

# A0 Vorwort

Das folgende Lehrbuch für Mikrocontrollertechnik soll das Selbststudium ermöglichen, dient aber auch als Kurs für eine Abschlussklasse in der Techniker-Ausbildung in Luxemburg.

Das Lehrbuch gliedert sich in drei Module (A, B und C), die jeweils aufeinander aufbauen. Ein viertes Modul wird Projekte enthalten. Es werden abwechselnd theoretische und praktische Kapitel behandelt, wobei der Schwerpunkt auf den praktischen Programmieraufgaben liegt. Die theoretischen Kapitel dienen als Lernstoff zur Wissenskontrolle. Einige Kapitel sind recht ausführlich und dienen dann als eher Referenz (z.B. Adressiermodi). Aus diesen sind dann nur die zum jeweiligen Zeitpunkt benötigten Teile zu erlernen.

Im praktische Teil sind Aufgaben und Text eng verwoben, so dass der größte Teil der Aufgaben gelöst werden muss um den Stoff zu erschließen. Es wird das Prinzip "**learning by doing**" angewendet, da dies die Kompetenz zum Problemlösen verbessert und nur selbst Erarbeitetes wirklich über einen längeren Zeitraum behalten wird.

Dies ist auch der Grund, warum keine Lösungen zu den Aufgaben veröffentlicht werden. Die Gefahr ist zu groß, dass der Lernende glaubt, alleine durch Lesen der Lösung könnte er den Stoff verinnerlichen.

Ein umfangreicher Anhang, der beim Programmieren immer griffbereit sein sollte, schließt den Kurs ab.

Im Kurs wird der Baustein ATmega32 bzw. ATmega32A<sup>1</sup> aus der 8-Bit-AVR<sup>®</sup>-Familie der Firma ATMEL<sup>®</sup> verwendet. Allerdings sollten auch alle Programme auf dem kleineren Controller ATmega8A funktionieren. Eventuell sind die Ports anzupassen<sup>2</sup>. Der ATmega16A ist pinkompatibel mit dem ATmega32A und kann natürlich auch verwendet werden. Der ältere AT90S8535 (bzw. die neuere Version ATmega8535) ist auch pinkompatibel und er kann für viele der Aufgaben eingesetzt werden. Allerdings sind einige Register geändert worden (z.B. Timer) so dass leichte Änderungen im Code nötig sind.

In den ersten Kapiteln werden die Bausteine mit dem internen Takt von 1 MHz betrieben, so wie sie auch vom Werk aus programmiert wurden. In einem späteren Kapitel wird dann ein externer Quarz mit 16 MHz verwendet.

Für die Erstellung des Kurses wurden hauptsächlich die Datenblätter von ATMEL<sup>®</sup>, die freie Enzyklopädie Wikipedia und die folgenden Bücher zu Rate gezogen:

G. Schmit      Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie,  
2 Auflage, Oldenburg, ISBN 3-486-58016

R. Walter      AVR Mikrocontroller Lehrbuch, 2 Auflage, [www.rowalt.de](http://www.rowalt.de)

1 Bei den Bausteinen mit dem Zusatz A handelt es sich um die neuere stromsparendere Variante. Diese sind vollständig kompatibel zu den älteren Varianten ohne A.

2 Aufpassen muss man auch, wenn man nicht mit der Definitionsdatei arbeitet, da zum Beispiel die Interruptvektortabellen unterschiedlich sind (gilt auch für den ATmega16)!

W. Tampert AVR-RISC-Mikrocontroller, 2 Auflage, Franzis, ISBN 3-7723-5476-9

Mit Sicherheit befinden sich "Fähler" ;-)) in diesem Kurs. Mit der Zeit werden es hoffentlich immer weniger. Sollte Ihnen ein Fehler auffallen so würde ich mich über eine Benachrichtigung freuen. Meine E-Mail-Adresse lautet: "weigu@weigu.lu". Vielen Dank im Voraus.

## **Lernen macht Spaß!**

Auch wenn nicht jeder mit dieser Erkenntnis einverstanden ist, so war es doch die treibende Kraft, die diesen Kurs entstehen ließ.

Genau diesen Spaß wünsche ich allen Lesern die diesen Kurs erarbeiten.

Guy WEILER  
([www.weigu.lu](http://www.weigu.lu))

## Darstellung

Für die Darstellung von Hexadezimalzahlen wird die in C übliche Schreibweise mit vorgestelltem **0x** verwendet (Bsp.: **0x5A**)<sup>3</sup>. Zahlen ohne Zusatz sind Dezimalzahlen. Für Binärzahlen wird die Schreibweise mit vorgestelltem **0b** (Bsp.: **0b01110111**) verwendet. ASCII-Zeichen und Strings werden zwischen Hochkomma gesetzt (Bsp.: **'A'**).

Englische Worte werden kursiv (*italic*) dargestellt. Assembler ist grau unterlegt. **Wichtiger Text** ist ohne Serifen und fett (*bold*) dargestellt. **Arbeitsregister**, **SF-Register**, **Flags** (einzelne Bits in SF-Registern) und **Pins** des Controller sind in unterschiedlichen Farben dargestellt.

Es werden die neueren auf den Zweierpotenzen beruhenden Binärpräfixe "kibi", "mebi", "gibi", "tebi" für binäre Vielfache von Einheiten verwendet. Die vorgestellten Symbole sind Ki, Mi, Gi und Ti.

Beispiel: Ein Kibibit wird mit dem Symbol Kibit oder manchmal Kib bezeichnet (Achtung: Großbuchstabe K, Kleinbuchstabe b für bit) und sind 1024 Bit. Ein Gigibyte wird mit dem Symbol GiB bezeichnet (Achtung: Großbuchstabe G, Großbuchstabe B für Byte) und sind 2<sup>30</sup> Byte.

Name	Symbol	Wert
kibi	Ki	2 <sup>10</sup> = 1.024
mebi	Mi	2 <sup>20</sup> = 1.048.576
gibi	Gi	2 <sup>30</sup> = 1.073.741.824
tebi	Ti	2 <sup>40</sup> = 1.099.511.627.776

Weitere Informationen auf: <http://de.wikipedia.org/wiki/Bit> und <http://de.wikipedia.org/wiki/Byte>

<sup>3</sup> In PASCAL und BASIC-Programmen wird für Hexzahlen meist ein Dollarzeichen vorgestellt (Bsp.: \$5A). Im Text wird bei Hexzahlen auch manchmal statt einem vorgestellten Zeichen ein "h" nachgestellt (Bsp.: 5Ah)

