

# AVR-8-bit-Mikrocontroller

## Arbeiten mit AVR-Assembler

### Teil 00 - Inhaltsangabe

#### Teil 01 - Einführung

- 1 Eine Einführung in die Assembler-Sprache
  - 1.1 Warum Assembler ?
  - 1.2 In der Kürze liegt die Würze - Schnell wie der Wind

#### Teil 02 - Erste Schritte

- 2 Grundausstattung und Installation der Werkzeuge
  - 2.1 Testboard, Mikrocontroller, Programmer
  - 2.2 AVR Studio
    - 2.2.1 Der Editor
    - 2.2.2 Der Übersetzer (Assembler)
    - 2.2.3 Der Simulator

#### Teil 03 - Struktur und Syntax der Assembler-Sprache

- 3 Struktur und Syntax der Assembler-Sprache
  - 3.1 Struktur der Assembler-Zeilen
  - 3.2 Struktur der Programm-Datei
  - 3.3 Kommentare
  - 3.4 Kopf des Programms
  - 3.5 Assembler-Direktive (Directivs)
  - 3.6 Assembler-Ausdrücke (Expressions)
    - 3.6.1 Operanden
    - 3.6.2 Operatoren
    - 3.6.3 Funktionen
    - 3.6.4 Operationen
  - 3.7 Daten- und Adress-Struktur
    - 3.7.1 Nomenklatur: Abkürzungen für Register und Operanden
    - 3.7.2 AVR-Speicher-Räume aus Assembler-Sicht
      - 3.7.2.1 Programm-Speicher
      - 3.7.2.2 Daten-Speicher
      - 3.7.2.3 EEPROM-Speicher
    - 3.7.3 Adress-Modi
      - 3.7.3.1 Direkte Adressierung eines einzelnen Arbeits-Registers
      - 3.7.3.2 Direkte Adressierung von 2 Arbeits-Registern
      - 3.7.3.3 Direkte Adressierung eines I/O-Registers
      - 3.7.3.4 Direkte Daten-Adressierung im SRAM
      - 3.7.3.5 Indirekte Daten-Adressierung mit Displacement
      - 3.7.3.6 Indirekte Daten-Adressierung
      - 3.7.3.7 Indirekte Daten-Adressierung mit Pre-Decrement
      - 3.7.3.8 Indirekte Daten-Adressierung mit Post-Decrement
      - 3.7.3.9 Adressierung des Programm-Speichers
      - 3.7.3.10 Adressierung des Programm-Speichers mit Post-Increment
      - 3.7.3.11 Direkte Adressierung des Programm-Speichers
      - 3.7.3.12 Indirekte Adressierung des Programm-Speichers
      - 3.7.3.13 Relative Adressierung des Programm-Speichers

#### Teil 04 - Instruktionen

- 4 Mnemotechnische Abkürzungen der Instruktionen
  - 4.1 Arithmetische und logische Instruktionen
  - 4.2 Verzweigungs-Instruktionen
  - 4.3 Instruktionen für den Daten-Transfer
  - 4.4 Bit- und Bit-Test-Instruktionen

#### Teil 05 - Register

- 5 Register-Datei
  - 5.1 Arbeits-Register
  - 5.2 Unterschiede der Arbeits-Register
  - 5.3 Pointer-Register
  - 5.4 Empfehlungen zu Registerwahl

**AVR-8-bit-Mikrocontroller**  
**Arbeiten mit AVR-Assembler**  
**Teil 00 - Inhaltsangabe**

**Teil 06 - I/O-Ports**

- 6 Die Ein-/Ausgabe-Ports der AVR's
  - 6.1 Die Ports der AVR's
  - 6.2 Die Ports als I/O-Register
  - 6.3 Das Status-Register SREG

**Teil 07 - Statisches RAM**

- 7 Statisches RAM - SRAM
  - 7.1 Beschreibung
  - 7.2 Verwendung von SRAM
  - 7.3 Zugriff auf das SRAM
  - 7.4 Verwendung des SRAM als Stack
  - 7.5 Einrichtung des Stapels
  - 7.6 Verwendung des Stapels
  - 7.7 Fehlermöglichkeiten beim (Hoch-)Stapeln

**Teil 08 - Programm-Ablaufsteuerung**

- 8 Steuerung des Programmablaufs
  - 7.1 Der Reset
  - 7.2 Linearer Programmablauf und Verzweigungen
  - 7.3 Zeitzusammenhänge beim Programmablauf
  - 7.4 Makros im Programmablauf
  - 7.5 Unterprogramme (Subroutines)
  - 7.6 Interrupts im Programmablauf

**Teil 09 - Zahlendarstellungen**

- 9 Zahlendarstellungen im Assembler
  - 9.1 Dual-Zahlen
  - 9.2 Hexadezimal-Zahlen im ASCII-Format
  - 9.3 Oktal-Zahlen
  - 9.4 BCD-Zahlen (binär kodierte Dezimal-Zahlen)
  - 9.5 Zahlen im ASCII-Format
  - 9.6 Gleitkomma-Zahlen

**Teil 10 - Bitmanipulationen**

- 10 Umwandlungen von Bits, Bytes und Zahlen
  - 10.1 Bitmanipulationen, Schieben und Rotieren
  - 10.2 Umwandlung einer Dual-Zahl in eine ungepackte BCD-Zahl
  - 10.3 Umwandlung einer BCD-Zahl in eine Dual-Zahl
  - 10.4 Umwandlung einer Dual-Zahl in eine Dezimal-Zahl im ASCII-Format
    - 10.4.1 Lösung 1
    - 10.4.2 Lösung 2
    - 10.4.3 Lösung 3
  - 10.5 Umwandlung einer Dezimal-Zahl im ASCII-Format in eine Dual-Zahl
  - 10.6 Umwandlung einer Dual-Zahl in eine Hexadezimal-Zahl im ASCII-Format
  - 10.7 Umwandlung einer Hexadezimal-Zahl im ASCII-Format in eine Dual-Zahl
  - 10.8 Lineare Umwandlung

**Teil 11 - Rechnen in Assembler**

- 11 Rechnen in Assembler
  - 11.1 Addition, Subtraktion und Vergleich
  - 11.2 Multiplikation
  - 11.3 Division

# AVR-8-bit-Mikrocontroller

## Arbeiten mit AVR-Assembler

### Teil 00 - Inhaltsangabe

#### Vorbemerkung

Nichts ist vollkommen - und nichts ist endgültig! So auch nicht dieses Tutorial! Deshalb bitte immer erst nach dem neuesten Datum schauen. Vielleicht gibt es wieder etwas Neues oder eine Fehlerbereinigung oder eine etwas bessere Erklärung. Wer Fehler findet oder Verbesserungen vorzuschlagen hat, bitte melden ([info@alenck.de](mailto:info@alenck.de)).

Immer nach dem Motto: Das Bessere ist Feind des Guten und nichts ist so gut, dass es nicht noch verbessert werden könnte.

Bild-, Beispiel-, Form- und Tabellen-Nummern sind nach folgendem Schema aufgebaut, damit bei Einfügungen/Löschungen nicht alle Nummern wieder geändert werden müssen (hier bunt dargestellt):

Darstellungsart	Abschnitt-LfdNummer:	Beschreibung	allgemeines Schema - Beispiele:
•	<b>Bild 5.1.4-02:</b>	<b>Daten-Adress-Raum</b>	Benummerung eines Bildes
•	<b>Beispiel 5.1.4-03:</b>	<b>EEPROM-Speicherung</b>	Benummerung eines Beispiels/-gruppe
•	<b>Form 5.1.3-01:</b>	<b>Die <code>main</code>-Funktion</b>	Benummerung einer Formdarstellung
•	<b>Tabelle 5.1.4-01:</b>	<b>Schlüsselwörter vom CVAVR</b>	Benummerung einer Tabelle