

AVR von Anfang an – Lernpaket Mikrocontrollertechnik mit BASCOM

„Der leichte Einstieg in die Mikrocontroller-Technik“, schreibt Franzis vielversprechend. Und in der Tat, selten hat man solch ein komplettes Paket in der Hand, das mit vollständiger Software und einer durchdachten Hardware den Einsteiger an die Hand nimmt und ihn in die Lage versetzt, von Beginn an den Umgang mit den derzeit wohl populärsten Mikrocontrollern zu erlernen und deren Programmierung auf besonders einfache Weise zu erforschen. Wir stellen das Lernpaket vor und wollen hier vor allem einen globalen Eindruck über dessen Aufbau und Möglichkeiten vermitteln.

Platine, Bauteile, Handbuch, CD...

Das ist im Wesentlichen das, was man nach Öffnen des Kartons vorfindet. Natürlich gilt das erste Augenmerk der Platine (siehe Titel)! Darauf finden sich zwei Chips – ein USB-Controller FT 232R, der die Verbindung zwischen UART-Schnittstelle des ATmega-AVR-Controllers und der USB-Schnittstelle auf dem Board realisiert, und der erwähnte AVR-Controller vom Typ ATmega88. Ringsum einige Leuchtdioden, Taster, ein Poti und einige Experimentierfassungen, auf die auch die Ports des AVR-Controllers geführt sind.

Und wie soll der AVR programmiert werden? Von einer ISP-Schnittstelle keine Spur! Das erste Querlesen des Handbuchs bringt es an den Tag: Auf dem ATmega88 ist bereits ein Bootloader installiert, der dafür sorgt, dass man keine ISP benötigt, sondern Programme einfach von der Entwicklungsumgebung aus direkt via USB in den AVR laden kann. Der erste einsteigerfreundliche Punkt! Kein Kabelgewirr, keine zusätzliche Schnittstelle, kein Herumschlagen mit einem zusätzlichen Programm – nur eine USB-Verbindung.

Der zweite Blick gilt dem beiliegenden, 222 Seiten starken Handbuch. Schon das Inhaltsverzeichnis macht Appetit und

lässt erkennen, dass man sich hier tatsächlich an den Einsteiger richtet: Punkt für Punkt wird zunächst in die Technik der Mikrocontroller eingeführt, eine Übersicht über die AVR-Familie gegeben und der verwendete ATmega88 sowie die einzelnen Bauteile des Lernpakets genauer vorgestellt.

Nach der Inbetriebnahme mit Programmierung des FT232R geht es dann an das softwaretechnische Kernstück: BASCOM. Diese einfach erlernbare Programmiersprache, ein komplexer BASIC-Dialekt, garantiert eines – schnellen Erfolg. Darum geht der Autor des Buches auch sofort ans Eingemachte – die ersten Programme und deren Erläuterung. Darauf folgen eine ausführliche Einführung in die einzelnen BASCOM-Befehle, Programmstrukturen, Abläufe, stets begleitet durch praktische Experimente auf dem Experimentierboard, sowie auf dem Löwenanteil des Buchumfangs zig praktisch nachvollziehbare Programme, die sich allesamt auf der begleitenden CD-ROM befinden und so direkt zu eigenen Programmierexperimenten einladen.

Hier findet man eine riesige Ansammlung von praktisch einsetzbaren Programmen vom einfachen Tastenentprellen (dem Einsteigerprogramm neben „Hallo Welt“) bis hin zu schon komplexeren Steuerungen.

Der Mikrocontroller

Als Controller kommt ein ATmega88 zum Einsatz, der wahlweise mit 1 bis 16 MHz extern (Quarz) oder 8 MHz intern (RC-Oszillator) getaktet wird. Er verfügt über 25 I/O-Ports, von denen immerhin 23 frei nutzbar sind als Eingang, Ausgang sowie für Sonderfunktionen wie z. B. PWM-Ausgang (acht PWM-Kanäle verfügbar). Die beiden restlichen I/O-Ports dienen als ADC-Port für den internen 10-Bit-A/D-Konverter. Zur Kommunikation besitzt der AVR eine UART-Schnittstelle, die einerseits per Bootloader das einfache Programmieren des AVR und andererseits den Datenverkehr „nach außen“ ermöglicht.

Der ATmega88 besitzt 8 KByte Flash-Programmspeicher, 512 Byte EEPROM und 1 KByte SRAM. Damit ist er auch für längere Programme und den vorgesehenen Einsatzzweck gut mit Speicher bestückt.

Bootloader

Wie bereits kurz angerissen, wird der ATmega88 bereits mit einem vorinstallierten Bootloader ausgeliefert. Zusammen mit der Voreinstellung des Bootloader-Fusebits fängt also der Controller nach dem Start an, das Bootloader-Programm abzuarbeiten, das den Controller in die Lage versetzt, direkt über USB und UART mit dem BASCOM-Programm programmiert zu werden.

Es geht los!

Genug der Vorreden, Lernpaket kommt von Lernen, also fangen wir an. Wie bei den meisten USB-Geräten muss auch für die Experimentierplatine zunächst ein Treiber für den USB-Port installiert werden. Dazu sollte man sicherstellen, dass kein weiteres USB-Gerät mit FT232-Chip an den PC angeschlossen ist.

Nach dem Einlegen der mitgelieferten CD erscheint deren Hauptmenü (Abbildung 1). Hieraus ist zunächst die Installation des FTDI-Treibers zu starten. Nach Anschließen der Experimentierplatine an den PC erkennt Windows diese als neues USB-Gerät, und im Geräte manager von Windows erscheint dessen Schnittstelle als virtueller COM-Port (USB Serial Port), z. B. als COM5.

Um den USB-Chip zu konfigurieren, befindet sich ein sehr interessantes und auch für andere Projekte sehr gut nutzbares EEPROM-Programmiersystem, „MProg“, auf der CD (Abbildung 2). Dieses ist zu installieren und die Einstellungen sind in wenigen, im Buch gut beschriebenen Schritten vorzunehmen. So wird u. a. festgelegt, dass die Betriebsspannungsversorgung via USB erfolgt und der USB-Chip als reines I/O-Gerät (Simulation eines seriellen Kabels) arbeitet. Nun ist nur noch BASCOM zu installieren. Nach einem Neustart des PCs kann es dann losgehen.

BASCOM-AVR

Bevor wir beginnen, noch ein paar Worte zu BASCOM-AVR. BASCOM ist ein von der niederländischen Firma MCS [1] vertriebener BASIC-Compiler, der als Cross-Compiler in der Programmiersprache BASIC geschriebene Programme in einen durch den Mikrocontroller ausführbaren Maschinencode um-



Bild 1: Das Hauptmenü der Begleit-CD mit allen mitgelieferten Software-Bestandteilen des Lernpakets

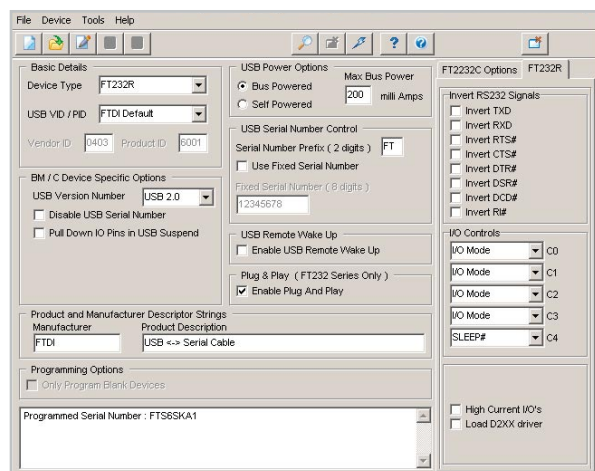


Bild 2: Über MProg kann der FTDI-Chip seiner Aufgabe gerecht angepasst werden.

setzt. Zusätzlich erlaubt auch dieser BASIC-Dialekt für besonders maschinennahe und zeitkritische Aufgaben die Einbettung von Assembler-Routinen.

Als komplettes Programmiersystem verfügt BASCOM aber auch über eine integrierte Entwicklungsumgebung, ein Terminalprogramm und einen AVR-Simulator sowie einige weitere nützliche Werkzeuge, z. B. zur Gestaltung eigener LCD-Layouts. Der Vorteil von BASIC als Programmiersprache ist allgemein bekannt: sie ist leicht erlernbar. Neben dem reinen BASIC-Dialekt bietet BASCOM aber auch komplexe, auf die jeweilige Prozessorfamilie zugeschnittene Routinen, die über einen einfachen Befehl zugänglich sind und den Programmierer von der in manchen anderen Programmiersprachen notwendigen „Code-Ochsentour“ entlasten. So kommt man naturgemäß schneller zum Erfolg eines fertigen Programms und kann sich mehr der eigentlichen Lösung der Programmieraufgabe widmen. Derartige komplexe Abläufe sind in BASCOM, ähnlich wie in C, als Bibliotheken verfügbar.

Die auf der Franzis-CD enthaltene Version von BASCOM ist die Demo-Version, deren einzige Beschränkungen gegenüber der 89 Euro kostenden Vollversion die Begrenzung auf 4 KByte Programmcode (für die meisten Aufgaben vollkommen ausreichend) und der Ausschluss der Xmega-AVRs sind. Es lohnt sich, die MCS-Seite ab und an zu besuchen, es gibt

immer einmal Upgrades und neue Bibliotheken, zudem ist sie der Ausgangspunkt zur internationalen Nutzergemeinschaft, die auch in Deutschland zahlreiche Foren und Webseiten unterhält. Es gibt hier quasi alles an Wissen und Programmbausteinen, was man sich nur denken kann, man muss oft seine eigene Lösung nur noch an bereits vorhandene Programmbausteine anpassen.

Da ist die deutsche Befehlsbeschreibung des Franzis-Buchs mit zahlreichen Beispielen, sowohl im Buch als auch sofort nachvollziehbar auf der CD, eine perfekte Ergänzung für den Einsteiger, kann er sich doch das Ganze in Deutsch erklärt unmittelbar am Objekt erarbeiten. So lernt man wirklich jeden Befehl und seine Auswirkung perfekt kennen.

BASCOM-AVR-IDE

Startet man das Programm, empfängt uns die sogenannte IDE, also die Entwicklungsumgebung (Abbildung 3, hier ist zur Demonstration schon ein Programm aus dem Einsteigerlehrgang geladen). Das große Fenster links ist der Programmeditor, darüber befindet sich die Werkzeugleiste (Toolbar), die die meisten Funktionen des darüber liegenden Menüs unmittelbar zugänglich macht. Rechts kann man entweder einen Adobe-PDF-Viewer aufrufen, über den man sich z. B. die auf der CD vorhandenen Datenblätter und weitere Informationen aller beteiligten Bauelemente aufrufen kann, oder – für das unmittelbare Programmieren noch interessanter – die Pinouts der jeweils für das Programm zu benutzenden Controller. Der Clou daran: Klickt man einen Pin des Pinouts an, erscheinen dessen Funktionen unmittelbar im Fenster darunter.

Zunächst erscheint die Programmoberfläche in Englisch, aber über das Optionsmenü ist sie auf Deutsch umstellbar. Zur Arbeit mit der Entwicklungsumgebung sind nur wenige Einstellungen wie die des Chip-Typs (Abbildung 4), der zu erzeugenden Files, des Kommunikationsweges und des Programmiergerätes – hier des Bootloaders – nötig. Dann kann es unmittelbar ans Programmieren gehen. Hier kommt wieder die Stärke des Begleitbuchs zum Tragen: Der Autor erklärt wirklich Befehl für Befehl, was Programmzeile für Programmzeile geschieht. Etwas derart Einsteigerfreundliches haben wir bisher nur bei unserem myAVR-Kurs [2] erfahren, bei dem es ja seinerzeit um die Assemblerprogrammierung ging.

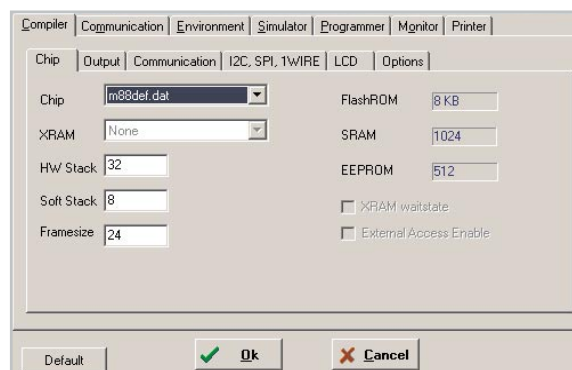


Bild 4: Eines der Konfigurationsfenster von BASCOM, hier die Einstellung des benutzten Controllers

Der Einsteigerkurs

Natürlich beginnt der Autor mit dem obligatorischen „Hallo-Welt“-Programm (siehe Abbildung 3), für das man nicht einmal irgendetwas auf dem Experimentierboard verdrahten muss, mit dem Programm bringt man lediglich die LED PB6 auf dem Board zum Blinken. Die darauf im Buch folgende Programmerläuterung sucht ihresgleichen an einfacher Verständlichkeit.

Noch ein Wort zu den Beispielprogrammen. Die des Buches befinden sich nicht im umfangreichen Samples-Ordner von BASCOM, sondern als Beispiel-Ordner auf der CD. Zumindest diesen sollte man sich in ein entsprechendes BASCOM-Verzeichnis auf dem PC kopieren, damit man sie auch editieren und kompilieren kann.

Dann geht es los mit dem eigentlichen Kennenlernen von BASCOM. Nach einem kurzen Exkurs durch den grundsätzlichen Aufbau eines Programms schreiben wir unser erstes „richtiges“ Programm (Abbildung 5), ein kleines Rechenprogramm, das wieder ausführlich kommentiert wird. Jetzt geht es in die einzelnen Bestandteile der Programmiersprache: Datentypen, Variable, Arrays, Operatoren, Strukturen, Befehle, Konfigurationen – immer einzeln und immer mit nachvollziehbarem Programmbeispiel. Auch speziellen Programmieraufgaben wie der Timer-Programmierung, dem Umgang mit dem Analog-Digital-Wandler (ADC), Interrupt-Behandlung, UART-Schnittstelle widmet sich der Kurs.

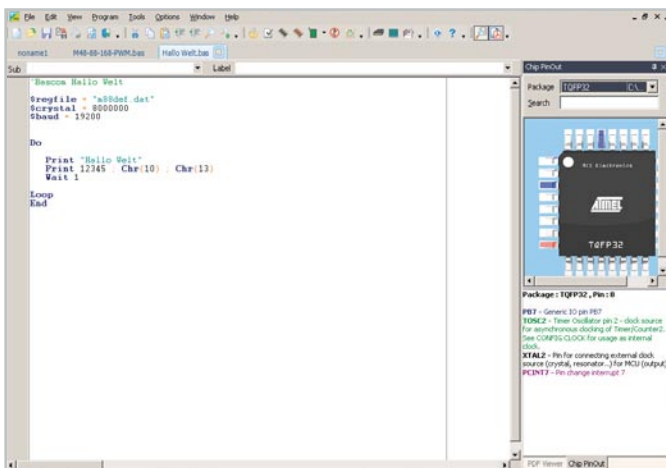


Bild 3: Die Entwicklungsumgebung (IDE): rechts die praktische Erläuterung der Pinbelegung des AVR-Controllers

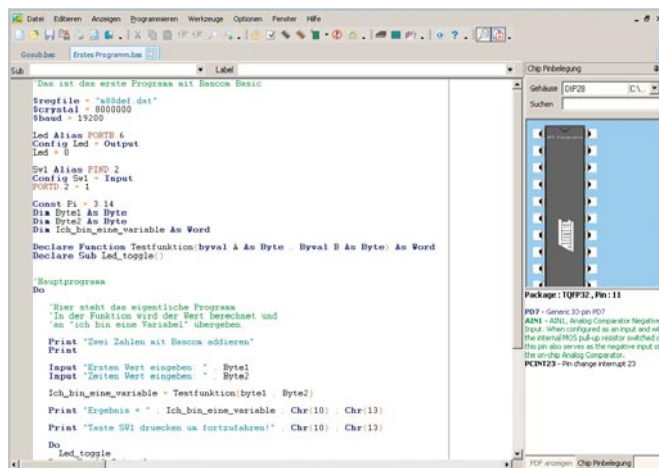


Bild 5: Übersichtlich, gut kommentiert und im Handbuch ausführlich erklärt – das erste Rechenprogramm

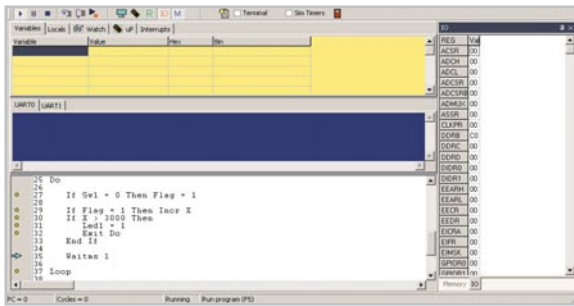
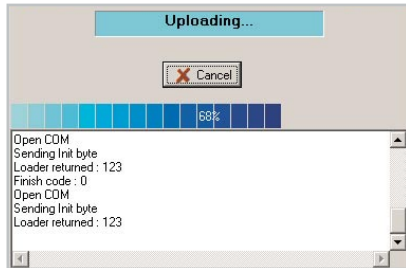


Bild 6: Der komfortable Simulator von BASCOM – mit Terminal, Hardware-Simulation und genauen Register- und Speicheransichten

Bild 7: So einfach geht das Programmieren des AVR dank Bootloader – ein Klick genügt.



Simulieren, Kompilieren, Programmieren
 BASCOM verfügt über einen eigenen Programmsimulator, der es einfach macht, ein geschriebenes und kompiliertes Programm direkt zu testen, bevor es auf den Controller geschrieben wird. Er erlaubt auch eine Hardware-Simulation, z. B. des A/D-Konverters oder einer LCD-Anzeige, und enthält ein Mini-Terminal-Programm zur Werteausgabe und für Eingaben. Abbildung 6 zeigt den Simulator in Aktion. Hat man per Simulator das Programm überprüft (kein Muss, aber auch hier lernt man viel über die Funktion des AVR), kann es kompiliert und über die Option „zum Chip senden“ auf den AVR programmiert werden (Abbildung 7). Noch einmal sei daran erinnert – das Kompilieren funktioniert nur, wenn das .bas-Programm zuvor auf der Rechner-Festplatte gespeichert wurde. Direktladen von der CD und kompilieren geht nicht.

Experimente mit dem AVR-Board

Nach dem Einsteigerkurs geht es in die echte Praxis, beginnend mit ein paar wichtigen Standard-Grundlagen wie Tastenentprellung per Software. Aber schnell folgen dann die ersten Experimente, wo man wirklich etwas sieht und hört, mit Ausgaben auf externe LEDs, den Signalgeber, mit Blitzschaltungen, Messschaltungen usw. Einige Schaltungen und Programme demonstrieren die Zusammenarbeit mit dem Terminalprogramm auf dem PC (Abbildung 8 zeigt Beispiele für eine Pulslängenmessung, eine Kapazitätsmessung, einen bereits recht komfortablen Datenlogger und ein kleines Ratespiel). Hier kann man dann unmittelbar mit Hilfe fertiger und nur noch einzubindender Bibliotheken auch ein LC-Display (Beispiel direkt unter [1] zu finden) oder eine LED-Multiplexanzeige anbinden. Sehr nützlich für spätere eigene Programmieraufgaben: die praktische Einführung in die Nutzung der PWM-Ports und des ADC.

Zum Schluss gibt es dann noch echte Leckerbissen, nämlich professionelle Visualisierungen von Messwerten mittels MS-Visual-Basic-Express-Anwendungen, so ein 8-Kanal-Voltmeter (Abbildung 9) oder ein Speicheroszilloskop. Und auch die

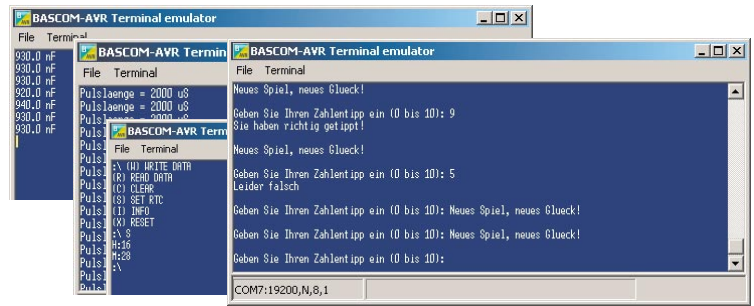


Bild 8: Das integrierte Terminalprogramm ist ein idealer Helfer für Ein- und Ausgaben der Beispielprogramme.

Bild 9: Messdaten aus dem AVR – 8-Kanal-Voltmeter, visualisiert mittels Visual Basic.

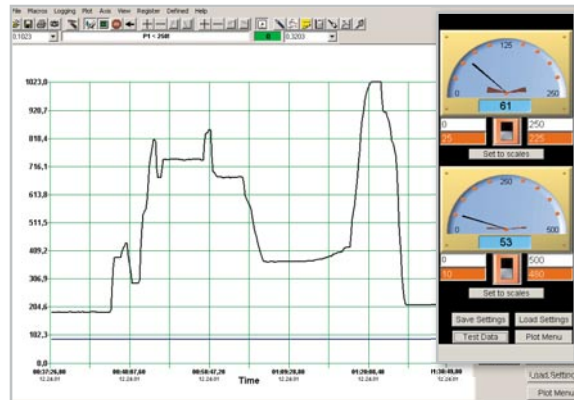


Bild 10: Perfekte Messdaten-Visualisierung mit StampPlot

Steuerung des Controllers via VB-Net ist im Buch beschrieben. Aber es geht noch besser – auf der CD findet sich ein echter Profi-Datenlogger-Visualisierer, das Programm StampPlot. Hier müssen Sie nur noch festlegen, in welcher grafischen Form die vom AVR gelieferten Messdaten dargestellt werden sollen, und schon haben Sie eine wirklich professionelle Messwertdarstellung (Abbildung 10) auf dem Schirm! Ist man hier angekommen, hat man den Einstieg in die professionelle AVR-Programmierung schon vor Augen, und spontan fallen einem weitere Anwendungen und eigene Applikationen ein. Für die ist das vorhandene Software-Equipment eine hervorragende Voraussetzung, für die eigenen Applikationen mit einer Eigenbau-AVR-Schaltung ist dann nur noch ein ISP-Programmer oder ebenfalls das Installieren des Bootloaders nötig, alles ebenfalls im Buch beschrieben. Fazit: Empfehlung. Das Lernpaket ist genau das richtige Werkzeug für den einfachen Einstieg in die AVR-Programmierung bis hin zur Erarbeitung eigener Hard- und Software-Applikationen!

Links:

[1] www.mcselec.com

[2] www.elv.de/output/controller.aspx?cid=588&detail=1&detail2=73