

---

# Vorwort

Herzlich willkommen bei Arduino und herzlich willkommen bei Computerexperimenten zum Anfassen. Arduino<sup>1</sup> ist ein Open Source-Projekt, das sowohl aus Hardware als auch aus Software besteht. Ursprünglich sollte Arduino Künstlern und Designern den Zugang zu Microcontrollern für interaktive Kurse erleichtern. Heute verwenden Fachleute und Laien Arduino, um Projekte im Physical-Computing zu realisieren – und Sie können das auch.

Mit Arduino sammeln Sie mit Computern grundlegende Erfahrungen, die Sie seit den 1980er-Jahren nicht mehr machen konnten, denn seitdem gab es keine Selbstbau-Computer mehr. Und mit Arduino wird die Entwicklung von selbstentwickelten Elektronikprojekten leicht gemacht, egal, ob Prototyp oder ausgefeiltes Spielzeug. Die Zeiten sind vorbei, in denen Sie Massen an Theorie und obskure Programmiersprachen lernen mussten, nur um eine LED zum Blinken zu bringen. Sie können Ihr erstes Arduino-Projekt in wenigen Minuten umsetzen, ohne Vorlesungen in Elektrotechnik belegen zu müssen.

Sie müssen also nichts über Elektronikprojekte wissen, um dieses Buch lesen zu können, und Sie legen von Anfang an richtig los. Auf den ersten Seiten lernen Sie nicht nur die wichtigsten Elektronikbauteile kennen, Sie erfahren auch, wie Sie die Programme schreiben, die Ihre Projekte zum Leben erwecken.

Dieses Buch ist von Anfang an praxisorientiert und vermeidet graue Theorie. Ich erkläre die Grundlagen für alle Projekte, und in jedem Kapitel finden Sie einen Abschnitt zur Fehlersuche – für den Fall, dass etwas schiefgeht. Dieses Buch bietet Ihnen einen Schnelleinstieg, mit dem Sie sofort loslegen und Ihre eigenen Projekte verwirklichen können.

## Für wen dieses Buch gedacht ist

Wenn Sie an Elektronik interessiert sind und besonders daran, Ihre eigenen Spielzeuge, Modelle und Ideen umzusetzen, haben Sie das richtige Buch gekauft. Auch wenn Arduino ein hübsches Werkzeug für Designer und Künstler ist, können nur

---

1 <http://arduino.cc>

Softwareentwickler alle seine Möglichkeiten voll ausschöpfen. Wenn Sie also schon Software entwickelt haben – vorzugsweise in C/C++ oder Java –, bringen Sie die besten Voraussetzungen mit.

Aber es gibt noch etwas: Sie müssen die Projekte in diesem Buch nachbauen, ausprobieren und nach Ihrem Gusto verändern. Viel Spaß dabei! Haben Sie keine Angst, Fehler zu machen. Die Abschnitte zur Fehlersuche und die praktischen Erfahrungen, die Sie von Projekt zu Projekt sammeln, werden Ihnen bei allen Schwierigkeiten helfen. Über Elektronik zu lesen, ohne dabei etwas auszuprobieren, das ist nur die halbe Miete. Sie kennen das alte Sprichwort »Wir erinnern uns an fünf Prozent dessen, was wir hören, an zehn Prozent Geschriebenes und zu 95 Prozent an Dinge, die wir selbst erlebt haben.« Und machen Sie sich keine Sorgen: Sie benötigen keine Vorkenntnisse in Elektronikprojekten!

Wenn Sie noch niemals programmiert haben, fangen Sie mit einem Programmierkurs an oder lesen Sie zuerst ein Einsteigerbuch über Programmierung (Learn to Program [Pin09] ist ein netter Einstieg). Dann lesen Sie »Die C-Programmiersprache« [KR90] oder Sie lernen C++ mit »Die C++-Programmiersprache« [Str10].

## Was Sie in diesem Buch finden

Dieses Buch besteht aus drei Teilen (»Erste Schritte mit Arduino«, »Elf Arduino-Projekte« und die Anhänge). In Teil 1 erlernen Sie die Grundlagen, die Sie kennen müssen, um die Projekte in Teil 2 zu bauen, lesen Sie die Kapitel also in ihrer Reihenfolge und gehen Sie die Übungen durch. Die Kapitel im zweiten Teil bauen ebenfalls aufeinander auf und verwenden jeweils Verfahren und Programmteile aus früheren Kapiteln.

Es gibt jedoch eine Ausnahme: In diesem Buch erstellen Sie mehrere Google Chrome-Apps, die Ihren Webbrowser mit dem Arduino verbinden. Anhang D, *Steuerung des Arduino über einen Webbrowser*, erklärt im Einzelnen, wie Chrome Apps funktionieren, weshalb Sie es nach Kapitel 4, *Eine Bibliothek für einen Morse-Code-Generator*, lesen sollten.

Hier ist eine Kurzübersicht:

- Das Buch beginnt mit den Grundlagen der Arduino-Entwicklung. Sie lernen, wie Sie die IDE verwenden und wie Sie Programme kompilieren und hochladen. Schnell beginnen Sie mit dem ersten Projekt – dem Elektrowürfel –, das Ihnen den Umgang mit einfachen Bauteilen wie Leuchtdioden, Tastern und Widerständen zeigt. Mit dem Morse-Code-Generator sehen Sie, wie einfach es ist, Ihre eigenen Arduino-Bibliotheken anzulegen.
- Dann lernen Sie den Umgang mit analogen und digitalen Sensoren. Sie verwenden einen Temperatursensor und einen Ultraschallsensor, um einen sehr genauen digitalen Zollstock und eine Einparkhilfe zu bauen. Dann verwenden Sie einen dreiachsigen Beschleunigungsmesser, um Ihren eigenen Spielcontroller zu bauen und eine Abwandlung des Spiels *Breakout* zu programmieren.

- Bis zu diesem Zeitpunkt haben Sie Daten meistens über LEDs oder den seriellen Anschluss des Arduino ausgegeben. Nun werden Sie den Arduino an einen handelsüblichen Fernseher anschließen und eigene Videosignale daran ausgeben. Als Beispiel erstellen Sie eine grafische Thermometeranwendung, die Sie auf dem Fernseher in Ihrem Wohnzimmer wiedergeben können.
- Bei der Elektronik müssen Sie Ihre Spielzeuge nicht zwingend selbst bauen. Sie können auch mit vorhandener Hardware herumspielen und Sie werden sehen, wie leicht es ist, z. B. den Nunchuk für Nintendo Wii in eigenen Programmen einzusetzen. Damit haben Sie schon bald alles Material und alle Fähigkeiten, um Ihre eigene Videospielekonsole zu bauen.
- Der Arduino ist kein Einzelgänger, sondern kann hervorragend mit verschiedenen Netzwerktechnologien zusammenarbeiten. Sie werden lernen, wie Sie den Arduino auf verschiedene Weise mit dem Internet verbinden und wie Sie damit Twitter-Meldungen und E-Mails verschicken. Außerdem konstruieren Sie eine Alarmanlage, die Ihnen eine E-Mail schickt, sobald sich jemand in Ihrer Abwesenheit in Ihrem Wohnzimmer bewegt.
- Anwendungen oder Geräte mit dem Nunchuk zu steuern, ist nett, eine Fernbedienung ist aber eine viel spannendere Sache. Daher bauen Sie als Nächstes eine Universal-Fernbedienung, die Sie sogar von Ihrem Webbrowser aus steuern können.
- Schließlich arbeiten Sie mit Elektromotoren, um ein Spaßgerät für Ihr nächstes Programmierprojekt zu bauen. Sie können es mit Ihrem Versionskontrollsystem verbinden, und immer dann, wenn eine Kompilierung fehlschlägt, bewegt es einen Pfeil auf den Namen des verantwortlichen Programmierers.
- In den Anhängen lernen Sie die Grundlagen zur Elektronik und zu Lötarbeiten. Sie finden außerdem weiterführende Informationen über die Programmierung einer seriellen Schnittstelle und zur Arduino-Programmierung allgemein. In einem weiteren Anhang wird erklärt, wie Sie den Arduino über einen Webbrowser steuern können.

Jedes Kapitel beginnt mit einer ausführlichen Liste der Bauteile und Werkzeuge, die Sie für das vorgestellte Projekt benötigen. Jedes Kapitel enthält außerdem viele Fotos und Diagramme, die zeigen, wie alles zusammengebaut wird. Ferner finden Sie im Buch immer wieder Beschreibungen interessanter Arduino-Projekte in den Randnotizen (graue Kästen).

Manchmal läuft jedoch einfach etwas schief, und die Fehlersuche in Schaltkreisen kann schwer und komplex werden. Daher finden Sie in jedem Kapitel den Abschnitt *Wenn es nicht funktioniert*, in dem typische Probleme und deren Lösungen beschrieben werden.

Bevor Sie die Lösungen in diesen Abschnitten lesen, sollten Sie jedoch versuchen, sie zunächst einmal selbst zu finden. Das ist der beste Weg, um zu lernen.

Für den unwahrscheinlichen Fall, dass Sie überhaupt keine Probleme bekommen, finden Sie am Ende jedes Kapitels eine Liste mit Übungen, die Ihre Fähigkeiten weiter verbessern.

## Arduino Uno und die Arduino-Plattform

Nach dem Erscheinen verschiedener Arduino-Boards und IDE-Versionen entschied sich das Arduino-Team, die Version 1.0 der Plattform zu spezifizieren. Die Versionsnummerierung war zunächst allerdings nicht unmittelbar einsichtig. Zu Beginn des Projekts hatten die Entwickler die Versionsnummer bei jedem neuen Release um 1 erhöht. Als sie bei Nr. 23 angekommen waren, entschieden sie jedoch, die Versionsnummer 1.0 für das nächste Release zu verwenden. Daher ist Arduino 1.0 neuer als Arduino 23.

Arduino 1.0 wurde Ende 2011 freigegeben und bildet die Referenz für alle Entwicklungen. Außerdem ist Arduino Uno erschienen, und es gab schrittweise Verbesserungen an der IDE und den dazugehörigen Bibliotheken.

Daneben hat das Arduino-Team weitere Arduino-Platinen entwickelt, z. B. den Arduino Due<sup>2</sup>, den Arduino Leonardo<sup>3</sup> und den Arduino Yún<sup>4</sup>, die entweder leistungsstärkere Mikrocontroller oder zusätzliche Hardware wie etwa ein WLAN-Modul aufweisen.

Die meisten der neuen Platinen wiesen eine neue ARM-Prozessorarchitektur auf, die nicht mit der Architektur der AVR-Prozessoren älterer Modelle kompatibel ist. Um dieses Problem zu lösen, hat das Arduino-Team parallel zur Version 1.0.x der Arduino IDE auch die Version 1.5.x entwickelt, die alle verschiedenen Prozessorarchitekturen unterstützen wird, ohne dass der Benutzer eingreifen muss. Zurzeit brauchen Sie Version 1.5 nur, wenn Sie den Arduino Due oder den Arduino Yún programmieren möchten. Für alle anderen Platinen verwenden Sie Version 1.0.x.

Daneben gibt es noch eine eigene IDE für den Arduino Galileo<sup>5</sup>. Diese Platine wurde von Intel entwickelt und ist mit dem Arduino kompatibel.

Die Informationen in diesem Buch sind auf dem neuesten Stand für die Version 1.0.6 der Arduino-Plattform und für den Arduino Uno. So gut wie alle Projekte funktionieren aber auch auf jüngeren Modellen wie dem Leonardo und dem Due sowie auf älteren wie dem Duemilanove und dem Diecimila. Der gesamte Code in diesem Buch wurde mit Arduino 1.0.6 getestet.

---

2 <http://arduino.cc/en/Main/ArduinoBoardDue>

3 <http://arduino.cc/en/Main/ArduinoBoardLeonardo>

4 <http://arduino.cc/en/Main/ArduinoBoardYun>

5 <http://arduino.cc/en/ArduinoCertified/IntelGalileo>

## Programmcode und Konventionen

Auch wenn dieses Buch von Open Source-Hardware und -Elektronik handelt, finden Sie darin viele Programmbeispiele. Sie sind nötig, um die Hardware zum Leben zu erwecken und mit ihr die Projekte umzusetzen.

Für alle Programme, die auf Arduino laufen, verwenden wir C/C++. Für alle Programme auf dem PC verwenden wir hauptsächlich JavaScript und Processing<sup>6</sup>, im Anhang C.2, *Serielle Verbindungen mit unterschiedlichen Programmiersprachen*, lernen Sie jedoch auch, andere Programmiersprachen zu verwenden, um mit Arduino zu kommunizieren.

## Online-Material

Für dieses Buch gibt es eine Website unter <http://www.dpunkt.de/arduino>, von der Sie alle Programme für die Projekte in diesem Buch herunterladen können. Sie können gern (in englischer Sprache) am Diskussionsforum teilnehmen und andere Leser oder mich online treffen. Wenn Sie in diesem Buch Fehler, Tippfehler oder andere Ärgernisse finden, lassen Sie es bitte auch mich und die anderen Leser über die Errata-Seite wissen.<sup>7</sup>

Auf der Webseite finden Sie außerdem einen Link auf Fotos bei Flickr<sup>8</sup>. Hier finden Sie alle Fotos dieses Buchs in hoher Auflösung. Es gibt dort auch Fotos von Projekten anderer Leser, und wir würden uns freuen, auch Ihre Fotos dort zu finden.

Fangen wir also an!

---

6 <http://processing.org>

7 <http://www.dpunkt.de/arduino>

8 <http://flickr.com>