

1 Vorwort

Atmel, ursprünglich bekannt als Hersteller nichtflüchtiger Speicher, wie EPROMs, EEPROMs und Flash-Speicher, ergänzte schon vor vielen Jahren seine Flash-Mikrocontroller auf der Basis von Intels 8051-Familie durch die Familie der AVR-Mikrocontroller, die äußerst populär geworden sind.

Die 8-Bit-RISC-Architektur wurde vom norwegischen Designhaus Nordic VLSI in Trondheim in enger Zusammenarbeit mit dem schwedischen Compilerhersteller IAR entwickelt.

Später kaufte Atmel dieses europäische Design, um ergänzt durch selbst entwickelte On-Chip Peripherie eine eigene Mikrocontroller-Familie auf den Markt zu bringen. Die Familie der AVR-Mikrocontroller konkurriert mit einigen gut eingeführten Familien auf dem Markt, wie Intels 8051 mit seinen vielen Derivaten unterschiedlicher Hersteller, Motorolas 6805 und 68HC11 und Microchips PICmicros.

Die Familie der AVR-Mikrocontroller besteht heute aus einer Vielzahl unterschiedlich ausgestatteter Bausteine. Mittlerweile gibt es zehn Produktgruppen – Automotive AVR, AVR Z-Link, CAN AVR, LCD AVR, Lighting AVR megaAVR, Smart Battery AVR, tinyAVR, USB AVR und XMEGA. Nicht alle können hier in die Betrachtungen einbezogen werden.

Die Webseite www.atmel.com/dyn/products/devices.asp?family_id=607 liefert eine Übersicht, über die aktuell von Atmel angebotenen AVR-Mikrocontroller.

Atmel stellt eine Reihe der zur Programmentwicklung für diese Mikrocontroller erforderlichen Tools gratis zur Verfügung. Programmieren in Assembler und Simulieren der Programme sind nach einem Gratis-Download von der Atmel Website sofort möglich. Verschiedene Starterkits werden zu sehr günstigen Preisen angeboten und durch die Möglichkeit der In-System Programmierung kann ein solches Kit auch schon mal als Programmierer verwendet werden.

Will man nicht den mitunter steinigem Weg der Programmierung in Assembler gehen, dann sucht man nach einer Möglichkeit, die AVR-Mikrocontroller in einer komfortableren Hochsprache programmieren zu können.

Neben Programmierumgebungen, wie sie beispielsweise durch IAR, oder E-LAB Computers für den eher professionellen Einsatz angeboten werden, können sich auch kostengünstigere und in vielen Belangen völlig ausreichend ausgestattete Entwicklungsumgebungen behaupten.

BASCOM-AVR - eine BASIC Entwicklungsumgebung für die AVR-Mikrocontroller - ist ein solches Beispiel. BASCOM-AVR ist mittlerweile breit eingeführt, vom Hersteller bestens unterstützt und soll in diesem Buch anhand von Anwendungen vorgestellt werden.

Mit BASCOM-AVR ist die Programmierung der AVR-Mikrocontroller nicht mehr den Software-spezialisten überlassen. Nach kurzer Einarbeitungszeit kann auch der Hardware-spezialist, Messtechniker oder Roboterfreak eigene Programme erstellen.

BASCOM-AVR erzeugt kompilierten Code. Der bei Interpretern übliche Overhead fällt weg. Die erzeugten Programme erfüllen damit die meisten Echtzeitanforderungen, ohne Assembler einsetzen zu müssen.

In dieser dritten, im Herbst 2010 fertiggestellten Auflage kommt BASCOM-AVR in der Version 1.12.0.0 zum Einsatz. Das Evaluationsboard STK500 von Atmel bildet zusammen mit dem

AVR Studio Atmels Integrierte Entwicklungsumgebung (Integrated Development Environment - IDE) für die Codeentwicklung und das Debugging. Auch der mit BASCOM-AVR erzeugte Code kann auf dem STK500 getestet werden. Erweiterungskarten zum STK500 dienen als Interface zu verschiedenen AVR Sub-Familien oder größeren Bausteinen. Es gibt aber zahlreiche einfache und preiswerte Boards, die nach der Entscheidung für einen bestimmten AVR Mikrocontroller ebenso eingesetzt werden können.

Es ist mir wichtig, einigen Personen an dieser Stelle Dank zu sagen:

- An erster Stelle Mark Alberts von MCS Electronics, der die BASCOM Programmierumgebung mit einem herausragenden Preis-Leistungs-Verhältnis entwickelt hat und ständig weiterentwickelt,
- Atmel für die Entwicklung und ständige Weiterentwicklung der AVR-Mikrocontroller, die neue Leistungsmerkmale in die Familie der Mikrocontroller eingebracht haben und
- Lars Wictorsson von LAWICEL für die Entwicklung der CANDIP Mikrocontrollermodule mit CAN Interface auf der Basis des AT90S8515 bzw. ATmega162 und der Mikrocontrollermodule stAVeR-24 bzw. stAVeR-40.