

Inhalt

Vorwort	13
1. Kapitel: Einführung	15
2. Kapitel: Elektronik-Grundlagen	16
3. Kapitel: Entwicklungsumgebung für AVR Mikrocontroller	24
Hardware	24
Software	28
Vorgehensweise	31
4. Kapitel: Der Start / Hello World	38
\$regfile	38
\$Crystal	38
REM ' (... ')	38
config Port	39
DDRx	39
END	40
LED an, LED aus	40
5. Kapitel: Ausgabe/Die Basis	41
Wait, Waitms, Waitus	41
Do loop	41
Set/reset	41
toggle	42
SOUND	42
2. LED	42
LED an mit Verzögerung aus	43
Zeitschalter	43
Blinker	43
Leuchtturm (Gleichtaktfeuer)	43
Metronom	44
Pieper	44
6. Kapitel: Variablen und Rechnen	45
DIM	45
Strings	45
Arrays	45
const	46
Rechnen: $x = x + 5$	46
LEN	46
MID	46
INCR, DECR	46
Binärzähler	47
7. Kapitel: Strukturierung/Programmsteuerung	49
Alias	49
Sprungmarken	49
goto	49
If..then	49
Select case	50
Unterprogramme	50
Lookup/Lookupstr	51

Lookdown	51
Restore/Read	51
Maulwurf-Schreck	52
Ampel, einfach	53
Ampel	54
Zahnputz-Timer	56
Saunatimer	58
8. Kapitel: Schleifen	61
Do loop	61
Do loop until	61
While wend	61
For next	62
Rotate/Shift/*2	62
Morsegeber	62
Baustellen-Lauflicht	64
Knight-Rider	65
Lauflicht mit Rotate	65
Lauflicht mit 16 Ausgängen (mit Overlay); Simulator	66
Musik mit SOUND	70
Musik nach Noten mit SOUND	72
9. Kapitel: Eingabe, digital	74
Definieren des Input-Ports bzw. Input-Pins	74
Abfragen des Input-Ports bzw. Input-Pins	74
Pullup-Widerstand (extern / intern)	74
Tastenentprellung	75
DEBOUNCE	75
Treppenhaus-Licht	76
LED an/aus (Entprellen)	77
Ausschalt-Timer	79
Doppelklick, Langdrücken, Kurzdrücken	80
Morsesummer	81
Fenster-offen-Alarm	82
Alarmanlage (Draht/Bewegung)	82
Pumpensteuerung mit Zeit	84
Pumpensteuerung mit Hysterese (Ab-/Zupumpen)	84
Einschalt-Menü	87
Quizmaster (mit Polling)	88
Logikbausteine/Konverter (direkt, Case, Lookup, Array, Read)	91
Stein-Schere-Papier	95
Wolf, Ziege, Kohlkopf	99
Würfel	102
Matrixtastatur	104
Schrankenöffner/Tresor	106
10. Kapitel: „Zufall“	108
Rnd	108
Würfel (rnd, Ausrollen, Muster)	108
Reaktionsspiel	110
Reaktionstester	113

Schau den Lukas	114
Gedächtnisspiel	115
Sisyphos	121
11. Kapitel: Analoge Eingabe und Ausgabe	123
Eingabe: ADC	123
Poti/LDR an ADC	123
Eingangsspannung steuert Lauflicht-Geschwindigkeit	124
Dämmerungssensor/Lichschranke	125
Pulsmesser	127
Kühlschrank-Alarm	128
Neigungsmesser-Alarm	129
Tasten mit ADC	132
Thermostat mit KTY81	135
Frostalarm/Joggertermometer	137
Binär-Voltmeter	139
Widerstandsmessung (Anzeige mit LEDs)	140
Ein-LED-Nachtlicht	141
Digital-Analog-Umwandlung	146
8 Pin Widerstandsleiter (Funktionsgenerator)	146
PWM und RC-Glied	152
Spannungsregler LM317	157
12. Kapitel: LC-Display	161
Config	161
LCD	161
CLS	161
Upperline/Lowerline	161
Cursor	161
Locate	161
Display	161
Einfache Textausgabe	162
Schaltjahr, Wochentag und Tag im Jahr	163
Selbstdefinierte Zeichen für LCD	165
Mondphasen	168
LCD mit 20 Spalten und 4 Zeilen	170
Temperatursausgabe von LM35-Sensor	174
Temperatursausgabe von LMx35-Sensor	175
Balkenanzeige für Temperatur (LM35)	176
Dioden- und Widerstandstester	178
Alarmlinie	182
Message-Box	184
13. Kapitel: EEPROM für dauerhaftes Speichern	191
Schreiben/lesen	191
Prüfen am Programmmanfang	191
Speichern eines Wertes im EEPROM	192
Speichern der Texte im EEPROM	194
Speichern in EEPROM beim Brennen mit \$eeprom/Data	196
Maximalwert in EEPROM speichern	199
Message-Box für Stromsparer	202
14. Kapitel: (Sieben)segmentanzeige	207
1 Stelle, direkte Ansteuerung	207

2 Stellen, direkte Ansteuerung.....	208
4 Stellen, Multiplex-Ansteuerung.....	209
4 Stellen, Multiplex-Ansteuerung mit Hochzählen mit Sekundentimer	211
4 Stellen mit Multiplex: Uhr.....	213
13-Segmentanzeige/Alphanumerische Anzeige	215
15. Kapitel: LED-Matrix.....	216
5x7	216
Buchstabe/Zeichen auf 5x7-Matrix	216
Selbstdefiniertes Zeichen: Herz.....	217
Anzeige einer Zeichenkette (String).....	218
Anzeigen Zeichenkette mit Pause zwischen gleichen Zeichen.....	219
Musteranzeige auf Dotmatrix-Anzeige.....	221
3x5 Anzeige.....	224
virtuell.....	227
16. Kapitel: Interrupts	228
Basis mit INT0 / INT1.....	229
Quizmaster mit PCINT.....	231
17. Kapitel: Timer/Counter/PWM	236
Timer	236
Timer für 1 Sekunde.....	240
Ausschalt-Timer mit Stufen.....	241
LED mit 1 Hz	243
100ms Signal	244
Frequenz erzeugen.....	245
440 Hz Signal	245
Smiley-Blitzer	246
Quarzuhr	252
Counter	257
Timer als Zähler	257
Frequenzzähler	258
PWM - Pulsweitenmodulation.....	260
Dimmen einer LED	263
Spannung mit RC-Glied	263
Ansteuerung Servo mit PWM-Signal	267
CTC-Modus.....	268
Töne mit CTC-Modus	270
Melodien mit CTC-Modus	271
18. Kapitel: Servo-Signale bzw. Fahrtregler-Signale.....	275
Servo-Signale erzeugen	275
Servotester mit SERVO-Befehl.....	275
Servo-Signal mit Timer	277
Ansteuerung mehrere Servos mit Timer.....	279
Servosignale auswerten	281
Auswertung eines Servosignales mit Tastfunktion.....	281
Auswertung eines Servosignals mit Memory-Funktion	282
Schalten von 4 Verbrauchern mit einem Fernsteuerkanal	283
19. Kapitel: DCF77-Funkuhr	287
Basis	289

DCF77-Uhr mit Wochentag, Monatsnamen und Empfangsanzeige	291
Manuelle Dekodierung	294
20. Kapitel: Drehimpulsgeber	297
ENCODER.....	297
Auswertung mit ENCODER-Befehl.....	299
Drehimpulsgeber mit 1 INT.....	299
Drehimpulsgeber mit 2 INT.....	300
Drehimpulsgeber mit Timer.....	301
21. Kapitel: Grafikdisplay	306
Grafikbefehle	306
Grafikdisplay Basis.....	309
Zeichensatz des Grafikdisplays.....	318
Etch A Sketch ®	320
Analoguhr	324
22. Kapitel: Farb-Grafikdisplay	330
\$Include	331
Grafik-Befehle	332
Basis.....	333
Mastermind.....	339
TicTacToe.....	348
Oszilloskop	360
23. Kapitel: IR-Signale.....	366
IR-Empfänger	366
IR-Sender.....	368
24. Kapitel: Serielle Datenübertragung	370
\$Baud	373
Print.....	374
Input	374
Waitkey	374
Inkey	374
Open.....	375
Serielle Daten senden und empfangen	375
Serielle Basis: Daten an PC senden	375
Logdatei erstellen.....	378
Senden an PC-Programm.....	382
Serielle Ausgabe und Eingabe	383
Software UART: PC schaltet LEDs via ATtiny13	384
Zeichenweise mit ATtiny13.....	385
Eingabe am PC – Ausgabe auf LCD.....	386
Seriell LC-Display.....	388
Seriell LC-Display mit ATmega8 ansteuern	390
Seriell Reichelt-Display mit ATtiny13 ansteuern.....	391
Seriell Newhaven-LC-Display mit ATtiny13 ansteuern.....	392
Seriell Fungizmos-Display mit Zufallszahl (ATtiny13).....	395
Senden von Mikrocontroller zu Mikrocontroller (ASCII)	396
Senden von Mikrocontroller zu Mikrocontroller mit PRINTBIN.....	399
GPS-Empfänger	400
GPS-Signal zeichenweise auswerten	401
GPS-Signal mit SPLIT-Befehl auswerten.....	404
USB <-> UART	407

25. Kapitel: I2C = TWI.....	408
Thermometer mit DS1621	409
Thermometer mit Alarm mit LM75.....	412
Thermometer mit Alarm mit DS1631.....	414
Kompass mit CMPS03	418
Entfernungsmesser mit SRF02	421
Uhr mit DS1307	423
EEPROM mit 24Cxx (xxk)	429
Porterweiterung PCF8574 / PCF8574A (8 I/O Pins).....	443
Porterweiterung PCA9555 (16 I/O Pins)	451
7-Segment-Ansteuerung mit SAA1064.....	459
BlinkM	467
1-Wire Thermometer DS1820.....	470
26. Kapitel: Weiteres	474
Watchdog-Timer.....	474
Watchdog Basis	474
Watchdog Basis mit Interrupt.....	475
Sleep-Modes	476
Powerdown – Wecken mit externem INT-Interrupt	476
Würfel mit Powerdown – Wecken mit externem INT-Interrupt.....	477
Powerdown – Wecken mit Watchdog-Interrupt	479
Powerdown – Wecken mit Watchdog für ADC-Messung.....	480
Powersave – Wecken mit Softclock für ADC-Messung.....	482
Fuse-Bits.....	484
Externer Quartz versus interner RC-Oszillator	484
27. Kapitel: AVR Butterfly	489
Butterfly Basis	501
Butterfly komplett	505
28. Kapitel: Arduino	531
29. Kapitel: Roboter.....	537
Asuro	537
3pi	539
NIBObee.....	541
30. Kapitel: Einsparmöglichkeiten / Fehlersuche	543
1.) Pins einsparen.....	543
2.) Code übersichtlicher	544
3.) Strom minimieren.....	544
4.) Bauteile sparen	545
5.) Programmspeicher sparen	545
Zu berücksichtigen.....	546
Anhang.....	547
BASCUM Variablen-Typen	547
Zahlen- und Stringmanipulationen	547
Syntax wichtigster Befehle	549
BASCUM	549
BASCUM-Befehle für Grafik-Display.....	551
Farb-Grafikdisplay	552
Übersicht AVR-Mikrocontroller.....	553

Pinbelegungen AVR-Mikrocontroller.....	553
LC-Display.....	560
Spannungsregler 7805.....	562
Selbsthaltung/Automatische Abschaltung.....	565
LEDs	568
5*7 Dotmatrix	571
7-Segmentanzeige / 16-Segmentanzeige.....	571
ISP-Stecker	574
Widerstandscode	575
smd-Code	575
Gehäuseformen	578
LCD-Zeichensatz (5*7).....	582
3x5 Dot Zeichensatz.....	584
Siebensegmentanzeige Zeichensatz	585
16-Segmentanzeige Zeichensatz	586
Pixel-Formulare für Grafikdisplay	587
Portrait	587
Landscape	588
Schalten höherer Ströme	589
Servo-Signal.....	612
Infrarot-Signal.....	614
DCF77-Signal	615
GPS-Signal.....	617
Morsecode.....	618
Zahlendarstellungen	620
Spannungsbezeichnungen	620
SOUND-Befehl.....	621
Temperaturmessung	623
Speicher (HWSTACK, SWSTACK, FRAME).....	639
myAVR-Erweiterungen	641
Programmablaufpläne und Struktogramme.....	659
Bezugsquellen	660
Bücher	662
Zeitschriften	665
Links	665

Vorwort

Dieses Buch richtet sich sowohl an Einsteiger der Elektronik und der Mikrocontrollertechnik als auch an diejenigen, die bereits die Grundlagen der Elektronik oder auch die Grundzüge der Mikrocontroller-Technik kennen.

Dem Einsteiger wird vermittelt, dass sich durch den Einsatz von Mikrocontrollern fast alle Aufgaben der (überwiegend digitalen) Elektronik in verschiedensten Anwendungsbereichen sehr effizient und einfach lösen lassen. So bekommt der Modelleisenbahner und der Modellbauer ein umfangreiches Wissen vermittelt ebenso wie der Roboterbauer oder der Hobbyelektroniker. Der Interessierte, Schüler, Auszubildende oder Student lernt Schritt für Schritt, komplexere Projekte umzusetzen.

Das Buch ist so aufgebaut, dass der Leser von einfachen Grundlagen mit simplen Blinkschaltungen bis zu anspruchsvollen aber dennoch stets verständlich dargestellten Anregungen mit GPS-Signalen, seriellen Übertragungen, LC- oder Farbgrafikdisplay durch sich steigernde Schwierigkeitsgrade geführt wird. Am Anfang eines jeden Kapitels werden die grundlegenden neuen Konstrukte bzw. Befehle vorgestellt, welche dann durch sehr viele Anregungen aus unterschiedlichsten Bereichen illustriert werden. Durch die vielen sorgsam ausgewählten und getesteten Beispiele wird dem Leser das Verständnis für die Materie vermittelt und die Angst genommen, in dieses spannende Thema einzusteigen.

Der fortgeschrittene Leser findet in diesem Buch eine Fundgrube von Anregungen und Informationen für die Praxis. Neben der Vielfältigkeit der Beispiele und deren Abdeckung der verschiedenen Bereiche wurde großer Wert darauf gelegt, das Buch so zu gestalten, dass es sowohl dem Einsteiger als auch dem Fortgeschrittenen als grundlegendes Nachschlagewerk dienen wird. Neben der Syntax der wichtigsten Sprachbefehle und den Pinbelegungen der wichtigsten AVR-Mikrocontroller bietet das Buch alle relevanten Informationen, die beim Bau von Projekten benötigt werden. Wie war noch die Pinbelegung beim Spannungsregler? Wie schalte ich höhere Ströme? Wie stelle ich Zeichen auf einem Display dar? Wie groß sollte der Vorwiderstand der LED sein? Diese und viele weitere immer wieder auftretende Fragen werden schnell und übersichtlich beantwortet. Dadurch ist das Buch ein wichtiger Begleiter in jeder Elektronik-Werkstatt.

Dieses Buch ist nicht 'ein weiteres' Buch über Elektronik. Oder 'ein weiteres' Buch über Mikrocontroller. Von beidem gibt es bereits einige – zum Teil sehr gute – Bücher. Darüber hinaus ist das Internet eine Fundgrube für unzählige Einführungen in die Elektronik bzw. in die Welt der Mikrocontroller und beherbergt viele Foren und liebevoll erstellte Webseiten mit sehr vielen Schaltungs- und Programmierideen. Das Buch bietet eine systematische Einführung – geeignet für Selbststudium oder als Grundlage in der Lehre – mit sehr vielen Anregungen und dient als umfangreiches Nachschlagewerk.

Es wird wenig vorausgesetzt. Am wichtigsten ist die Neugier und der Wunsch, ein faszinierendes Gebiet der Elektronik zu betreten. Ansonsten ist der Einstieg mit geringem finanziellen und zeitlichen Aufwand und mit wenig Wissen möglich. Anfangs werden die wichtigsten Grundlagen der Elektronik gelegt bzw. wiederholt (und im Anhang weiter vertieft). Dann wird die Entwicklungsumgebung beschrieben und schließlich in jedem Kapitel auf die Besonderheiten hingewiesen. Die Priorität liegt in diesem Buch ganz deutlich bei Verständlichkeit und Einfachheit. Es wird bewusst auf Assemblerprogrammierung, direkte Benutzung von Rechen- oder Status-Registern, Benutzung komplexer Bussysteme usw. verzichtet. Es wird nicht angestrebt, optimalen oder optimierten Code darzustellen sondern es werden einfache und verständliche Programme vorgestellt.

In den ersten Kapiteln werden Konstrukte benutzt, die der fortgeschrittene Leser der weiteren Kapitel eleganter zu lösen vermag. (Stichworte z.B.: Interrupts versus Polling. Schleifen versus Goto. PWM mit Timer versus selbstprogrammierte PWM.) Wem die vereinfachten Darstellungen zu vereinfacht vorkommen, der möge sich auf die Schulter klopfen, dass er es inzwischen besser kann. In den Beispielen soll nicht durch trickreiche Programmierung beeindruckt werden sondern es sollen

funktionierende und verständliche Beispiele als Anregungen und als Bausteine für komplexe Projekte dargestellt werden. Im Bereich der Hardware wird weitgehend auf diskrete elektronische Bauteile wie Transistoren, Kondensatoren, Spulen usw. verzichtet. Die Schaltungen bestehen größtenteils aus der Spannungsversorgung, einem Mikrocontroller und ein paar Tasten und Leuchtdioden. Das Faszinierende ist, dass sich somit die meisten Ideen realisieren lassen. Ggf. werden individuelle Ergänzungen z.B. zum Schalten höherer Ströme wie im Anhang beschrieben vorgenommen. Es wird kein starrer Rahmen durch ein bestimmtes Entwicklungsboard vorgegeben sondern es wird empfohlen, den jeweils geeigneten AVR-Mikrocontroller auszuwählen. Dadurch und durch die niedrigen Kosten der Mikrocontroller wird der Leser ermuntert, schnell eigene richtige Zielsysteme zu erstellen und nicht an einem geschlossenen Baukasten oder einem Mikrocontrollermodell festzuhalten.

Ziel des Buches ist, das Verständnis für den Einsatz von AVR-Mikrocontrollern zu wecken, dem Anfänger die Scheu vor dem Einstieg zu nehmen und durch sehr viele Anregungen und Informationen ein umfangreiches Nachschlagewerk zu sein. Der Nutzen stellt sich bei Hobby und im Beruf ein. Zum Einen geht es um den 'Einfachen Einstieg in die Elektronik' mit Hilfe von AVR-Mikrocontrollern und BASCOM und zum Anderen geht es um den einfachen Einstieg in die 'Elektronik mit AVR-Mikrocontrollern und BASCOM'.

AVR-Mikrocontroller wurden gewählt weil die Firma Atmel hier eine Familie von sehr modernen Mikrocontrollern anbietet, die alle benötigten Komponenten (z.B. schnelle Verarbeitung, intern erzeugbarer Takt, Analog-Digital-Wandler, Interrupts usw.) enthält. BASCOM wurde als Hochsprache gewählt weil sie leicht erlernbar ist, einen sehr effizienten Code erzeugt und in Internet-Foren eine sehr gute Unterstützung findet. Aber auch Anwender anderer Mikrocontrollerfamilien oder anderer Programmiersprachen werden in diesem Buch viele Anregungen finden.

Zu diesem Buch gibt es eine kostenfreie CD mit allen BASCOM-Quelltexten, Datenblättern, vielen Tools (Freeware bzw. Excel-Format) usw.

Leider konnte diese CD aus produktionstechnischen Gründen dem Buch nicht beigelegt werden. Gegen Überweisung von 5,- Euro für Material, Verpackung und Versand unter Angabe einer vollständigen Adresse in den Betreffzeilen wird die CD innerhalb weniger Tage zugesandt:

Kontoinhaber: Stefan Hoffmann

Kontonummer: 24 11 11 4 60

Bankleitzahl: 200 700 24

Institut: Deutsche Bank Hamburg

Lob und konstruktive Verbesserungsvorschläge für die nächste Auflage werden gerne entgegengenommen: bascom_buch@yahoo.de

1. Kapitel: Einführung

Die heutige Welt ist ohne Elektronik nicht mehr vorstellbar. Wir finden sie überall und in unterschiedlichsten Geräten: Wir benutzen elektronische Spiele, Taschenrechner, Uhren, Handys, Thermometer, GPS-Geräte und vieles mehr. Wir fahren Autos, benutzen Flugzeuge oder Fahrstühle. Genauso wie im "echten Leben" bilden wir die Realität im Modellbereich mit Ampelanlagen, Steuerungsanlagen usw. ab.

Die Elektronik hat sich in den vergangenen Jahrzehnten rasant weiterentwickelt. Durch zunehmende Integration und Verkleinerung sowie Massenproduktion wurden viele Geräte erst möglich bzw. finanziell abbildbar.

Mikrocontroller sind im Grunde kleine eigenständige Computer, die bereits Schnittstellen zur Außenwelt enthalten. Ein Mikrocontroller ersetzt sehr viele diskrete oder integrierte elektronische Bauteile. Vielfach benötigt man nur einen kleinen Mikrocontroller für Projekte, bei denen man früher sehr viele elektronische Bauteile benutzen musste. Dadurch, dass das Verhalten von Mikrocontrollern über ein Programm gesteuert wird ist der Entwickler bei der Erstellung von Projekten sehr flexibel und es können auch nachträglich problemlos Verbesserungen oder Erweiterungen (Firmware-Updates) vorgenommen werden. Das wird sogar von modernen Waschmaschinen bereits unterstützt. Da wir von Geräten mit Mikrocontrollern umgeben sind ist es sehr gut, zu verstehen, wie diese prinzipiell funktionieren und es ist auch sehr einfach möglich, eigene Geräte mit Hilfe von Mikrocontrollern zu erstellen.

Mikrocontroller sind fast überall und es lässt sich fast alles mit Mikrocontrollern lösen. Ein einfacher Blinker oder eine Ampelanlage für die Modelleisenbahn sind ebenso möglich wie die Steuerung eines Roboters oder die Steuerung industrieller Prozesse.

Es wird nur sehr wenig an Hardware benötigt; vieles ist in das Programm verlagert, wo es leichter erstell- und änderbar ist.

Es gibt verschiedene Familien von Mikrocontrollern und es gibt verschiedene Programmiersprachen. Im Rahmen dieses Buches werden Mikrocontroller der beliebten und bewährten AVR-Reihe von Atmel verwendet. Als Programmiersprache wird BASCOM verwendet. Sicherlich könnte man das meiste auch mit anderen Mikrocontrollern (z.B. PIC) oder anderen Programmiersprachen (z.B. C oder Assembler) lösen. Bei der Hardware sollte man sich einmal für eine Familie entscheiden und dann dabei bleiben. Die AVR-Mikrocontroller sind modern, leistungsfähig und sehr gut erhältlich. Bei der Wahl der Programmiersprache gibt es teilweise Glaubenskriege. Meist basiert das darauf, dass Menschen dazu neigen, das zu verteidigen, was sie am besten kennen. Ob die Ampelanlage für die Modellbahn mit C, Assembler oder BASCOM erstellt wurde macht keinerlei Unterschied. Wichtig ist hier ebenso wie bei komplexeren Projekten das Ergebnis. BASCOM, ein BASIC-Dialekt inklusive Entwicklungsumgebung, ist sehr leicht verständlich und erlernbar und es wird sehr effizienter Maschinencode erzeugt. Es wird angenommen, dass bei den meisten Lesern die Programmiersprache Mittel zum Zweck ist. Die schnelle Erlernbarkeit und gute Verständlichkeit sowie der schnelle Erfolg stehen in der Priorisierung ganz oben.