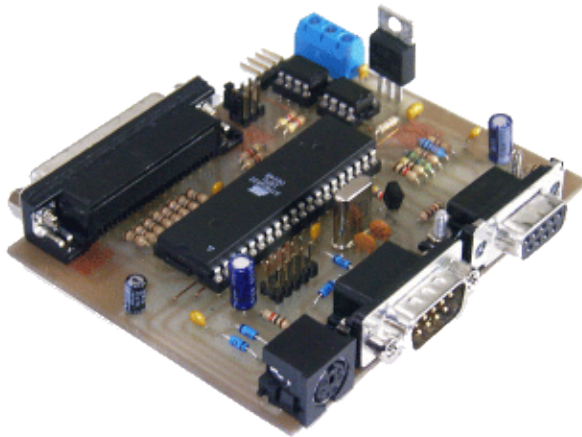


AVR-ChipBasic2: Ein BASIC-programmierbarer Einchip-Computer mit dem ATMega644

V1.36 (c) 2006-2011 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 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 kam 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 schon 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. Selbst dieses Programm ist noch einem zeitlichen Wandel unterlegen und so kann sich mit späteren Versionen noch das eine oder andere Feature ändern, wobei wenn möglich die Kompatibilität zu bestehenden Programmen gewahrt werden wird.

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 mit RGB Ansteuerung.

3.1 Hardware

- 30x23 Zeichen mit maximal 16 Vorder- und Hintergrundfarben und Pseudografik im Standardmodus
- PAL/NTSC und Synchronsignale über Jumper einstellbar
- 8 Videomodi, davon 4 mit Vollgrafik, 2 mit benutzerdefinierten Zeichen und ein User-Modus
- PS2-Tastatur zur Eingabe, ladbares Tastaturlayout
- 1-Kanalige Audioausgabe (Noten in 2 Klangfarben, Rauschen) mit Hüllkurve und Lautstärkeeinstellung
- I2C-Anschluss für optionales Daten-EEPROM, bis zu 8 LM75 Temperatursensoren oder andere I2C Devices
- serielle RS232-Schnittstelle mit 1200/2400 Baud und Ladungspumpe
- zweite serielle Schnittstelle des ATmega644P nutzbar
- parallele Druckerschnittstelle, auch als I/O und Analogeingänge nutzbar
- Am Parallelport können Speichererweiterungen angeschlossen werden

3.2 BASIC

- Tiny-Basic-Programmiersprache mit Erweiterungen
- eigene Fehlerbehandlung mit ONERR, periodischer Unterprogrammaufruf mit ONSYNC...
- 95 Programmzeilen mit maximal 32 nutzbaren Zeichen, Fullscreen-Editor, 8 Programme im internen Flash
- Sprites im Textmodus, Anzahl ist nur durch Array limitiert
- Schneller Bitblock-Transfer im Grafikmodus
- Einfacher, im Hintergrund laufender Sequenzer
- Up/Download von Programmen über die serielle Schnittstelle
- Listingdruck über die Druckerschnittstelle
- Tastenkombinationen für Abbruch, Neustart, Screenshot
- Monitor mit Variablen- und Stack- und Array-Anzeige, Einzelschrittbetrieb
- universelle I2C-Ansteuerung
- GOSUB mit Parameterübergabe, RETURN mit Rückgabewert
- programmübergreifender Aufruf von Unterprogrammen möglich
- Zusätzliche BASIC-Befehle einbindbar

3.3 SYSTEM

- eigenes Dateisystem auf ATMEL Dataflash
- Bootloader zum System-Update
- X-Modem Dateitransfer für Binärprogramme
- automatischer Start des Programmes nach Einschalten (über Tastendruck deaktivierbar), Programm wählbar
- umfangreiches API für die Entwicklung von Binärprogrammen und -bibliotheken
- zusätzliche Videotreiber einbindbar

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 und den USART des ATmega644 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 (1200/2400Bps) 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 Spezialbausteine (MAX232 etc.) notwendig. Um Strom zu sparen, lässt sich diese Funktion auch per Software abschalten, ebenso kann auch auf "Standardpegel" umgeschaltet werden, um z.B. USB zu RX/TX Wandler nutzen zu können. Mit der Version 1.27 hat auch Unterstützung für die zweite serielle Schnittstelle des ATmega644P Einzug gehalten. Hier sind standardmäßig 8 verschiedene Bitraten einstellbar. 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. Desweiteren können auch Treiberprogramme die parallele Schnittstelle nutzen, um z.B. LCDs anzusteuern. In diesen Fällen sind die "normalen" Ausgabefunktionen der parallelen Schnittstelle deaktiviert.

4.1 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.

4.2 Programm-Typen

Neben den BASIC-Programmen gibt es noch zwei weitere Typen. Da sind zum einen native AVR-Programme, die im Binärformat vorliegen. Erkannt werden sie daran, dass als 13.Byte ein "N" für native steht. Dieses "Flag" lässt sich sinnvollerweise auch nicht ändern. Die zweite Eigenheit haben Programme, deren Name mit einem Unterstrich beginnt. Diese werden nämlich beim Speichern nicht tokenisiert und lassen sich deswegen auch nicht starten. Hintergrund dafür ist, eine Möglichkeit zu haben, reine Textdateien zu erzeugen. Allerdings bestehen diese immer aus 95 Zeilen a 32 Zeichen, immer mit Leerzeichen aufgefüllt. Bei den Binärprogrammen wird dann noch zwischen einfachen Programmen, Bibliotheken und Treibern unterschieden.

5 Changelog

15.1.2011 Version 1.36

- Suchfunktion im Editor
- Die Shift-Befehle teilweise durch Shift-Operatoren («, ») ersetzt
- XOR-Operator ^
- Unterstützung von bis zu 64 KBytes externen Speicher am Parallelport
- Änderung des Bibliotheks- und Treiber-Konzeptes (Kodierung über Flagbyte)
- bei den PWM-Treibern lassen sich BUSY und STROBE als I/O nutzen
- neue Videotreiber
- Monitor aus allen Videomodi heraus aufrufbar, allerdings mit eingeschränkter Funktionalität
- Erweiterbarkeit des BASIC um zusätzliche Befehle
- erweitertes ADC-Setting über den Parameterbereich 0x100 ... 0x1ff
- erweitertes Bitraten-Setting für die zweite serielle Schnittstelle
- **BIN**-Programme werden jetzt als **AVR** angezeigt
- Korrekturen und Erweiterungen in der Dokumentation
- Beispiel-Programme überarbeitet
- erweitertes API
- Neue Binärprogramme (Chip8-Emulator)
- Bugfix: Alertbox beim Programm-Löschen überdeckte Programmsymbole
- Bugfix: Baudraten für zweite serielle Schnittstelle waren um Faktor 2 zu niedrig
- Bugfix: teilweise wurde bei Ausgabeumlenkung die Kanalnummer statt Variablenwert ausgegeben

11.4.2010 Version 1.28

- Bugfix: SPI() und Clone funktionierten nicht richtig wegen Problemen mit Include-Datei
- Bugfix: Dokumentation enthielt ein paar falsche Bilder

7.4.2010 Version 1.27

- Hardware-Änderung: zusätzlich 5V auf der Parallelschnittstelle
- Umgestaltung des Hauptmenüs
- zusätzliche Tastenkombinationen im Editor
- Cursorpositionen werden beim Verlassen des Editors im EEPROM gespeichert
- nur noch eine Tastaturlayouttabelle, dafür änderbar
- kleine Änderungen im Zeichensatz
- Neue Shift-Befehle (LSHIFT, ASHIFT)
- Baudrateneinstellung (BAUD)
- Bildschirm-Einstellung (WRAP)
- Unterstützung der zweiten seriellen Schnittstelle des ATmega644P (EBAUD, ESPUT, ESGET)

- CALL kann durch CA abgekürzt werden
- CALL alternativ mit Zeichenkette als 2.Parameter
- LFINd zum finden von Binärbibliotheken
- Binärprogramme mit eigenen Symbolen
- API erweitert (Auslesen des Keyboard-Status, Textparser, Treiberfunktionen)
- Videomode 7 entfernt, eigener Treiber kann auf Programmplatz 8 geladen werden
- Laderprogramm für Remote-Steuerung, unabhängig vom Autostart
- Bugfix: In manchen Fällen wurde bei Fehlern die falsche Zeile angezeigt
- Bugfix: ACOPY kann sich in einer Endlosschleife festhängen
- Bugfix: Bootloader funktionierte nicht in der seriellen Einstellung "standard"

5.9.2009 Version 1.10

- Aus GOSUB mit zwei Paramtern wird CALL, wegen flexibler Paramterlisten
- GOSUB und CALL können bis zu 5 Parameter haben
- RETURN kann einen Rückgabewert enthalten
- API überarbeitet und erweitert
- Bugfix: Tilde-Zeichen für US-Tastaturen hat gefehlt

27.7.2009 Version 1.03

- zwei weitere API-Funktionen zum kopieren von Zeichen in den User-Zeichensatz hinzugefügt
- API-Funktionen zur Nutzung von BASIC-Routinen
- Bugfix in der api_dataptr Funktion
- Programme können auch untokenisiert gespeichert werden (Programmname beginnt mit Unterstrich)

16.7.2009 Version 1.02

- neue Config-Page
- neuer Befehl **SCALE** um Messwerte umzurechnen
- API überarbeitet und erweitert, Dokumentation ist jetzt komplett
- **PLAY** hat jetzt vierten Parameter um Anzahl der Sequenzerzyklen zu steuern
- 2 Bugfixes im Bereich, der nicht durch den Bootlaoder aktualisiert werden kann
- Bugfix: GOSUB in FOR/NEXT Schleife erzeugte Fehlermeldung beim RETURN

26.6.2009 Version 1.00

- 3 neue Videomodi, Modus 1 wurde erweitert
- Umbaumöglichkeit auf 16 anstelle 8 Farben
- Mehr Möglichkeiten in der Sound-Ausgabe
- ONSYNC zum zeitgesteuerten Aufrufen von Unterprogrammen
- Support für native AVR-Anwendungen, API ist implementiert
- Zeichenfunktionen für Kreis- und Ellipsensegmente

- Ein paar neue Funktionen
- Zeilenenden (seriell senden) sind konfigurierbar
- Serielle Schnittstelle ist umschaltbar auf "Standardpegel"
- Autostartfunktion jetzt konfigurierbar, Autorun-Jumper entfällt
- freie Stellen im Zeichensatz aufgefüllt
- Tastaturtest beim Einschalten
- Überarbeitete Dokumentation
- REPEAT-UNTIL Schleifen

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 Videomodus (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-Sequenzler

14.01.2008 Erste öffentliche Mega644-Version (0.65)

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