

Mikrocontroller- technik

Assembler
mit
ATMEL®-
AVR®-Mikrocontrollern

Guy WEILER

www.weigu.lu

Copyright ©

Das folgende Werk steht unter einer Creative Commons Lizenz (<http://creativecommons.org>). Der vollständige Text in Deutsch befindet sich auf <http://creativecommons.org/licenses/by-nc-sa/2.0/de/legalcode>.



Creative Commons License Deed

Namensnennung-NichtKommerziell-Weitergabe unter gleichen Bedingungen 2.0 Deutschland

Sie dürfen:



den Inhalt vervielfältigen, verbreiten und öffentlich aufführen



Bearbeitungen anfertigen

Zu den folgenden Bedingungen:



Namensnennung. Sie müssen den Namen des Autors/Rechtsinhabers nennen.



Keine kommerzielle Nutzung. Dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.



Weitergabe unter gleichen Bedingungen. Wenn Sie diesen Inhalt bearbeiten oder in anderer Weise umgestalten, verändern oder als Grundlage für einen anderen Inhalt verwenden, dann dürfen Sie den neu entstandenen Inhalt nur unter Verwendung identischer Lizenzbedingungen weitergeben.

- Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter die dieser Inhalt fällt, mitteilen.
- Jede dieser Bedingungen kann nach schriftlicher Einwilligung des Rechtsinhabers aufgehoben werden.
- Nothing in this license impairs or restricts the author's moral rights.

Die gesetzlichen Schranken des Urheberrechts bleiben hiervon unberührt.

Das Commons Deed ist eine Zusammenfassung des Lizenzvertrags in allgemeinverständlicher Sprache.

Mikrocontrollertechnik

MODUL

A

Inhaltsverzeichnis MODUL A

A0	Vorwort.....	1
	Darstellung.....	3
A1	Einführung.....	5
	Aufbau eines Mikroprozessorsystems.....	6
	Der Mikroprozessor (μP).....	7
	Der Speicher (Zentralspeicher).....	8
	RAM (Random Access Memory).....	8
	ROM (Read Only Memory).....	8
	Ein-/Ausgabe-Bausteine.....	8
	Adress-, Daten- und Steuerbus.....	9
	Vom Mikroprozessor zum Mikrocontroller.....	9
	CISC und RISC.....	10
	CISC (Complex Instruction Set Computing).....	10
	RISC-Prozessoren (Reduced Instruction Set Computing)	11
	Die "Von Neumann-Architektur" und die "Harvard-Architektur".....	11
	Von Neumann-Architektur (http://de.wikipedia.org/wiki/Von-Neumann-Architektur).....	11
	Harvard-Architektur (http://de.wikipedia.org/wiki/Harvard-Architektur).....	13
	Aufbau eines Mikrocontrollersystems.....	14
	Der Mikrocontroller (μC).....	15
	Der interner Speicher des ATmega32.....	17
	Der Programm- oder Befehlspeicher (ROM, Flash-EPROM).....	17
	Der Arbeits- oder Datenspeicher.....	18
	Das statische SRAM (Static Random Access Memory).....	19
	Die Arbeitsregister (r0-r31).....	19
	Die SF-Register (special function register, Sonderfunktions-Register).....	19
	Der Stapelspeicher.....	20
	Der nichtflüchtige Speicher (EEPROM).....	20
	Zusammenfassung:.....	21
A2	Assemblerprogrammierung.....	23

Die Assembler-Programmervorlage.....	25
A3 Digitale Ein- und Ausgabe.....	35
Digitale Daten ausgeben.....	35
Das erste Programm (A303_dig_out_8bit.asm).....	35
Digitale Daten einlesen.....	43
Das dritte Programm (A305_dig_in_1bit.asm).....	43
Pull-Up Pull-Down.....	45
Die Maskierung von Daten.....	47
Die AND-Verknüpfung:.....	47
Die OR-Verknüpfung:.....	50
Die XOR-Verknüpfung:.....	51
A4 Befehle und Adressierung.....	53
Die Befehle der ATmega-Mikrocontroller.....	53
Datentransferbefehle (Datentransportbefehle).....	54
Arithmetische und logische Operationen (Befehle).....	54
Bitorientierte Befehle.....	54
Sprungbefehle (jump), Verzweigungsbefehle (branch) und Unterprogrammbe- fehle (call).....	55
Sonstige Befehle.....	55
Das Zustands- oder Statusregister SREG.....	56
Adressierungsarten.....	57
Adressierung des Datenbereichs:.....	58
Die unmittelbare Adressierung (r15-r31).....	58
Die direkte Registeradressierung.....	61
Die direkte Adressierung der SF-Register (SonderFunktions-Register).....	63
Die direkte Adressierung des Datenspeichers (SRAM).....	64
Die indirekte Adressierung.....	66
Indirekte Adressierung mit automatischem Erhöhen bzw. Erniedrigen des Adresszeigers.....	69
Indirekte Adressierung mit festem (konstantem) Abstand.....	70
Indirekte Adressierung mit "push" und "pop".....	72
Wiederholung.....	72

Adressierung des Programmbereichs.....	73
Relative Adressierung des Programmspeichers mit "rjmp" und "rcall".....	73
Direkte Adressierung des Programmspeichers mit "jmp" und "call".....	74
Indirekte Adressierung des Programmspeichers mit "ijmp" und "icall".....	74
Indirekte Adressierung von Konstanten im Programmspeicher mit "lpm".....	74
A5 Zeitschleifen.....	81
8-Bit-Zeitschleife.....	81
16-Bit-Zeitschleife.....	82
Verschachtelte Zeitschleifen.....	85
Externer Quarz mit 16 MHz.....	87
A6 Unterprogramme.....	89
Globale Variablen.....	92
Lokale Variablen.....	93
Retten und Wiederherstellen mit "push" und "pop".....	93
Externe Unterprogramme Einbinden mit ".INCLUDE".....	94

Neue Befehle in den einzelnen Kapiteln:

A2: rjmp, clr, ldi, out

A3: ser, sbi, cbi, sbis, sbic, in, com, mov, andi, and, brne, ori, or, eor

A4: nop, breq, cpi, subi, add, sub, sts, lds, st, ld, inc, dec, st+, ldd, adiw, lpm

A5: sbiw, sbci

A6: rcall, ret, push, pop, tst