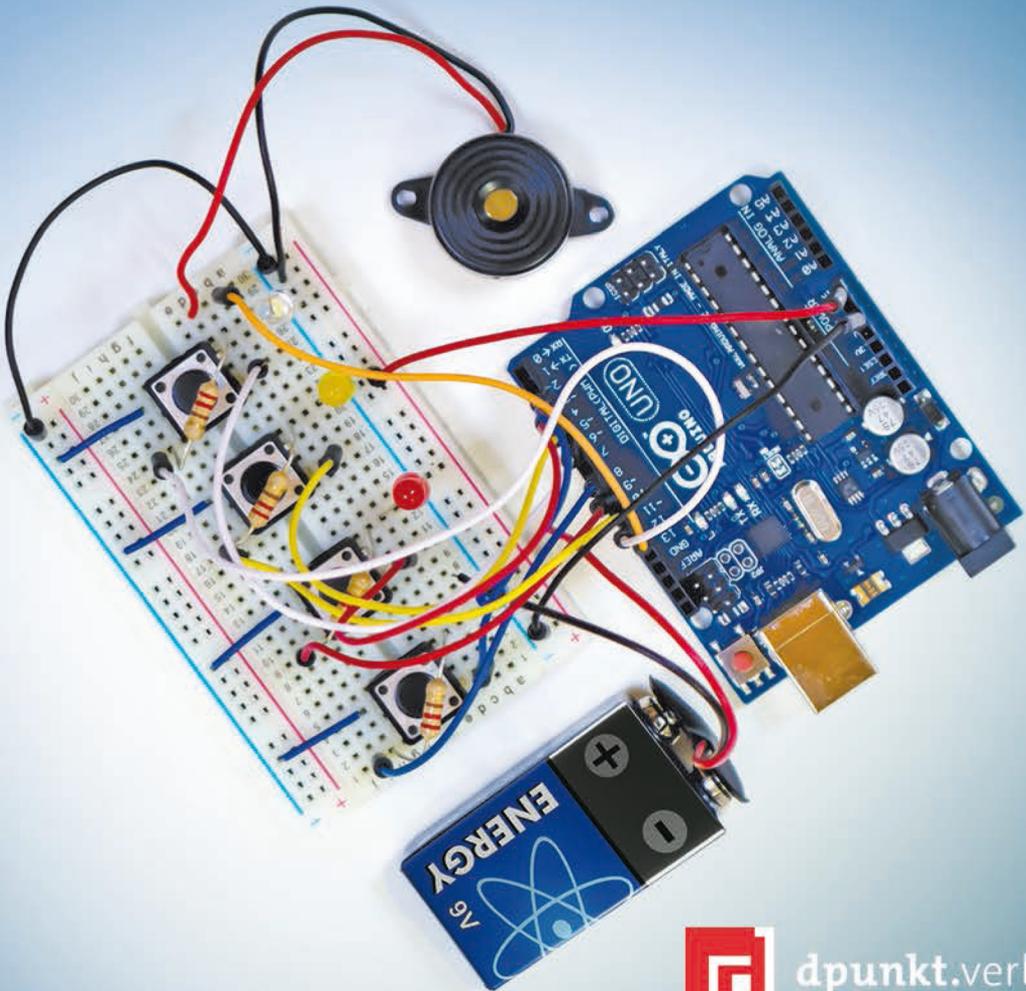


Mark Geddes

# Arduino-Projekte

25 Bastelprojekte für Maker zum Loslegen



dpunkt.verlag

Papier  
plus<sup>+</sup>  
PDF.

Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei dpunkt.plus<sup>+</sup>:

[www.dpunkt.de/plus](http://www.dpunkt.de/plus)

Mark Geddes

# Arduino-Projekte

25 Bastelprojekte für Maker zum Loslegen



dpunkt.verlag

Mark Geddes

Lektorat: Dr. Michael Barabas

Copy-Editing: Ursula Zimpfer

Übersetzung & Satz: G&U Language & Publishing Services GmbH, [www.gundu.com](http://www.gundu.com)

Herstellung: Susanne Bröckelmann

Umschlaggestaltung: Helmut Kraus, [www.exclam.de](http://www.exclam.de)

nach der Originalvorlage von No Starch Press

Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-86490-415-8

PDF 978-3-96088-089-9

ePub 978-3-96088-090-5

mobi 978-3-96088-091-2

1. Auflage 2017

Copyright © 2017 dpunkt.verlag GmbH

Wiebinger Weg 17

69123 Heidelberg

Copyright © 2016 by Mark Geddes. Title of English-language original: *Arduino Project Handbook*, ISBN 978-1-59327-690-4, published by No Starch Press.

German-language edition copyright © 2017 by dpunkt.verlag. All rights reserved.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

**Cameron und Jemma, ihr seid die  
Schöpfer und Macher der Zukunft.**

**Euch ist dieses Buch gewidmet!**

# Inhaltsübersicht

Danksagungen .....	xv
Einführung .....	xvi
Projekt 0: Erste Schritte .....	2

## Teil 1: LEDs

Projekt 1: Eine tastergesteuerte LED .....	22
Projekt 2: Lichtdimmer .....	28
Projekt 3: LED-Balkenanzeige .....	34
Projekt 4: Stroboskoplicht wie in der Disco .....	39
Projekt 5: Ein Pflanzen-Monitor .....	45
Projekt 6: Ein Geisterdetektor .....	53

## Teil 2: Sound

Projekt 7: Arduino-Musik .....	64
Projekt 8: Ein Gedächtnistrainer .....	69
Projekt 9: Geheime Klopfzeichen .....	78

## Teil 3: Servomotoren

Projekt 10: Ein Joystick-gesteuerter Laser .....	86
Projekt 11: Ein ferngesteuerter Servo .....	93

## Teil 4: LCDs

Projekt 12: Eine LCD-Anzeigetafel .....	102
Projekt 13: Eine Wetterstation .....	110
Projekt 14: Ein Wahrsager .....	117
Projekt 15: Ein Reaktionszeit-Spiel .....	124

## Teil 5: Numerische Zähler

Projekt 16: Ein elektronischer Würfel .....	134
Projekt 17: Raketenstart .....	143

## Teil 6: Sicherheit

Projekt 18: Einbruchalarm .....	156
Projekt 19: Ein Laser-Stolperdraht .....	163
Projekt 20: Eine Roboter-Kanone .....	170
Projekt 21: Ein Bewegungsmelder mit Alarm .....	178
Projekt 22: Ein Codeschloss .....	185
Projekt 23: Drahtlose Zutrittskontrolle mit einer ID-Karte .....	194

## Teil 7: Fortgeschrittene Projekte

Projekt 24: Lightshow in Regenbogenfarben .....	208
Projekt 25: Bauen Sie Ihren eigenen Arduino! .....	219

## Anhang

Anhang A: Bauteile .....	229
Anhang B: Pinbelegung des Arduino .....	243

# Inhaltsverzeichnis

Danksagungen .....	xv
Einführung .....	xvi
Die Revolution des Arduino .....	xvii
Über dieses Buch .....	xvii
Der Aufbau dieses Buchs .....	xix
Projekt 0: Erste Schritte .....	2
Hardware .....	3
Der Arduino Uno .....	3
Stromversorgung .....	3
Steckbretter .....	4
Jumperkabel .....	5
Programmierung des Arduino .....	6
Die IDE-Schnittstelle .....	6
Arduino-Sketches .....	7
Bibliotheken .....	7
Ihren Arduino testen: mit einer blinkenden LED .....	8
Funktionsweise des Sketches .....	10
Bauteileliste .....	11
Ihren Arbeitsplatz einrichten .....	13
Ausstattung und Werkzeuge .....	15
Kurzanleitung zum Löten .....	18
Sicherheit kommt zuerst .....	20

## Teil 1: LEDs

Projekt 1: Eine tastergesteuerte LED .....	22
So funktioniert es .....	24
Der Aufbau .....	25
Der Sketch .....	27
Projekt 2: Lichtdimmer .....	28
So funktioniert es .....	30
Der Aufbau .....	31
Der Sketch .....	33

<b>Projekt 3: LED-Balkenanzeige .....</b>	<b>34</b>
So funktioniert es .....	36
Der Aufbau .....	37
Der Sketch .....	38
<b>Projekt 4: Stroboskoplicht wie in der Disco .....</b>	<b>39</b>
So funktioniert es .....	41
Der Aufbau .....	41
Der Sketch .....	43
<b>Projekt 5: Ein Pflanzen-Monitor .....</b>	<b>45</b>
So funktioniert es .....	47
Der Aufbau .....	48
Der Sketch .....	51
<b>Projekt 6: Ein Geisterdetektor .....</b>	<b>53</b>
So funktioniert es .....	55
Der Aufbau .....	55
Der Sketch .....	59

## Teil 2: Sound

<b>Projekt 7: Arduino-Musik .....</b>	<b>64</b>
So funktioniert es .....	66
Der Aufbau .....	67
Der Sketch .....	67
<b>Projekt 8: Ein Gedächtnistrainer .....</b>	<b>69</b>
So funktioniert es .....	71
Der Aufbau .....	71
Der Sketch .....	73
<b>Projekt 9: Geheime Klopfzeichen .....</b>	<b>78</b>
So funktioniert es .....	80
Der Aufbau .....	81
Der Sketch .....	82

## Teil 3: Servomotoren

Projekt 10: Ein Joystick-gesteuerter Laser .....	86
So funktioniert es .....	88
Der Aufbau .....	89
Den Laser montieren .....	90
Der Sketch .....	92
Projekt 11: Ein ferngesteuerter Servo .....	93
So funktioniert es .....	95
Die Installation .....	96
Der Aufbau .....	97
Der Sketch .....	98

## Teil 4: LCDs

Projekt 12: Eine LCD-Anzeigetafel .....	102
So funktioniert es .....	104
Die LCD-Anzeige einrichten .....	104
Der Aufbau .....	105
Der Sketch .....	107
Projekt 13: Eine Wetterstation .....	110
So funktioniert es .....	112
Der Aufbau .....	112
Der Sketch .....	116
Projekt 14: Ein Wahrsager .....	117
So funktioniert es .....	119
Der Aufbau .....	119
Der Sketch .....	122
Projekt 15: Ein Reaktionszeit-Spiel .....	124
So funktioniert es .....	126
Der Aufbau .....	127
Der Sketch .....	130

## Teil 5: Numerische Zähler

Projekt 16: Ein elektronischer Würfel .....	134
So funktioniert es .....	136
Der Aufbau .....	137
Der Sketch .....	140
Projekt 17: Raketenstart .....	143
So funktioniert es .....	145
Der Aufbau .....	145
Eine richtige Zündung .....	149
Der Sketch .....	151

## Teil 6: Sicherheit

Projekt 18: Einbruchalarm .....	156
So funktioniert es .....	158
Der Aufbau .....	158
Der Sketch .....	161
Projekt 19: Ein Laser-Stolperdraht .....	163
So funktioniert es .....	165
Der Aufbau .....	165
Der Sketch .....	168
Projekt 20: Eine Roboter-Kanone .....	170
So funktioniert es .....	172
Der Aufbau .....	173
Der Sketch .....	176
Projekt 21: Ein Bewegungsmelder mit Alarm .....	178
So funktioniert es .....	180
Der Aufbau .....	181
Der Sketch .....	183

Projekt 22: Ein Codeschloss .....	185
So funktioniert es .....	187
Das Tastenfeld testen .....	187
Der Aufbau .....	189
Der Sketch .....	191
 Projekt 23: Drahtlose Zutrittskontrolle mit einer ID-Karte .....	 194
So funktioniert es .....	196
Der Aufbau .....	198
Der Sketch .....	203

## Teil 7: Fortgeschrittene Projekte

Projekt 24: Lightshow in Regenbogenfarben .....	208
So funktioniert es .....	210
Der Aufbau .....	212
Der Sketch .....	215
 Projekt 25: Bauen Sie Ihren eigenen Arduino! .....	 219
So funktioniert es .....	221
Den Chip vorbereiten .....	222
Die Arduino-Schaltung bauen .....	223

## Anhang

Anhang A: Bauteile .....	229
Die Bauteile .....	230
Arduino Uno R3 .....	230
9-V-Batteriepaket .....	230
Steckbrett .....	230
LED .....	231
Widerstand .....	231
Taster .....	231
Potenziometer .....	232
Bodenfeuchtesensor HL-69 .....	232
Piezo-Summer .....	232
Servomotor .....	233

Joystick .....	233
Infrarotempfänger .....	233
LCD-Anzeige .....	234
Feuchtigkeitssensor DHT11 .....	234
Kugelschalter .....	234
RGB-LED .....	235
7-Segment-LED-Anzeige .....	235
7-Segment-Anzeige mit vier Ziffern .....	235
Ultraschallsensor .....	236
Fotowiderstand .....	236
Raketenwerfer RC V959 .....	236
Passiver Infrarotsensor (PIR) .....	237
Tastenfeld .....	237
RFID-Leser .....	237
RGB-Matrix .....	238
Schieberegister .....	238
ATmega328p .....	238
16-MHz-Quarz .....	239
5-V-Spannungsregler .....	239
Kondensator .....	239
Scheibenkondensator .....	240
Batterieclip .....	240
Fachhändler .....	240
Weitere Anbieter in Europa .....	240
Die Farbcodes von Widerständen .....	241
<b>Anhang B: Pinbelegung des Arduino .....</b>	<b>243</b>



# Danksagungen

Ein herzliches Dankeschön geht an das fantastische Team von No Starch Press, besonders an Elizabeth Chadwick und Serena Yang, für ihre Unterstützung und Hilfe beim Schreiben dieses Buchs. Dank auch an Christopher Stanton für die technische Durchsicht und seine Vorschläge.

Es gäbe dieses Buch nicht ohne die Schöpfer des Arduino: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino und David Mellis. Vielen Dank, dass ihr mich in die Wunderwelt des Arduino eingeführt habt. Vielen Dank auch an Ken Shirriff, Warwick Smith, Steven de Lannoy und Abdullah Alhazmy für ihre freundliche Genehmigung, ihre Projekte verwenden zu dürfen.

Ich danke meiner wundervollen Frau, Emily, für ihre Unterstützung und Geduld im letzten Jahr und dafür, dass sie mir ein Zimmer unseres Hauses als Bastelhöhle überlassen hat, sodass ich all diese Projekte bauen konnte. Und natürlich dafür, dass sie der Versuchung widerstanden hat, diese Höhle täglich aufzuräumen!

Ich danke meinen Eltern, Frank und Lorna, dass sie mir als Kind die Freiheit ließen, Dinge auseinanderzunehmen, und dass sie sich nicht darüber beschwerten, wenn überall an mir Kabel hingen. Ohne ihre Förderung hätte ich meine Liebe zur Elektronik und zu Bastelprojekten nicht entdeckt. Vielen Dank auch an David und Linda für ihre fantastische Unterstützung, ihre Ermutigung und ihr Vertrauen.

# Einführung

Der Arduino ist ein kleiner, kostengünstiger Computer, der mit seiner Programmierung unendlich viele Bastelprojekte steuern kann. Wie Sie gleich sehen werden, kann der Arduino als Basis für zahlreiche Projekte dienen, zum Beispiel für einen Geisterdetektor, einen per Joystick gesteuerten Laser, einen elektronischen Würfel, einen Laser-Stolperdraht, einen Bewegungsalarm, ein Codeschloss für Ihre Eingangstür und viele andere. All diese Projekte sind einfach zu bauen und haben eines gemeinsam: Sie nutzen die Möglichkeiten des Arduino.

In den frühen 1980er-Jahren fiel mir in einer Buchhandlung ein tolles Buch mit einem Titel so ähnlich wie »Technische Spielereien« in die Hände. Die Projekte waren recht einfach, so wie beispielsweise ein Leuchtturm, der aus Taschenlampenbirnchen und einer alten Uhr bestand, die für die Drehung sorgte. Die Anregungen in diesem Buch beflügelten meine Fantasie und seit dieser Zeit habe ich immer irgendetwas gebastelt.

Ich war so neugierig, dass ich verschiedene elektrische Geräte auseinandernahm, um herauszufinden, wie sie funktionieren. Normalerweise schaffte ich es nicht, sie wieder zusammenzubauen, aber auf diese Weise erhielt ich eine ganze Reihe elektronischer Komponenten zum Basteln. (Das ist übrigens eine tolle Möglichkeit, an eine Menge Bauteile zu kommen.)

Ich kann mich gut daran erinnern, wie ich viele Taschenlampenbirnchen zusammenschaltete, um ein Flutlicht für mein Tipp-Kick-Spiel zu bauen, und einen Lautsprecher installierte, um in der Halbzeitpause Musik abzuspielen. Ich schaffte es sogar, aus einem Star-Wars-Spielzeug einige LEDs auszubauen, nur um sie dann zu zerstören, weil ich damals nicht wusste, was ein Widerstand war. Mit kleinen Motoren, Summern und Solarzellen baute ich Alarmanlagen und raffinierte Autos und natürlich ruinierte ich dabei auch einige Motoren!

Etwa zur selben Zeit (1983) brachte die Firma Sinclair in Großbritannien den Mikrocomputer ZX Spektrum 48k in den Handel und führte damit Heimcomputer im Massenmarkt ein. (Die USA hatten den Commodore 64.) Obwohl er als seriöser Computer gedacht war, wurde der ZX Spektrum auch viel zum Spielen eingesetzt, da er mit der einfachen Programmiersprache Basic ausgestattet war. So kam es, dass überall im Lande Softwarefirmen in Schlafzimmern entstanden und die Menschen Spiele für den ZX schrieben.

Dies weckte mein Interesse am Programmieren und gleichzeitig konnte ich meine beiden Hobbys miteinander verbinden. Das »Physical Computing«, bei dem Software und Hardware mit der echten Welt

interagieren, gab es schon in den achtziger Jahren, war aber eher im Bereich der Hochleistungsrechner und Robotik angesiedelt und für Privatanwender undenkbar. Jetzt, mit der Einführung des Arduino, etwa 30 Jahre später, bastele ich wieder Elektronikprojekte, aber heute kann ich diese Projekte mittels Programmierung auch zum Leben erwecken.

## Die Revolution des Arduino

Einfach ausgedrückt ist der Arduino ein kleiner Computer, der programmiert werden kann, um verschiedenste Elektronikkomponenten zu verbinden und zu steuern. Der Arduino verfügt über eine Reihe von Anschlüssen, die entweder als Eingang definiert werden können, was bedeutet, dass sie Daten von Bauteilen wie Schaltern, Tastern und Sensoren empfangen, oder als Ausgang, was bedeutet, dass mittels Daten Bauteile wie Motoren, Lampen und Summer gesteuert werden können. Diese Art programmierbare Entwicklungsplatine wird auch als Mikrocontroller bezeichnet.

Das Arduino-Projekt begann 2005 in Ivrea, Italien, mit dem Ziel, ein Gerät zu entwickeln, mit dem interaktive Designprojekte von Studenten gesteuert werden konnten und das kostengünstiger war als andere zu dieser Zeit verfügbare Prototyping-Systeme. Die Gründer Massimo Banzi und David Cuartielles benannten das Projekt nach einer Bar namens »Arduino« (ein italienischer Männervorname, der »starker Freund« bedeutet).

Die Arduino-Platine besteht aus zwei Hauptelementen: der Hardware oder dem Mikrocontroller, dem Gehirn der Platine, und der Software, mit dem Sie das Gehirn programmieren. Die Software wird auch als integrierte Entwicklungsumgebung (IDE) bezeichnet und kann kostenlos heruntergeladen werden.

Die IDE ist eine einfache Schnittstelle, die auf einem Computer unter Windows, OS X oder Linux läuft. Mit der IDE können Sie einen Sketch erstellen (ein Arduino-Programm) und es dann mit einem PC und einem USB-Kabel auf die Arduino-Platine hochladen. Der Sketch sagt der Hardware, was sie tun soll. Hardware und Software werde ich in den nächsten Kapiteln detaillierter beschreiben.

Der Arduino kann mit Batterien, über USB oder über eine externe Stromquelle gespeist werden. Ist der Arduino einmal programmiert, können Sie ihn von Ihrem Computer abkoppeln, denn mit seiner Stromversorgung funktioniert er selbstständig.

## Über dieses Buch

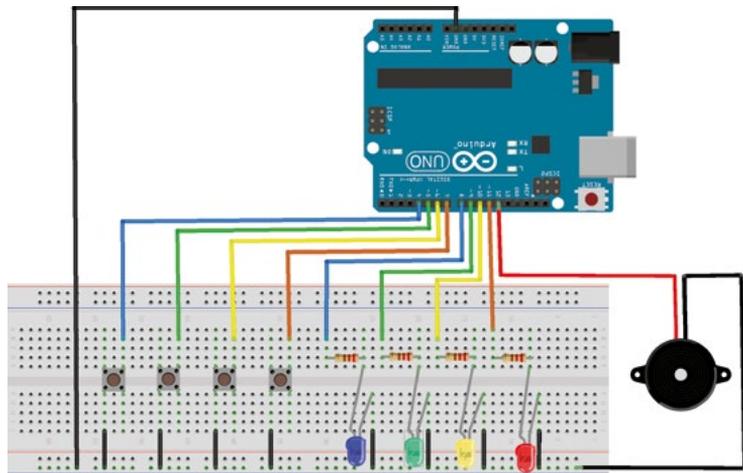
Warum habe ich dieses Buch geschrieben? Das Internet steckt voller Tutorien, Videos und Artikel, die den Arduino und damit mögliche Projekte beschreiben. Viele davon sind jedoch nicht detailliert genug

oder der notwendige Programmcode fehlt. Wie das Buch über technische Spielereien, das mich vor vielen Jahren inspiriert hat, soll mein Buch Ihnen dabei helfen, durch die gelernten Fähigkeiten und Techniken eigene Projekte auf die Beine zu stellen.

In diesem Buch werden Sie die meisten Projekte mithilfe eines Steckbretts konstruieren. So verstehen Sie am besten, wie Schaltungen aufgebaut werden, denn ihre Verbindungen sind flexibel: Wenn Sie einen Fehler machen, können Sie einfach das Kabel oder Bauteil umstecken und es noch einmal versuchen. In jedem Projekt wird schrittweise erklärt, wie die Hauptkomponenten angeschlossen werden, und Fotografien zeigen Ihnen den Aufbau. Als Schnellreferenz finden sich für die meisten Projekte auch Tabellen zu den Verbindungen.

Die Projekte verfügen über einen Schaltplan, in dem die Verbindungen exakt gezeigt werden, so wie in Abbildung 1. Sie sind mit einem Programm namens Fritzing erstellt worden (<http://www.fritzing.org>), einer kostenlosen Open-Source-Software, mit der Sie Ihren Projektaufbau optisch visualisieren können.

**Abbildung 1:**  
Beispiel eines Fritzing-  
Diagramms



Zu jedem Projekt gibt es auch den für den Arduino notwendigen Programmcode, sodass Sie vorher keine Programmiersprache erlernen müssen. Die ersten Projekte enthalten einfache Erklärungen darüber, was im Programmcode geschieht, damit Sie auch Ihre eigenen Veränderungen daran vornehmen können, wenn Sie möchten. Sie müssen den Code nicht abtippen, sondern können ihn unter <http://www.dpunkt.de/arduino-projekt> herunterladen.

Die Projekte in diesem Buch fangen bei den Grundlagen an und werden dann immer komplexer. Allerdings steigt dieses Buch nicht tief in die Theorie der Elektronik oder Programmierung ein, aber es bietet Ihnen einen guten Ausgangspunkt dafür. In diesem Buch möchte ich Ihnen zeigen, wie

Sie Ihre eigenen Projekte entwickeln. Mit einigen technischen Grundlagen im Gepäck können Sie sich auf den kreativen Schaffensprozess konzentrieren. Die Idee dahinter ist, dass ein Verständnis der Funktionsweise von Schaltungen Ihre Fantasie beflügeln kann, wie diese Schaltungen in der Praxis eingesetzt werden können.

Dieses Buch bietet Ihnen beispielsweise Informationen über die Pinbelegungen des Arduino, die Sie auch für andere Projekte verwenden können. Sie können auch Projekte miteinander kombinieren, um kompliziertere und spannendere Spielzeuge zu entwickeln.

Viele Arduino-Bücher stellen die Programmierung in den Vordergrund, und das hat auch seine Berechtigung. Ich bin jedoch der Meinung, dass man Elektronik auch über »Plug and Play« erlernen kann. Indem Sie meine Projekte nachbauen, erfolgt das Lernen Schritt für Schritt.

## **Der Aufbau dieses Buchs**

Dieses Buch beginnt mit einfachen Projekten und steigert sich wie nachfolgend aufgeführt hin zu komplexeren Projekten, sodass es Ihnen dabei hilft, immer mehr über Bauteile zu erfahren und Ihre Fähigkeiten zu entwickeln.

### **Teil 1: LEDs**

Zuerst lernen Sie, wie Sie einfache LEDs mit Schaltern und variablen Widerständen steuern, und kombinieren die Bauteile dann, um Stroboskoplichter, Pflanzen-Monitore, die Ihnen mitteilen, wann Sie gießen müssen, und sogar einen Geisterdetektor zu bauen.

### **Teil 2: Sound**

In diesem Teil lernen Sie den Piezo-Summer kennen, ein sehr nützliches Bauteil, das Töne erzeugt und sie auch erkennen kann. Sie werden mit dem Arduino Musik machen, ein einfaches, aber lustiges Gedächtnisspiel bauen und ein Schloss mit Geheimcode entwickeln, das erkennt, wie stark geklopft wird.

### **Teil 3: Servomotoren**

Alle Projekte in diesem Teil verwenden Servomotoren. Das sind kleine Motoren mit einem Arm, die für alle möglichen Dinge eingesetzt werden können. Sie bauen einen Joystick-gesteuerten Laser und nutzen eine Fernbedienung, um mit ihr die Servomotoren zu steuern.

### **Teil 4: LCDs**

In vielen Projekten ist eine LCD-Anzeige nützlich, um Meldungen und Ergebnisse darzustellen. In diesem Teil lernen Sie, wie Sie einen LCD-Bildschirm einrichten, bauen eine Wetterstation und entwickeln zwei Spiele: einen Wahrsager und ein Messgerät für Ihre Reaktionszeit.

## **Teil 5: Numerische Zähler**

In diesem Teil verwenden Sie 7-Segment-Anzeigen, um einen elektronischen Würfel und eine Countdown-Anzeige für ein Raketenstartsystem zu bauen, das eine Lunte zündet.

## **Teil 6: Sicherheit**

Diese etwas komplexeren Projekte zeigen Ihnen, wie Sie Ihr Haus mit Stolperdrähten und Einbruchalarmen schützen, mit Bewegungsmeldern Alarm auslösen oder Raketen abfeuern und Sicherheitssysteme bauen, die Tastenfelder und Kartenleser verwenden, um den Zugang zu beschränken.

## **Teil 7: Fortgeschrittene Projekte**

In diesem letzten Teil kombinieren wir den Arduino mit einer LED-Matrix, um eine Lightshow in Regenbogenfarben zu bauen. Zum Schluss kompletieren Sie Ihre Kenntnisse, indem Sie für zukünftige Projekte Ihren eigenen Arduino bauen.

Die Projekte müssen nicht in der angegebenen Reihenfolge gebaut werden. Wenn Sie etwas sehen, was Ihnen gefällt, und Sie sich sicher genug fühlen, es zu bauen, dann legen Sie los. Ich empfehle, zuerst einige der einfacheren Projekte auszuprobieren, denn dort finden Sie Informationen, die das Verständnis der späteren Projekte erleichtern.

Als ich mit dem Arduino zu experimentieren begann, habe ich ein Buch wie dieses gesucht, aber nicht gefunden. Für mich war das der Anreiz, das vorliegende Buch zu schreiben. Ich hoffe, Sie haben bei der Lektüre und beim Bau der Projekte genauso viel Spaß wie ich beim Schreiben.



ARDUINO

BOARD MODEL  
**UNO** R3  
ELECTRONICS

OPEN-SOURCE PLATFORM  
PROTOTYPING

MADE IN ITALY



WWW.ARDUINO.CC



ROHS  
TO  
COMPLIANT  
CARBON FOOTPRINT  
ZERO

# Projekt 0

## Erste Schritte

Bevor Sie mit dem Arduino erste Projekte bauen können, benötigen Sie einige Grundkenntnisse. Wir werfen also jetzt erst einmal einen Blick auf die Hard- und Software, die in diesem Buch verwendet wird, und sehen uns an, wie sie auf einem Computer eingerichtet wird. Anschließend testen Sie Ihren Arduino mit einem einfachen LED-Projekt und probieren einige hilfreiche Techniken, wie z.B. das Löten und das Herunterladen nützlicher Codebibliotheken.

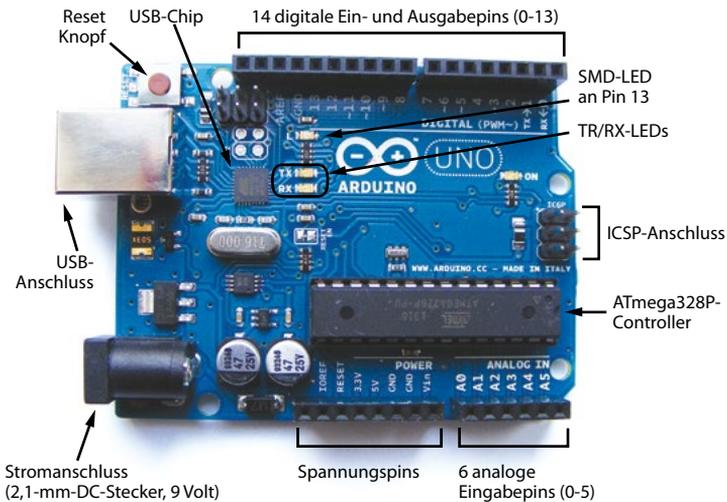
## Hardware

Zuerst schauen wir uns die Arduino-Uno-Platine und typische Komponenten an, die Sie in fast jedem Projekt verwenden werden.

### Der Arduino Uno

Es gibt im Handel unzählige Arduino-Platinen. Dieses Buch behandelt ausschließlich die meistverkaufte Variante, den Arduino Uno, der in Abbildung 0-1 gezeigt wird. Der Arduino Uno ist quelloffen (die Entwicklung darf also frei kopiert werden), weshalb Sie neben dem Original für ca. 25 € auch kompatible Platinen für ca. 15 € finden.

Sehen wir uns jetzt die verschiedenen Elemente des Arduino Uno an.



**Abbildung 0-1:**

Die Platine  
des Arduino Uno

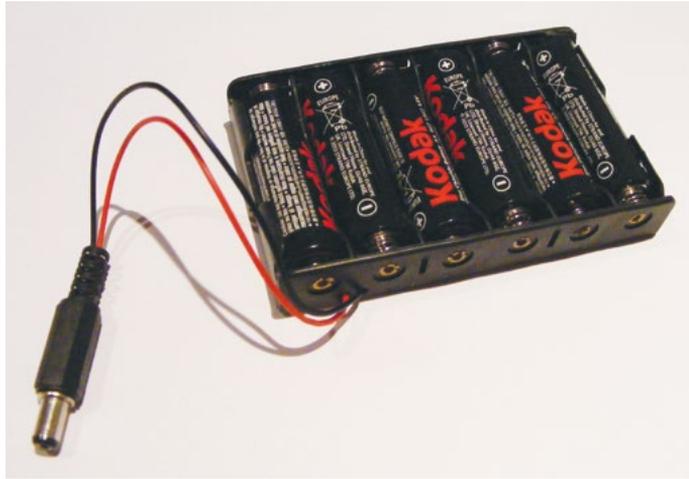
Der Arduino steuert Bauteile, z.B. Motoren oder LEDs, die Sie an ihn anschließen, indem er Ihnen Informationen als Ausgabe bereitstellt (Informationen werden also aus dem Arduino herausgesendet). Daten, die der Arduino von einem Sensor liest, stellen Eingaben dar (Informationen, die in den Arduino hineinkommen). Es gibt 14 digitale Ein- und Ausgabepins (Pin 0-13). Jeder kann entweder als Ein- oder als Ausgang konfiguriert werden. Eine Übersicht über die vollständige Pinbelegung befindet sich in Anhang B.

### Stromversorgung

Der Arduino Uno wird über den USB-Anschluss Ihres Computers mit Strom versorgt, wenn Sie ihn an Ihren PC anschließen, um beispielsweise ein Programm hochzuladen. Ist der Arduino nicht an Ihren PC angeschlossen, können Sie ihn mit einem 9-V-Adapter oder einem 9-Volt-Batteriepaket mit 2,1-mm-DC-Anschluss mit Strom versorgen, bei dem der mittige Anschluss mit plus verbunden ist, wie in Abbildung 0-2 gezeigt. Stecken Sie den Stecker einfach in den Stromanschluss des Arduino.

**Abbildung 0-2:**

Ein 9-V-Batteriepaket, mit dem Sie den Arduino mit Strom versorgen können

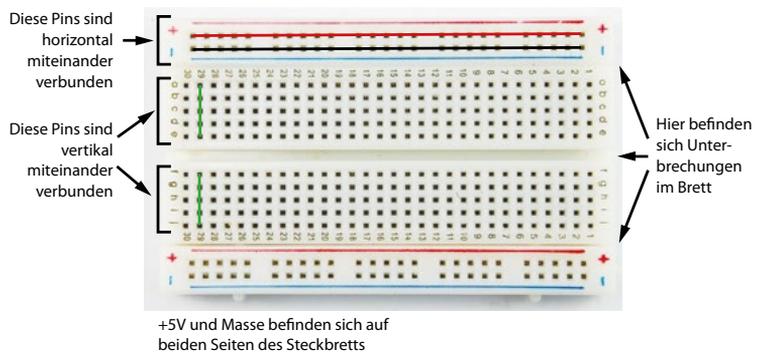


**Steckbretter**

Ein Steckbrett, auch Breadboard genannt, dient als Konstruktionsbasis für Elektronik-Prototypen. Alle Projekte dieses Buchs werden auf einem Steckbrett aufgebaut und nicht zusammengelötet.

Das Wort Steckbrett geht zurück auf die Zeiten, in denen Elektronikprojekte noch auf einem Holzbrett aufgebaut wurden. In das Brett wurden Nägel gehämmert und daran die Bauteile befestigt – ohne LötKolben. Die heutigen Steckbretter (siehe Abbildung 0-3) sind aus Plastik mit vorgebohrten Löchern (Verbindungspunkte genannt), in die Sie Bauteile oder Drähte stecken können, die dann von Klammern festgehalten werden. Die Bohrungen sind durch Streifen leitfähigen Materials verbunden, die sich unter dem Brett befinden.

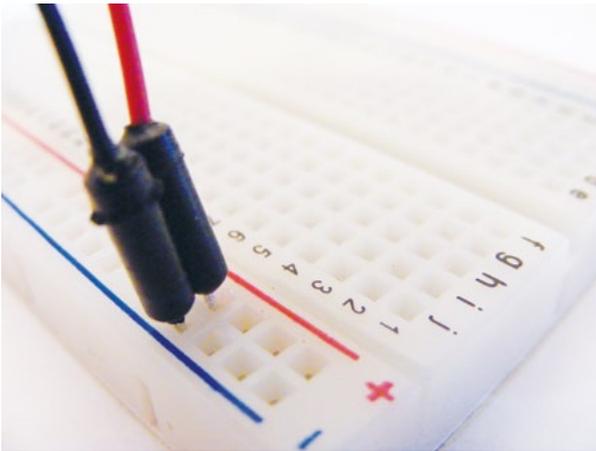
**Abbildung 0-3:**  
Anschlüsse an einem Steckbrett



Steckbretter gibt es in verschiedenen Größen. Um die Projekte diese Buchs zu bauen, benötigen Sie vier Steckbretter: zwei große, zumeist mit 830 Bohrungen, ein halbes mit 420 Bohrungen und ein Mini-Steckbrett mit 170 Bohrungen. Die großen eignen sich ideal für Projekte mit LCD-Anzeige oder vielen Bauteilen und die halben und Mini-Bretter mehr für kleinere Projekte.

Für die Projekte dieses Buchs empfehle ich den Kauf von Steckbrettern wie in Abbildung 0-3. In der Mitte des Bretts gibt es eine Unterbrechung, die Sie oft mit Komponenten überbrücken können. Die Unterbrechung hilft Ihnen dabei, die Pins einzeln anzuschließen, sodass sie nicht unbeabsichtigt kurzgeschlossen werden, was zum Scheitern des Projekts oder zur Beschädigung von Bauteilen führen kann.

Die blauen und roten Linien oben und unten sind Stromschienen, mit denen Sie die ins Steckbrett eingesteckten Bauteile mit Strom versorgen können (Abbildung 0-4). Die Stromschienen verbinden alle Bohrungen der Schiene horizontal, wobei die roten Linien für die Versorgungsspannung und die blauen für Masse (auch GND genannt) stehen.



## Jumperkabel

Mit Jumperkabeln stellen Sie Verbindungen auf dem Steckbrett her. Es handelt sich um Drähte mit kleinen Halterungen am Ende, die das Einstecken ins Steckbrett und Rausziehen erleichtern. (Sie können auch eigene Kabel verwenden, es sollten aber feste Drähte sein, die sich in die Klammern einführen lassen.)

Wenn Sie ein Jumperkabel in die Bohrung eines Steckbretts einstecken, wird es mit einer kleinen Federklemme festgehalten und eine elektrische Verbindung hergestellt, die sich meist über fünf Bohrungen erstreckt. Dann können Sie weitere Bauteile in benachbarte Bohrungen stecken, um eine Schaltung aufzubauen (siehe Abbildung 0-5).

### TIPP

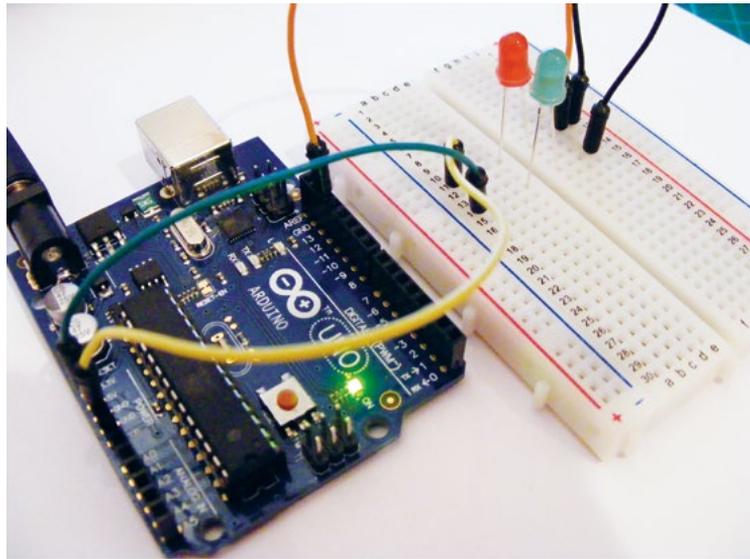
*Meist werden rote Drähte für 5-V-Versorgungsspannung verwendet und schwarze für Masse (GND), sodass Sie sie leicht erkennen können. Für alle anderen Leitungen können Sie beliebige Farben verwenden.*

### Abbildung 0-4:

Stromschienen für Versorgungsspannung und Masse

**Abbildung 0-5:**

Eine kleine Schaltung auf dem Steckbrett

**HINWEIS**

*Da sich die IDE-Versionen sehr schnell ändern können, erläutere ich nicht, wie sie zu installieren sind. Sie werden aber feststellen, dass dies sehr einfach zu bewerkstelligen ist. Alle Versionen der IDE und genaue Installationshinweise für Ihr Betriebssystem finden*

*Sie online unter <http://www.arduino.cc/>.*

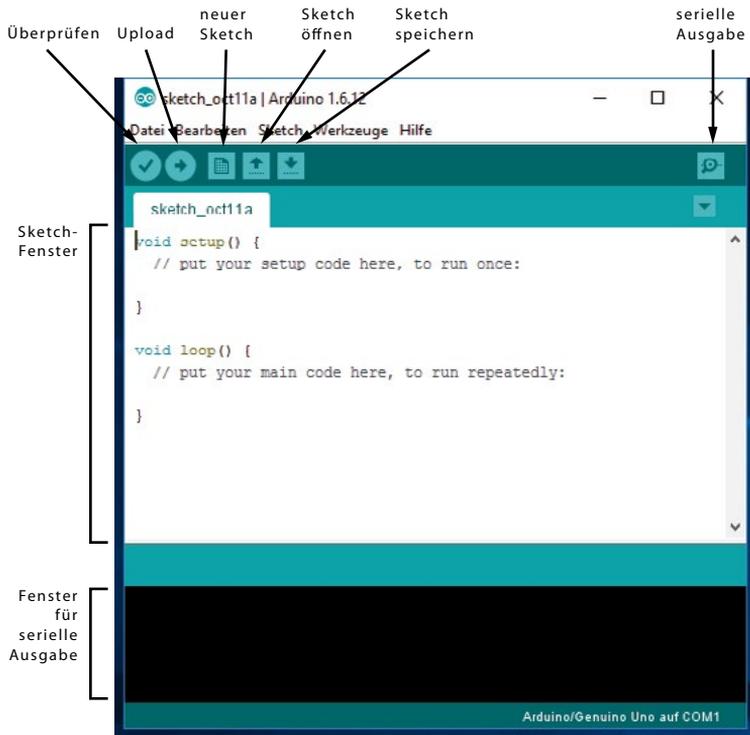
## Programmierung des Arduino

Damit unsere Projekte das machen, was wir wollen, müssen wir Programme schreiben, die dem Arduino entsprechende Anweisungen geben. Dazu verwenden wir ein Werkzeug, das sich integrierte Entwicklungsumgebung nennt (Integrated Development Environment – IDE). Die Arduino-IDE kann kostenlos unter <http://www.arduino.cc/> heruntergeladen werden und läuft unter Windows, OS X und Linux. Sie ermöglicht Ihnen, Computerprogramme zu schreiben (als eine Reihe von Schritt-für-Schritt-Anweisungen, die in der Arduino-Welt Sketche genannt werden), die Sie dann mithilfe eines USB-Kabels auf Ihren Arduino hochladen. Ihr Arduino wird die Befehle in Form von Interaktionen mit der Außenwelt ausführen.

## Die IDE-Schnittstelle

Wenn Sie die Arduino-IDE öffnen, sollte sie ähnlich aussehen wie in Abbildung 0-6.

Die IDE besteht oben aus einer Werkzeugleiste mit Schaltflächen für die am meisten verwendeten Funktionen. In der Mitte liegt das Code- oder Sketch-Fenster, in dem Sie Ihre Programme schreiben und ansehen können. Unten befindet sich das Fenster für die serielle Ausgabe. Dort werden Kommunikationsmeldungen zwischen PC und Arduino ausgegeben sowie Fehlermeldungen, die beim Kompilieren eines Sketches auftreten können.



**Abbildung 0-6:**  
Die Arduino-IDE

## Arduino-Sketche

Für jedes Projekt zeige ich Ihnen den dazugehörigen Sketch und erkläre ihn. Alle Sketche stehen auch zum Download bereit unter: <http://www.dpunkt.de/arduino-projekt>.

Wie jedes Programm bestehen Sketche aus sehr strengen Anweisungen und sind daher sehr fehleranfällig. Um zu prüfen, ob Sie den Sketch richtig kopiert haben, klicken Sie oben im Fenster auf den grünen Haken. Es handelt sich dabei um die Schaltfläche **Überprüfen**, die eine Fehleranalyse durchführt und Ihnen im seriellen Ausgabefenster mitteilt, ob der Sketch korrekt kompiliert wurde. Wenn Sie nicht weiterkommen, können Sie den Sketch jederzeit herunterladen und in die IDE einfügen.

## Bibliotheken

In der Arduino-Welt ist eine Bibliothek ein kleines Stück Programmcode, der eine bestimmte Funktion ausführt. Statt diesen Code mehrfach in Ihren Sketchen zu verwenden, können Sie einen Befehl hinzufügen, der den Code aus der Bibliothek aufruft. Das spart Ihnen Zeit und erleichtert die Verbindung zu Sensoren, Anzeigen oder Modulen.

Die Arduino-IDE enthält eine Reihe eingebauter Bibliotheken, zum Beispiel LiquidCrystal, mittels derer Sie bequem mit LCD-Anzeigen kommunizieren, viele weitere sind online verfügbar. Für die Projekte dieses