

**AVR-8-bit-Mikrocontroller**  
**Gruppe 100 - Technologie der AVR-8-bit-Mikrocontroller**  
**Teil 101 - AVR-Architektur**



### **Teil 101 - AVR-Architektur**

#### 1 Die Architektur der AVR-Mikrocontroller - der CPU-Kern

- 1.1 Einleitung und Geschichtliches
- 1.2 Die Architektur: RISC contra CISC
- 1.3 ALU - Arithmetic Logic Unit
- 1.4 Das Status Register
- 1.5 Die Arbeits-Register
- 1.6 Der Stack Pointer
- 1.7 Zeitliche Steuerung der Verarbeitung
- 1.8 RESET- und Interrupt-Verwendung

### **Teil 102 - Harvard-Architektur**

#### 2 Die Harvard-Architektur - eine Speicher-Philosophie

- 2.1 Flash-Programm-Speicher
- 2.2 SRAM-Daten-Speicher
- 2.3 EEPROM-Daten-Speicher
- 2.4 Ein-/Ausgabe-Speicher
- 2.5 Beschreibung der Register
  - 2.5.1 **EEARH** und **EEARL** - **EEPROM Address Register (Low/High)**
  - 2.5.2 **EEDR** - **EEPROM Daten Register**
  - 2.5.3 **EECR** - **EEPROM Control Register**
  - 2.5.4 **GPIOR2** - **General Purpose I/O Register 2**
  - 2.5.5 **GPIOR1** - **General Purpose I/O Register 1**
  - 2.5.6 **GPIOR0** - **General Purpose I/O Register 0**

### **Teil 103 - Clock**

#### 3 Der Systemtakt und seine Optionen

- 3.1 Gliederung der Takt-Systeme (Clock Systems)
- 3.2 Möglichkeiten der Takt-Erzeugung (Clock Sources)
  - 3.2.1 Quarz-Oszillator mit verminderter und voller Amplitude
  - 3.2.2 Quarz-Oszillator mit niedriger Frequenz
  - 3.2.3 Kalibrierter interner RC-Oszillator
  - 3.2.4 Interner Oszillator mit 128 kHz
- 3.3 Externe Takt-Erzeugung
- 3.4 Timer/Counter Oscillator
- 3.5 Systemtakt-Teiler (Prescaler)
- 3.6 Register für die Einstellung/Teilung des Systemtaktes
  - 3.6.1 **OSCCAL** - **OSCillator CALibration Register**
  - 3.6.2 **CLKPR** - Register für die Teilung des Systemtaktes (**CLock PR**escaler)

### **Teil 104 - Sleep Modes**

#### 4 Das Power-Management - Leistungsreduzierung (Sleep Modes)

- 4.1 Leerlauf-Modus (Idle Mode)
- 4.2 ADC-Rausch-Reduzierung (ADC Noise Reduction Mode)
- 4.3 Power Down Mode
- 4.4 Power Save Mode
- 4.5 Standby Mode
- 4.6 Minimierung der Leistung
- 4.7 Register für die Einstellung des Power-Managements
  - 4.7.1 **SMCR** - **Sleep Mode Control Register**
  - 4.7.2 **PRR** - **Power Reduction Register**

**AVR-8-bit-Mikrocontroller**  
**Gruppe 100 - Technologie der AVR-8-bit-Mikrocontroller**  
**Teil 101 - AVR-Architektur**

## Teil 105 - Systemsteuerung und RESET

### 5 Systemsteuerung und RESET

- 5.1 Rücksetzen des AVR
- 5.2 Die verschiedenen RESETS
  - 5.2.1 Power ON Reset
  - 5.2.2 Externer Reset
  - 5.2.3 Brown Out Erkennung - Internal Voltage Reference
  - 5.2.4 Watchdog System Reset - Watchdog Timer
- 5.3 Register für die Einstellung des Power-Managements
  - 5.3.1 **MCUSR** - **M**icro **C**ontroller **U**nit **S**tatus **R**egister
  - 5.3.2 **WDTCR** - **W**atchdog **T**imer **C**ontrol **R**egister

## Teil 106 - Interrupt-Vektoren

### 6 Die Interrupt-Vektoren

- 6.1 Das Konzept
- 6.2 Die Interrupt-Vektoren des **ATmega88**
- 6.3 Externe Interrupts
- 6.4 Register für die Steuerung der Interrupts
  - 6.4.1 **MCUCR** - **M**CU **C**ontrol **R**egister
  - 6.4.2 **EICRA** - **E**xternal **I**nterrupt **C**ontrol **R**egister **A**
  - 6.4.3 **EIMSK** - **E**xternal **I**nterrupt **M**aSK **R**egister
  - 6.4.4 **EIFR** - **E**xternal **I**nterrupt **F**lag **R**egister
  - 6.4.5 **PCICR** - **P**in **C**hange **I**nterrupt **C**ontrol **R**egister
  - 6.4.6 **PCIFR** - **P**in **C**hange **I**nterrupt **F**lag **R**egister
  - 6.4.7 **PCMSK2** - **P**in **C**hange **M**aSK **R**egister **2**
  - 6.4.8 **PCMSK1** - **P**in **C**hange **M**aSK **R**egister **1**
  - 6.4.9 **PCMSK0** - **P**in **C**hange **M**aSK **R**egister **0**

## Teil 107 - I/O Ports

### 7 Die Ein-/Ausgabeports (I/O-Ports)

- 7.1 Überblick
- 7.2 Ports zur allgemeinen Ein- und Ausgabe
- 7.3 Alternative Funktionen der Port-Pins
  - 7.3.1 Alternative Funktionen von Port B
  - 7.3.2 Alternative Funktionen von Port C
  - 7.3.3 Alternative Funktionen von Port D
- 7.4 Register für die Steuerung der Interrupts
  - 7.4.1 **MCUCR** - **M**CU **C**ontrol **R**egister
  - 7.4.2 **PORTB** - **P**ORT **B** **D**ata **R**egister
  - 7.4.3 **DDRB** - **P**ort **B** **D**ata **D**irection **R**egister
  - 7.4.4 **PINB** - **P**ort **B** **I**nput **P**INs **A**ddress
  - 7.4.5 **PORTC** - **P**ORT **C** **D**ata **R**egister
  - 7.4.6 **DDRC** - **P**ort **C** **D**ata **D**irection **R**egister
  - 7.4.7 **PINC** - **P**ort **C** **I**nput **P**INs **A**ddress **ss**
  - 7.4.8 **PORTD** - **P**ORT **D** **D**ata **R**egister
  - 7.4.9 **DDRD** - **P**ort **D** **D**ata **D**irection **R**egister
  - 7.4.10 **PIND** - **P**ort **D** **I**nput **P**INs **A**ddress

## Teil 108 - PWM

### 8 Timer/Counter, Prescaler und Pulsweiten-Modulation (PWM)

- 8.1 Einleitung
- 8.2 **8-bit** Timer/Counter**0** mit PWM
  - 8.2.1 Überblick
  - 8.2.2 Counter Unit
  - 8.2.3 Output Compare Unit
  - 8.2.4 Compare Match Output Unit
  - 8.2.5 Operations-Modi
  - 8.2.6 Zeit-Diagramme
  - 8.2.7 Register für die Steuerung der Timer/Counter**0**
    - 8.2.7.1 **TCCR0A** - **T**imer/**C**ounter**0** **C**ontrol **R**egister **A**
    - 8.2.7.2 **TCCR0B** - **T**imer/**C**ounter**0** **C**ontrol **R**egister **B**

**AVR-8-bit-Mikrocontroller**  
**Gruppe 100 - Technologie der AVR-8-bit-Mikrocontroller**  
**Teil 101 - AVR-Architektur**

- 8.2.7.3 **TCNT0** - Timer/CouNTER0 Register
- 8.2.7.4 **OCR0A** - Output Compare Register A vom Timer/Counter0
- 8.2.7.5 **OCR0B** - Output Compare Register B vom Timer/Counter0
- 8.2.7.6 **TIMSK0** - Timer/Counter0 Interrupt MaSK Register
- 8.2.7.7 **TIFR0** - Timer/Counter0 Interrupt Flag Register
- 8.3 **16-bit** Timer/Counter1 mit PWM
  - 8.3.1 Überblick
  - 8.3.2 Zugriff auf **16-bit** Register
  - 8.3.3 Counter Unit
  - 8.3.4 Input Capture Unit
  - 8.3.5 Output Compare Units
  - 8.3.6 Compare Match Output Unit
  - 8.3.7 Operations-Modi
  - 8.3.8 Zeit-Diagramme
  - 8.3.9 Register für die Steuerung der Timer/Counter1
    - 8.3.9.1 **TCCR1A** - Timer/Counter1 Control Register A
    - 8.3.9.2 **TCCR1B** - Timer/Counter1 Control Register B
    - 8.3.9.3 **TCCR1C** - Timer/Counter1 Control Register C
    - 8.3.9.4 **TCNT1H** und **TCNT1L** - Timer/CouNTER1
    - 8.3.9.5 **OCR1AH** und **OCR1AL** - Output Compare Register 1 A (Low/High)
    - 8.3.9.6 **OCR1BH** und **OCR1BL** - Output Compare Register 1 B (Low/High)
    - 8.3.9.7 **ICR1H** und **ICR1L** - Input Capture Register 1 (Low/High)
    - 8.3.9.8 **TIMSK1** - Timer/Counter1 Interrupt Mask Register
    - 8.3.9.9 **TIFR1** - Timer/Counter1 Interrupt Flag Register
- 8.4 Prescaler für Timer/Counter0 und Timer/Counter1
  - 8.4.1 **GTCCR** - General Timer/Counter Control Register
- 8.5 **8-bit** Timer/Counter2 mit PWM und asynchronen Operationen
  - 8.5.1 Überblick
  - 8.5.2 Counter Unit
  - 8.5.3 Output Compare Unit
  - 8.5.4 Compare Match Output Unit
  - 8.5.5 Operations-Modi
  - 8.5.6 Zeit-Diagramme
  - 8.5.7 Asynchrone Operation von Timer/Counter2
  - 8.5.8 Timer/Counter Prescaler
  - 8.5.9 Register für die Steuerung der Timer/Counter2
    - 8.5.9.1 **TCCR2A** - Timer/Counter2 Control Register 2 A
    - 8.5.9.2 **TCCR2B** - Timer/Counter2 Control Register 2 B
    - 8.5.9.3 **TCNT2** - Timer/CouNTER2 Register
    - 8.5.9.4 **OCR2A** - Output Compare Register A vom Timer/Counter2
    - 8.5.9.5 **OCR2B** - Output Compare Register B vom Timer/Counter2
    - 8.5.9.6 **TIMSK2** - Timer/Counter2 Interrupt MaSK Register
    - 8.5.9.7 **TIFR2** - Timer/Counter2 Interrupt Flag Register
    - 8.5.9.8 **ASSR** - ASynchronous Status Register
    - 8.5.9.9 **GTCCR** - General Timer/Counter Control Register

## Teil 109 - SPI

- 9 Das Serial Peripheral Interface (SPI)
  - 9.1 Überblick
  - 9.2 Funktionalität des Slave Select Pins
  - 9.3 Daten-Modi
  - 9.4 Register für die Steuerung des SPI
    - 9.4.1 **SPCR** - SPI Control Register
    - 9.4.2 **SPSR** - SPI Status Register
    - 9.4.3 **SPDR** - SPI Daten Register

## Teil 110 - USART

- 10 Der universelle synchrone/asynchrone serielle Receiver und Transmitter (USART)
  - 10.1 USART0
    - 10.1.1 Überblick
    - 10.1.2 Erzeugung des Taktes

# AVR-8-bit-Mikrocontroller

## Gruppe 100 - Technologie der AVR-8-bit-Mikrocontroller

### Teil 101 - AVR-Architektur

- 10.1.3 Formate der Signale
- 10.1.4 Initialisierung des USART
- 10.1.5 Daten senden - USART Transmitter
- 10.1.6 Daten empfangen - USART Receiver
- 10.1.7 Asynchrone Datenübertragung
- 10.1.8 Kommunikations-Modus im Multi-Processing
- 10.1.9 Register für die Steuerung des USART
  - 10.1.9.1 **UDRn** - USART I/O Daten Register n
  - 10.1.9.2 **UCSRnA** - USART Control und Status Register n A
  - 10.1.9.3 **UCSRnB** - USART Control und Status Register n B
  - 10.1.9.4 **UCSRnC** - USART Control und Status Register n C
  - 10.1.9.5 **UBRRnL** und **UBRRnH** - USART Baud Rate Register n (Low/High)
- 10.2 USART im SPI-Modus
  - 10.2.1 Überblick
  - 10.2.2 Erzeugung des Taktes
  - 10.2.3 SPI-Daten-Modi und Zeitsteuerung
  - 10.2.4 Formate der Signale
  - 10.2.5 Datenübertragung
  - 10.2.6 AVR USART **MSPIM** (Master SPI Mode) im Vergleich mit AVR SPI
  - 10.2.7 Register für die Steuerung des USART im SPI-Modus
    - 10.1.7.1 **UDRn** - USART MSPIM I/O Daten Register n
    - 10.1.7.2 **UCSRnA** - USART MSPIM Control und Status Register n A
    - 10.1.7.3 **UCSRnB** - USART MSPIM Control und Status Register n B
    - 10.1.7.4 **UCSRnC** - USART MSPIM Control und Status Register n C
    - 10.1.7.5 **UBRRnL** und **UBRRnH** - USART Baud Rate Register n (Low/High)

### Teil 111 - TWI

- 11 Das Serielle Interface für die 2-Draht-Übertragung (TWI)
  - 11.1 TWI-Bus-Definition
  - 11.2 Datentransfer und Formate der Signale
  - 11.3 Multi-Master - Bus-Systeme, Vermittlung und Synchronisation
  - 11.4 Überblick über die TWI-Module
  - 11.5 Gebrauch des TWI
  - 11.6 Datenübertragung
  - 11.7 Multi-Master-System
  - 11.8 Register für die Steuerung des TWI
    - 11.8.1 **TWBR** - TWI Bit Rate Register
    - 11.8.2 **TWCR** - TWI Control Register
    - 11.8.3 **TWSR** - TWI Status Register
    - 11.8.4 **TWDR** - TWI Daten Register
    - 11.8.5 **TWAR** - TWI (Slave) Address Register
    - 11.8.6 **TWAMR** - TWI (Slave) Address Mask Register

### Teil 112 - Analog-Komparator

- 12 Der Analog-Komparator
  - 12.1 Überblick
  - 12.2 Multiplexed Input
  - 12.3 Register für die Steuerung des AC
    - 12.3.1 **ADCSRB** - ADC Control und Status Register B
    - 12.3.2 **ACSR** - Analog Comparator Control und Status Register
    - 12.3.3 **DIDR1** - Digital Input Disable Register 1

### Teil 113 - ADC

- 13 Der Analog-Digital-Umsetzer (ADC)
  - 13.1 Überblick
  - 13.2 Start einer Umsetzung
  - 13.3 Takt-Steuerung (Prescaling und Conversion)
  - 13.4 Auswahl des Kanals und der Referenz-Spannung
  - 13.5 ADC-Rausch-Unterdrückung
  - 13.6 ADC-Ergebnis
  - 13.7 Register für die Steuerung des ADC

**AVR-8-bit-Mikrocontroller**  
**Gruppe 100 - Technologie der AVR-8-bit-Mikrocontroller**  
**Teil 101 - AVR-Architektur**

- 13.7.1 **ADMUX** - **ADC MU**ltipleXer Selection Register
- 13.7.2 **ADCSRA** - **ADC** Control und **Status Register A**
- 13.7.3 **ADCL** und **ADCH** - **ADC** Data Register (**Low/High**)
- 13.7.4 **ADCSRB** - **ADC** Control und **Status Register B**
- 13.7.5 **DIDR0** - **D**igital Input **D**isable Register **0**

### **Teil 114 - Bootloader**

#### 14 Bootloader-Unterstützung

- 14.1 Überblick
- 14.2 Applikationsbereich und Bootloader-Flash-Bereich
- 14.3 **RWW** - **R**ead **W**hile **W**rite und No Read While Write
- 14.4 Bootloader Lock-Bits
- 14.5 Laden des Bootloader-Programms
- 14.6 Flash-Adressierung während des Self Programming Mode
- 14.7 Self Programming des Flash-Speichers
- 14.8 Register für die Steuerung des Bootloaders
  - 14.8.1 **SPMCSR** - **S**tore **P**rogram **M**emory **C**ontrol und **S**tatus **R**egister

### **Teil 115 - Speicher-Programmierung**

#### 15 Speicher-Programmierung

- 15.1 Lock Bits im Lock Bit Byte (Lock Bit Protection Modes)
- 15.2 Fuse Bits
  - 15.2.1 Extended Fuse Byte
  - 15.2.2 Fuse High Byte
  - 15.2.3 Fuse Low Byte
- 15.3 Signature Bytes
- 15.4 Calibration Byte
- 15.5 Parallel (High Voltage) Programming
- 15.6 Serial Programming

# AVR-8-bit-Mikrocontroller

## Gruppe 100 - Technologie der AVR-8-bit-Mikrocontroller

### Teil 101 - AVR-Architektur

#### 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
•	<b>Bild 5.1.4-02: Daten-Adress-Raum</b>	Benummerung eines Bildes
•	<b>Beispiel 5.1.4-03: EEPROM-Speicherung</b>	Benummerung eines Beispiels
•	<b>Form 5.1.3-01: Die main-Funktion</b>	Benummerung einer Formdarstellung
•	<b>Tabelle 5.1.4-01: Schlüsselwörter vom CVAVR</b>	Benummerung einer Tabelle

#### Gravierende Änderungen gegenüber der Vorversion

1.

Völlig neue Strukturierung in **Gruppen** und **Teile**, um das Tutorial umfassend ordnen zu können. Die **Abschnitte** in den **Teilen** sind weitgehend erhalten geblieben.

Gruppenbezeichnung	Kurzbezeichnung
Gruppe 100: Technologie der AVR-8-Bit-Mikrocontroller	Technologie
Gruppe 200: Einsetzen von AVR-Tools	Tools
Gruppe 300: Arbeiten mit AVR-Assembler 3xx_Programm_yyyyy	ASM-Programmierung ASM-Programm-Beispiel
Gruppe 400: AVR-ASM-Projekte 4xx_Projekt_yyyyy	ASM-Projekte ASM-Projekt-Bezeichnung
Gruppe 500: CodeVisionAVR C-Compiler 5xx_Programm_yyyyy	C-Programmierung C-Programm-Beispiel
Gruppe 600: AVR-C-Projekte 6xx_Projekt_yyyyy	C-Projekte C-Projekt-Bezeichnung

**xx** steht für die laufende Nummer innerhalb des **Teils**, in dem das Programm bzw. das Projekt erscheint und **yyyyy** steht für die Programm- bzw. Projekt-Kurz-Bezeichnung.

2.

Notwendige Änderungen auf Grund Neuinstallation von **Windows 7**.

3.

**Windows 7** machte eine Installation von **CodeVisionAVR V2.60** als Vollversion notwendig. Daraus leiten sich auch viele Änderungen im Detail für die C-Programmierung (**Gruppe 500**) ab.

4.

Neu-Installation von **AVR Studio Vers. 4.19** unter **Windows 7**

5.

Zur Demonstration des Tools **AVR Studio** ist in **Gruppe 200** eine Trennung in **Teil 205 - Assembler und AVR Studio** und **Teil 206 - C-Compiler und AVR Studio** vorgenommen worden.

6.

**ASM-** und **C-Projekte** werden jeweils in eigenen Gruppen gesammelt (**Gruppe 400** für Assembler- und **Gruppe 600** für C-Projekte).

**AVR-8-bit-Mikrocontroller**  
**Gruppe 100 - Technologie der AVR-8-bit-Mikrocontroller**  
**Teil 101 - AVR-Architektur**

**IN BEARBEITUNG**