

Mikrocontroller-Praxiskurs für Arduino-Einsteiger

mit dem neuen
Elektor Arduino Nano MCCAB Trainingsboard

Einführung in die Grundlagen
der Hard- und Software

U1 = 2.345 V
U2 = 0.123 V

P1 = 8.765 kΩ
P2 = 0.456 kΩ

Write: 23792 µs
Clear: 3500 µs



```
*/  
#  
#def // die  
LED (LED_PIN); // Objek  
// Function, die von der Ob  
wird  
void switchTurnedOn() {  
  Led.toggle(); // der Zustand  
}  
KeySwitch Key(SK4, ACTIVE_HIGH,  
void setup() { } // nichts zu  
void loop() { } // nichts zu tun
```

Wolfgang Trampert

Mikrocontroller-Praxiskurs für Arduino-Einsteiger

mit dem neuen
Elektor Arduino Nano Trainingsboard
MCCAB

Einführung in die Grundlagen der Hard- und Software
mit vielen praktischen Beispielen



● © 2023: Elektor Verlag GmbH, Aachen.

Alle Rechte vorbehalten.

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch auszugsweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet.

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen können auch dann eingetragene Warenzeichen sein, wenn darauf nicht besonders hingewiesen wird. Sie gehören dem jeweiligen Warenzeicheninhaber und unterliegen gesetzlichen Bestimmungen.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für die Mitteilung eventueller Fehler sind Verlag und Autor dankbar.

● **Erklärung**

Der Autor und der Herausgeber dieses Buches haben alle Anstrengungen unternommen, um die Richtigkeit der in diesem Buch enthaltenen Informationen sicherzustellen.

Sie übernehmen keine Haftung für Verluste oder Schäden, die durch Fehler oder Auslassungen in diesem Buch verursacht werden, unabhängig davon, ob diese Fehler oder Auslassungen auf Fahrlässigkeit, Unfall oder andere Ursachen zurückzuführen sind.

ISBN 978-3-89576-524-7

Umschlaggestaltung: Elektor, Aachen
Satz und Aufmachung: Wolfgang Trampert
Druck: Ipskamp Printing, Enschede, Niederlande

● **Elektor-Verlag GmbH, Aachen — www.elektor.de**

Elektor ist Teil der Unternehmensgruppe Elektor International Media (EIM), der weltweit wichtigsten Quelle für technische Informationen und Elektronik-Produkte für Ingenieure und Elektronik-Entwickler und für Firmen, die diese Fachleute beschäftigen. Das internationale Team von Elektor entwickelt Tag für Tag hochwertige Inhalte für Entwickler und DIY-Elektroniker, die über verschiedene Medien (Magazine, Videos, digitale Medien sowie Social Media) in zahlreichen Sprachen verbreitet werden.

Vorwort

Jeder, der nicht grade als Einsiedler in einer Höhle im Wald haust, hat in der heutigen Zeit Berührung mit Mikrocontrollern. Oft nehmen wir das gar nicht bewusst wahr, aber in die meisten Geräte unseres täglichen Lebens (Mikrowelle, Kaffeeautomat, Smartphone, KFZ, Waschmaschine, Heizungs-, Garagentor- und Aufzugsteuerung ...) ist heute – mindestens – ein Mikrocontroller „eingebettet“, der ihre Steuerung übernimmt. Man spricht hier von einem »Embedded System«.

In unserem *Mikrocontroller-Praxiskurs für Einsteiger* werden wir lernen, wie man auch ohne große Erfahrung in Elektronik und Programmiersprachen eigene Projekte mit einem Mikrocontroller realisieren kann. Beim Googeln im Internet findet man viele Berichte von Leuten, denen es ohne jegliche Vorkenntnisse gelang, mit einem Arduino®-Mikrocontrollerboard z.B. eine automatische Fischfütterungsanlage für ihren Karpfenteich, eine Lichtschranke zur Personenzählung, einen Temperatur- und Feuchtigkeitsregler für ihre Räucherkerze oder ihre Hausbrauerei, einen Alkoholtester oder ein Schneehöhenmessgerät selbst zu entwickeln und zu bauen. Für den Modellbauer ist der Mikrocontroller das Universalwerkzeug zum Aufpeppen seiner Modelle. Schülern und ihren Lehrern bietet der Mikrocontroller im Hinblick auf die aktuell besonders intensive Förderung der MINT-Fächer (Mathematik, Informatik, Naturwissenschaft und Technik) in den Schulen ein großartiges Mittel zur Umsetzung ihrer Ideen.

Bei der Realisierung unserer Projekte hilft uns die *Arduino IDE*, eine Software-Entwicklungsumgebung, die man kostenlos aus dem Internet auf den eigenen PC herunterladen kann und die das gesamte Softwarepaket enthält, das man für ein eigenes Mikrocontroller-Projekt braucht:

Wir schreiben unsere Programme („Apps“) mit dem Editor der Arduino IDE auf dem PC, übersetzen sie mit dem in die IDE integrierten Compiler in die Bits und Bytes, die der Mikrocontroller versteht und laden sie über ein USB-Kabel in den Speicher des Mikrocontrollers. Zur Erstellung unserer Programme müssen wir das relativ komplizierte Innenleben eines Mikrocontrollers gar nicht kennen, denn das »Arduino-Konzept« unterstützt uns durch viele kostenlos verfügbare Bibliotheken (*Libraries*, eine Art „App Store“) bei der Steuerung der internen Bausteine des Mikrocontrollers. Diese Libraries enthalten quasi fertige „Software-Pakete“ sogenannte *Functions*, die wir wie eine »Black Box« in unsere Programme einbinden können.

Realisieren werden wir unsere Projekte auf dem *MCCAB Trainingsboard*, einer Leiterplatte, die über ein USB-Kabel mit dem PC verbunden ist und die neben einem Arduino®-Mikrocontrollermodul alle für unsere Übungen benötigten Bauteile wie Leuchtdioden, Schalter, Taster, akustische Signalgeber etc. enthält. Auch externe Sensoren, Motoren oder Baugruppen können mit unserem Mikrocontroller-Übungssystem abgefragt oder gesteuert werden.

Der Schwerpunkt des Kurses liegt auf den praktischen Übungen, denn wir werden uns die erforderlichen Kenntnisse durch „Learning by Doing“ aneignen. Ganz ohne Theorie geht es allerdings nicht, deshalb finden wir im vorderen Teil des Buches die nötigen Grundlagen der Hard- und Software eines Mikrocontrollersystems. Doch dann folgt auch schon der umfangreiche praktische Teil, der zuerst noch recht einfach gehalten ist. Mit jeder weiteren Übung kommen jedoch neue Hard- und Software-Komponenten hinzu, die jeweils ausführlich erklärt werden. So erhält der Einsteiger Schritt für Schritt einen stetig zunehmenden Kenntnisstand der vielfältigen Möglichkeiten, die ein Mikrocontroller bietet.

Wie soll man nun dieses Buch verwenden? Ein absoluter Neuling auf dem Gebiet der Elektronik und der Programmierung sollte dieses Buch von Anfang bis Ende durcharbeiten, um zuerst einen Grundstock an Wissen aufzubauen, bevor er mit dem praktischen Teil des Kurses beginnt. Falls man aber bereits über einige Vorkenntnisse verfügt und zu den Leuten gehört, die dazu neigen, ein neues Gerät gleich einzuschalten und damit zu experimentieren und nur im Bedarfsfall auf das entsprechende Kapitel im Theorieteil zurückgreifen möchte, kann man auch mit Kapitel 4 – dem Praxisteil des Buches – starten.

Eine weitere Möglichkeit besteht darin, zuerst die zahlreichen Übungen zu bearbeiten. Falls die gestellten Aufgaben problemlos gelöst werden können, kann der entsprechende Abschnitt – zumindest vorläufig – übersprungen werden.

Allerdings sollte jeder Leser in allen Fällen zuerst die Betriebsanleitung des *MCCAB Trainingsboards* genau studieren (siehe Abschnitt 2.8), bevor er es zum ersten Mal in Betrieb nimmt, um eventuelle Schäden zu vermeiden.

Dem Arduino-Konzept ist es zu verdanken, dass auch Laien eigene Elektronik-Ideen verwirklichen können. Dies ist umso bemerkenswerter, wenn man bedenkt, dass für eine wirklich intensive Behandlung des Themas »Mikrocontroller« fundierte Kenntnisse in Informatik, Programmierung und elektronischer Schaltungstechnik erforderlich sind. Nicht umsonst dauert eine Ausbildung oder ein Studium dieser Fachrichtung mehrere Jahre.

Wir werden mit diesem Arduino-Praxiskurs bereits genügend Wissen aufbauen, um erste Projekte selbst realisieren zu können, denn wir schaffen den Grundstock für ein effizientes Arbeiten mit den faszinierenden Möglichkeiten der Arduino-Welt: Wir lernen die elementaren Grundlagen der Hardware und des Programmierens in der Programmiersprache C kennen.

Wer tiefer einsteigen und mehr zu diesem faszinierende Thema wissen möchte – und das wird jeder, wenn er nach ersten Erfolgserlebnissen einmal „Blut geleckt“ hat – kann weitermachen mit der demnächst erscheinenden Fortsetzung des Arduino-Praxiskurses (Literaturhinweis [16]), in der es dann ans „Eingemachte“ geht, denn neben einem tieferen Einstieg in die Sprache C werden wir Projekte realisieren, in denen wir z.B. Sensoren abfragen, physikalische Größen messen, den Mikrocontroller auf dem *MCCAB Trainingsboard* via Bluetooth mit dem Smartphone steuern, sogar ein Mini-Speicheroszilloskop bauen und vieles mehr.

Und nun wünsche ich allen Wissbegierigen viel Erfolg beim Durcharbeiten dieses Buches und bei der Lösung der gestellten Übungsaufgaben.

Wolfgang Trampert

Inhalt

Vorwort	3
Inhalt	5
1 Einführung	8
2 Die Hardware-Grundlagen	10
2.1 Die elektronische Verarbeitung von Daten und Signalen.....	10
2.1.1 Signalzustände und Zahlenbildung in der Digitaltechnik.....	12
2.1.2 Datenverarbeitung »digital« statt »analog«.....	13
2.2 Was ist ein Mikroprozessor?	14
2.3 Die Bestandteile eines Mikroprozessor-Systems.....	15
2.4 Vom Mikroprozessor zum Mikrocontroller	16
2.5 Die Unterschiede zwischen Mikrocontroller und PC	17
2.6 Die Vorgehensweise beim Einsatz eines Mikrocontrollers.....	19
2.7 Der Datenverkehr mit der „Außenwelt“	20
2.7.1 Digitale Ein- / Ausgänge (<i>GPIOs</i>).....	20
2.7.2 Analoge Ein- / Ausgänge.....	21
2.7.3 Serielle Ein- / Ausgänge	25
2.8 Das <i>MCCAB Trainingsboard</i>	29
3 Die Software-Grundlagen	37
3.1 Zahlensysteme	37
3.1.1 Das Dual- / Binärsystem.....	37
3.1.2 Das Hexadezimalsystem	38
3.1.3 Zahlensysteme in der Programmiersprache C.....	40
3.2 Die ALU, das „Rechenzentrum“ des Mikrocontrollers	40
3.3 Zeichencodes und Zeichenketten	42
3.3.1 Der ASCII-Zeichencode	42
3.3.2 Zeichenketten (<i>Strings</i>).....	42
3.4 Datentypen, Variablen und Konstanten	43
3.4.1 Der Datentyp <i>void</i>	44
3.4.2 Skalare Datentypen	45
3.4.3 Variablen	47
3.4.4 Zusammengesetzte Datentypen (<i>Arrays</i> und <i>Structures</i>)	49
3.4.5 Konstanten	54
3.4.6 Der Operator » <i>sizeof()</i> « zur Ermittlung der Datengröße.....	56
3.5 Das Anwenderprogramm (<i>Sketch</i>)	57
3.6 Die integrierte Entwicklungsumgebung (<i>Arduino IDE</i>).....	61
3.6.1 Download und Installation der <i>Arduino IDE</i>	61
3.6.2 Starten der <i>IDE</i>	62
3.6.3 Die Bedienungsoberfläche der <i>IDE</i>	62
3.7 Operatoren und Ausdrücke.....	70
3.7.1 Die arithmetischen Operatoren <i>-</i> , <i>+</i> , <i>*</i> , <i>/</i> , <i>%</i> , <i>++</i> , <i>--</i>	70
3.7.2 Die Vergleichsoperatoren <i>==</i> , <i>!=</i> , <i>></i> , <i>>=</i> , <i><</i> , <i><=</i>	72
3.7.3 Die logischen Operatoren <i>!</i> , <i>&&</i> , <i> </i>	73
3.7.4 Die Operatoren <i>>></i> und <i><<</i> zur Verschiebung von Bits	74
3.7.5 Die Zuweisungsoperatoren <i>=</i> , <i>+=</i> , <i>-=</i> , <i>*=</i> , <i>/=</i> , <i>%=</i> , <i><<=</i> , <i>>>=</i>	74
3.7.6 Hierarchie der Operatoren in C	75
3.7.7 Übungen zu Abschnitt 3.7	75
3.8 Anweisungen und Blöcke	78
3.9 Auswahlanweisungen	78
3.9.1 Die bedingte Anweisung (<i>if ... else ...</i>).....	78
3.9.2 Die Mehrfachauswahl (<i>switch - case</i>)	80

3.9.3	Übungen zu Abschnitt 3.9	81
3.10	Schleifen	83
3.10.1	Die <i>for</i> -Schleife	84
3.10.2	Die <i>while</i> -Schleife	86
3.10.3	Die <i>do-while</i> -Schleife	87
3.10.4	Die »break«-Anweisung zum vorzeitigen Verlassen einer Schleife	88
3.10.5	Übungen zu Abschnitt 3.10	89
3.11	Functions	92
3.11.1	Deklaration einer Function	93
3.11.2	Definition einer Function	94
3.11.3	Aufruf einer Function	94
3.11.4	Interrupt-Functions	95
3.11.5	Methoden - die Functions von Objektvariablen in der Sprache C++	96
3.11.6	Übungen zu Abschnitt 3.11	97
3.12	Die wichtigsten <i>Arduino-Functions</i>	99
3.12.1	Function <i>setup()</i> zur Initialisierung	100
3.12.2	Function <i>loop()</i> , die Endlosschleife	100
3.12.3	<i>pinMode()</i> zur Konfiguration der digitalen Ein- und Ausgänge	100
3.12.4	<i>digitalRead()</i> zum Einlesen eines digitalen Eingangssignals	101
3.12.5	<i>digitalWrite()</i> zur Ausgabe eines digitalen Signals	102
3.12.6	<i>analogRead()</i> zum Einlesen eines Analogwerts	103
3.12.7	<i>analogReference()</i> zur Auswahl der Referenzspannung des ADCs	104
3.12.8	<i>analogWrite()</i> zur Ausgabe eines PWM-Signals	105
3.12.9	<i>bitClear()</i> , <i>bitSet()</i> , <i>bitWrite()</i> , <i>bitRead()</i> zur Bit-Bearbeitung	106
3.12.10	<i>delay()</i> und <i>delayMicroseconds()</i> zur Verzögerung der Programmausführung ..	107
3.12.11	<i>millis()</i> und <i>micros()</i> zur Zeitmessung	109
3.12.12	<i>random()</i> und <i>randomSeed()</i> zur Generierung von Zufallszahlen	110
3.12.13	<i>map()</i> zur Bereichsanpassung	113
3.12.14	<i>Serial</i> zur Kommunikation mit dem <i>Seriellen Monitor</i> / <i>Seriellen Plotter</i>	115
3.13	Bibliotheken (Libraries)	123
3.14	Die Library »MCCAB_Lib« für das <i>MCCAB Trainingsboard</i>	125
3.14.1	Die Klasse <i>KeySwitch</i>	126
3.14.2	Die Klasse <i>Matrix</i>	129
3.14.3	Die Klasse <i>LED</i>	131
3.14.4	Die Klasse <i>LedBlock</i>	131
3.14.5	Die Klasse <i>Sound</i>	133
3.15	Die Library »LiquidCrystal_I2C« zur Steuerung des LC-Displays	137
4	Der Praxisteil mit Projekten und Übungen	143
4.1	Projekt 1: Begrüßung der Welt mit dem Seriellen Monitor der Arduino IDE	143
4.2	Übung 14: Dauerausgabe auf den Seriellen Monitor	147
4.3	Übung 15: »Echo«-Sketch für die Eingaben des Seriellen Monitors	149
4.4	Übung 16: Formatierte Ausgabe eines Zählerstandes	152
4.5	Übung 17: Ausgabe von Zahlenwerten in verschiedenen Zahlenformaten	155
4.6	Übung 18: Überprüfung der Größe der C-Datentypen	158
4.7	Projekt 2: Ein- und Ausgabe digitaler Werte mit den Functions der Arduino IDE	162
4.8	Übung 19: Blinklicht mit <i>Arduino-Functions</i>	165
4.9	Übung 20: Wechselblinker mit <i>Arduino-Functions</i>	168
4.10	Übung 21: Geschaltetes Blinklicht mit <i>Arduino-Functions</i>	171
4.11	Übung 22: Blinklichter mit unterschiedlichen Frequenzen (<i>Arduino-Functions</i>)	174
4.12	Projekt 3: Blinkende LED mit der Klasse <i>LED</i> der Library »MCCAB_Lib«	178
4.13	Übung 23: Drei mit unterschiedlichen Frequenzen blinkende LEDs	180
4.14	Übung 24: Verkehrsampel	182
4.15	Projekt 4: Einlesen eines (prellenden) Schalters mit der Function <i>digitalRead()</i>	186

Inhalt

4.16	Projekt 5: Einlesen eines (prellenden) Schalters mit der Klasse <i>KeySwitch</i>	192
4.17	Übung 25: Rastende Ein- / Aus-Funktion mit Tastschalter (Stromstoßschalter)	196
4.18	Übung 26: Geschaltetes Blinklicht mit den Klassen <i>KeySwitch</i> und <i>LED</i>	199
4.19	Übung 27: Blinklichter mit unterschiedlichen Frequenzen (<i>KeySwitch</i> und <i>LED</i>)	202
4.20	Projekt 6: Ein- und Ausgabe analoger Werte	206
4.21	Übung 28: Leuchtbalken mit Arduino-Functions	210
4.22	Übung 29: Vergleich der Einlesedauer analoger und digitaler Werte	212
4.23	Projekt 7: Lauflicht mit der Klasse <i>LedBlock</i> der Library <i>MCCAB_Lib</i>	215
4.24	Übung 30: Einzelpunkt-Lauflicht mit der Klasse <i>LedBlock</i>	219
4.25	Übung 31: Ausgabe von Binärzahlen mit der Klasse <i>LedBlock</i>	222
4.26	Übung 32: Leuchtbalken mit der Klasse <i>LedBlock</i>	226
4.27	Projekt 8: Steuerung einer 3 x 3 - LED-Matrix mit Arduino-Functions	228
4.28	Projekt 9: Steuerung einer 3 x 3 - LED-Matrix mit der Klasse <i>Matrix</i>	234
4.29	Übung 33: Rotierender Leuchtbalken auf der 3 x 3 - LED-Matrix	236
4.30	Übung 34: Zufallsmuster auf der 3 x 3 - LED-Matrix.....	241
4.31	Übung 35: Reaktionstest 1.....	245
4.32	Übung 36: Elektronischer Würfel.....	251
4.33	Übung 37: Gleichverteilung von Zufallszahlen.....	254
4.34	Projekt 10: Polizei im Einsatz: Martin-Horn und Blaulicht mit Arduino-Functions ...	260
4.35	Projekt 11: Martin-Horn und Blaulicht mit der Klasse <i>Sound</i> der <i>MCCAB_Lib</i>	266
4.36	Übung 38: Ausgabe von Tonimpulsen.....	271
4.37	Übung 39: Nano-Klavier	274
4.38	Übung 40: Big Ben.....	281
4.39	Übung 41: Heul doch!	285
4.40	Projekt 12: Verwendung und Darstellungsmöglichkeiten des LC-Displays.....	290
4.41	Übung 42: Der Zeichensatz des LCDs	299
4.42	Übung 43: Wir definieren eigene Zeichen für das LCD	303
4.43	Übung 44: Die Ausführungszeiten der LCD-Methoden	307
4.44	Übung 45: Reaktionstest 2.....	311
4.45	Übung 46: LCD-Voltmeter.....	316
5	Problembehandlung – Fehlersuche mit dem Seriellen Monitor.....	323
6	Literaturverzeichnis.....	324
7	Stichwortverzeichnis.....	325
8	Anhang	328
8.1	ASCII-Tabelle.....	328
8.2	Download-Quellen.....	330