

Philip &  
Christian Caroli

# SCHNELLEINSTIEG ARDUINO™

ARDUINO™-  
PROJEKTE OHNE  
VORWISSEN  
SELBST  
UMSETZEN



ALLES ÜBER BOARDS, SHIELDS,  
INSTALLATION, PROGRAMMIERUNG  
UND ELEKTRONIK FÜR DIE PRAXIS

FRANZIS

Philip & Christian Caroli  
**Schnelleinstieg**  
**Arduino™**

**Philip Caroli** studiert Informatik am KIT in Karlsruhe und beschäftigt sich schon seit vielen Jahren mit Elektronik und 3-D-Druck. Angefangen bei kleinen elektronischen Schaltungen über Reparaturen von Haushaltsgeräten bis hin zur Entwicklung von kleinen Robotern, hat Philip Caroli alles repariert, erweitert und programmiert, was ihm in die Hände fiel. Als Autodidakt hat er einen sehr praxisbezogenen Zugang zur Materie und kann seine Freude am Thema gut vermitteln. In seiner Freizeit ist er im FabLab Karlsruhe e.V. engagiert und hat dort unter anderem einen Lasercutter mit aufgebaut.

**Christian Caroli** ist bereits seit seiner Jugend begeisterter Elektronik-Bastler. Was zunächst mit dem Auseinanderschrauben und Wiederzusammenbauen seines Computers begann, setzte er in späteren Jahren mit Zusatzmodulen für seinen PC, diversen Mikrocontroller-Schaltungen und Kleingeräten bis zum heutigen Bau von 3-D-Druckern fort. Vor zwei Jahren hat der Bastler aus Leidenschaft den Verein FabLab Karlsruhe e.V. mitgegründet und mitaufgebaut, dessen Ziel es ist, den Umgang mit modernen Produktionsmitteln wie 3-D-Druckern und Lasercuttern zu lehren und zu ermöglichen. Hauptberuflich entwickelt Christian Caroli Soft- und Hardware in seiner eigenen Firma.

## Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

© 2015 Franzis Verlag GmbH, 85540 Haar bei München

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

**Arduino ist ein eingetragenes Markenzeichen der Arduino S.r.l.**

**Programmleitung:** Dr. Markus Stäuble

**Satz und Layout:** Nelli Ferderer, [nelli@ferderer.de](mailto:nelli@ferderer.de)

**art & design:** [www.ideehoch2.de](http://www.ideehoch2.de)

**Druck:** CPI-Books

Printed in Germany

**ISBN 978-3-645-60427-7**

# INHALT

<b>1. ARDUINO™-BOARDS UND -SHIELDS</b> .....	10
1.1 Hast du eine Idee, dann setze sie um! .....	10
1.2 Arduino Uno™ Revision 3: die Referenz .....	10
1.3 Alle Arduino™-Boards auf einen Blick .....	13
1.4 Erweiterungsplatinen: Arduino™-Shields .....	16
1.4.1 Proto-Shield für selbst gelötete Experimente .....	16
1.4.2 Ethernet-Shield für lange Verbindungsstrecken .....	17
1.4.3 Motor-Shield für die Ansteuerung von Motoren .....	18
1.4.4 Wi-Fi-Shield für die Drahtlosübertragung .....	20
<b>2. ARDUINO™-SOFTWARE SCHREIBEN</b> .....	22
2.1 Aufbau und Ablauf eines C-Programms .....	22
2.2 Kommunikation zwischen PC und Arduino™ .....	23
2.3 Installation der Entwicklungsumgebung .....	23
2.3.1 Arduino™-Setup auf einem Windows-PC .....	24
2.3.2 Arduino™-Setup auf einem Apple Mac .....	27
2.4 Einrichten der Entwicklungsumgebung .....	29
2.4.1 Zwei Einstellungen, ohne die es nicht geht .....	30
2.4.2 Erster Test mit einem Beispielprogramm .....	31
2.5 Erste Schritte in der C-Programmierung .....	32
2.5.1 Dem Prozessor Befehle geben und kommentieren ....	32
2.5.2 Wie man sich in den Initialisierungsprozess einklinkt ..	35

2.6	Ein Programm schreiben und installieren .....	37
2.6.1	Variablen als Platzhalter für Befehle nutzen .....	40
2.6.2	Schleifen als Kontrollinstrument einsetzen .....	45
2.6.3	Mit der if-Abfrage Bedingungen festlegen .....	48
2.6.4	Mit Funktionsaufruf Redundanzen vermeiden .....	54
2.6.5	String-Variablen für die Textausgabe nutzen .....	59
2.6.6	Felder definieren die Länge des Textfelds .....	61
2.6.7	Fehlerteufel mit serieller Ausgabe aufspüren .....	62
2.6.8	Weiterführende Hilfen für Entwickler .....	67
<b>3.</b>	<b>EINFÜHRUNG IN DIE ELEKTRONIK .....</b>	<b>68</b>
3.1	Spannung, Stromstärke, Widerstand .....	68
3.1.1	Potenzial und Potenzialausgleich .....	68
3.1.2	Volt: Spannung zwischen Plus und Minus .....	70
3.1.3	Ampere: die Stromstärke elektrischer Leiter .....	71
3.1.4	Ohm: der elektrische Widerstand .....	72
3.1.5	Rechnen mit Volt, Watt, Ampere, Ohm .....	74
3.2	Dioden geben dem Strom die Richtung .....	76
3.3	Kondensatoren speichern den Strom .....	77
3.3.1	Ausführungen und Bauformen von Kondensatoren ....	78
3.4	Transistoren verstärken und schalten .....	79
3.5	Breadboard-Schaltungen, ohne zu löten .....	81

---

<b>4. LÖTEN WIE DIE PROFIS</b> .....	82
4.1 Werkzeug, das man zum Löten braucht .....	82
4.1.1 LötKolben und Elektroniklötzinn .....	82
4.1.2 LötSchwamm zum Säubern des Kolbens .....	84
4.1.3 Eine Spitzzange gegen Brandblasen .....	85
4.1.4 Ein Seitenschneider und eine LötHilfe .....	85
4.1.5 Stahlwolle, um Korrosion zu entfernen .....	87
4.1.6 Kabel, Kupferlitze und Silberdraht .....	87
4.1.7 Schrumpfschlauch gegen blanke Kabel .....	88
4.2 Tipps für fachgerechtes Löten .....	90
<b>5. ARDUINO™ IM PRAXISEINSATZ</b> .....	92
5.1 Bau eines analogen Temperaturwarners .....	92
5.1.1 Motivation .....	93
5.1.2 Aufgabenstellung .....	93
5.1.3 Bauteile vorgestellt .....	93
5.1.4 Breadboard-Aufbau .....	96
5.1.5 Source Code .....	98
5.2 Bau einer analogen Steampunkuhr .....	102
5.2.1 Motivation .....	103
5.2.2 Aufgabenstellung .....	104
5.2.3 Exkurs Schrittmotor .....	104

5.2.4	Breadboard-Aufbau	109
5.2.5	Schaltplan	110
5.2.6	Externes Netzteil	110
5.2.7	Schrittmotor anschließen	111
5.2.8	Schalter anschließen	112
5.2.9	Ziffernblatt gestalten	112
5.2.10	Source Code	113
5.3	Bau einer rotierenden LED-Blume	121
5.3.1	Motivation	121
5.3.2	Aufgabenstellung	122
5.3.3	Alkaline-Batterien	123
5.3.4	Nickel-Metalhydrid-Akkus	123
5.3.5	Lithium-Ionen-Akkus	124
5.3.6	Blinkenlight-Shield und Schaltplan	125
5.3.7	LEDs und Widerstände	126
5.3.8	Mobile Stromversorgung	126
5.3.9	Funktionstest durchführen	127
5.3.10	Lüfter oder Motor	127
5.3.11	Source Code	128
5.4	Bau eines LCD-Textdisplays	133
5.4.1	Motivation	134
5.4.2	Aufgabenstellung	134
5.4.3	Poti für die Spannungseinstellung	135
5.4.4	Aufbau der Schaltung	136
5.4.5	Programm laden	139
5.4.6	Source Code	139

---

5.5	Bau einer zuverlässigen Wetterstation .....	143
5.5.1	Motivation .....	143
5.5.2	Aufgabenstellung .....	144
5.5.3	Arduino™-Wi-Fi-Shield .....	144
5.5.4	Sensor DHT22 .....	145
5.5.5	Sensor BMP085 .....	146
5.5.6	Breadboard-Aufbau .....	147
5.5.7	Source Code .....	148
<b>INDEX</b>	.....	<b>156</b>

Die Arduino-Erfolgsgeschichte nahm im Jahr 2005 im norditalienischen Ivrea, einer Kleinstadt nahe Turin, ihren Anfang. Professor Massimo Banzi hatte als gelernter Softwarearchitekt den Auftrag, seinen Studenten neue Wege des interaktiven Designs beizubringen, auch Physical Computing genannt. Mit seinem Team entwickelte er für die studentische Zielgruppe eine leicht zu bedienende und zu bestückende Hardwareplatine »in Blau«, die den Einstieg in die Elektronik erleichtern sollte, und das, ohne sich vorher mit Algebra auseinandersetzen zu müssen. Do it yourself! Hast du eine Idee, dann setze sie um. Die Keimzelle für die Arduino-Entwicklungsplattform war angelegt.

## 1.1 Hast du eine Idee, dann setze sie um!

Die ersten Platinen waren schnell verteilt. Der Bekanntheitsgrad der kleinen blauen Boards nahm derart schnell zu, dass man mit der Produktion kaum noch nachkam. Immer mehr Studenten und Elektronikbastler wollten auch so ein Gerät haben – und das Kind brauchte einen Namen, der aber schnell gefunden war: Arduino, benannt nach der Lieblingsbar der am Projekt beteiligten Personen.

Übrigens – eines der ersten Projekte war ein Wecker, der an der Decke an einem Kabel aufgehängt war. Drückte man die Snooze-Taste, zog er sich spöttisch ein Stück höher, bis nichts anderes mehr übrig blieb, als sich komplett zu erheben, um ihn auszuschalten.

## 1.2 Arduino Uno™ Revision 3: die Referenz

Verlieren wir keine Zeit und beginnen wir direkt mit dem Stoff, aus dem unsere Träume sind: mit dem Arduino Uno, der am weitesten verbreiteten Platine. Der Uno steht stellvertretend für alle anderen Arduino-Boards, denn es gibt mittlerweile so viele, dass sie alle zu besprechen den Rahmen dieses Buchs sprengen würde.

Der Arduino Uno ist als Nachfahre in direkter Linie des ursprünglichen Ur-Arduino entstanden. Er wurde über die Jahre stets weiterentwickelt und verbessert und ist in der Revision 3 der jüngste Spross der längsten Arduino-Entwicklungsreihe. Gleichzeitig ist er auch das am besten getestete Board und dient als Referenz für fast alle Software-Bibliotheken. Da er so zahlreich produziert wird, ist er auch eines der günstigsten Arduino-Boards auf dem Markt.



Der Arduino Uno in der SMD-Ausführung - klein, günstig, universell.

Auf dem Uno sitzt ein ATmega328-Mikrocontroller, der mit 16 MHz (Megahertz) betrieben wird. Er hat 20 digitale Anschlüsse, die wahlweise als Eingang (Input) oder Ausgang (Output) verwendet werden können, bei sechs von ihnen kann eine Pulsweitenmodulation hinzugeschaltet werden, bei sechs anderen können auch analoge Signale ausgelesen werden.

Über einen USB-Stecker kann man den Uno an den Computer anschließen. Da USB praktischerweise auch eine Stromquelle mit maximal 500 mA (Milliampere) bereitstellt, ist es möglich, das Board darüber mit Strom zu versorgen, was für einfache Schaltungen oft ausreichend ist. Bei umfangreicheren oder stromhungrigeren Projekten kann der Uno über den separaten Stromanschluss an eine variable Spannung von 7 bis 12 V (Volt) angeschlossen werden – die notwendigen 5 V für den Mikrocontroller werden über den integrierten Spannungswandler erzeugt.



### ZU VIEL STROM AUS DEM USB-ANSCHLUSS?

Sollten Sie einmal aus Versehen zu viel Strom aus dem USB-Anschluss ziehen, müssen Sie sich um Ihren Computer keine Sorgen machen, denn der Arduino trennt die Stromversorgung bei Kurzschlüssen und Überstrom automatisch, bis der Fehler behoben wurde.

Natürlich hat ein Arduino auch einen Reset-Knopf und verfügt über einen ICSP-Anschluss, über den ein Programm auf den Mikrocontroller übertragen werden kann. Da aber jeder Arduino von Haus aus auch mit einem Bootloader

#### LISTE DER TECHNISCHEN DATEN

<b>Mikrocontroller</b>	ATmega328
<b>Flash-Speicher</b>	32 KByte und 0,5 KByte für den Bootloader
<b>SRAM</b>	2 KByte
<b>EEPROM</b>	1 KByte
<b>Prozessortakt</b>	16 MHz
<b>Betriebsspannung</b>	5 V
<b>Eingangsspannung</b>	7-12 V
<b>Eingangsspannung (max.)</b>	6-20 V
<b>Digitale I/O-Pins</b>	20
<b>Davon PWM</b>	6
<b>Davon analoger Eingang</b>	6 (10 Bit, 1.024 Abstufungen)
<b>Max. Strom pro I/O-Pin</b>	40 mA
<b>Max. Strom am 3,3-V-Pin</b>	50 mA
<b>Anschlüsse</b>	USB, Stromanschluss, ICSP
<b>Kommunikation</b>	USB (seriell), 1 x UART TTL (5 V) seriell, I <sup>2</sup> C, SPI
<b>Besonderheiten</b>	separater Mikrocontroller für die USB-Kommunikation

versehen ist, wird der ICSP-Anschluss selten benötigt, da man seine Programme mithilfe des Bootloaders über den USB-Anschluss übertragen kann, was wesentlich komfortabler ist.

Kommunizieren kann der Uno mit der Außenwelt zum einen über USB, zum anderen auch über die serielle Schnittstelle UART TTL (5 V), die in den Pins 0 und 1 untergebracht ist. Der Mikrocontroller unterstützt ebenfalls I<sup>2</sup>C- und SPI-Kommunikation, die z. B. den Anschluss digitaler Sensoren ermöglichen.

Als Besonderheit der neueren Generationen ist der separate ATmega16U2-Mikrocontroller zu erwähnen, der die recht komplexe USB-Kommunikation mit dem Computer übernimmt und die bisherigen FTDI-Treiber ersetzt, die manchmal Probleme bereiteten und vor allem nicht so schnell waren.

Den Arduino Uno Revision 3 gibt es in zwei verschiedenen Varianten. Bei der SMD-Edition steckt der Prozessor in einem SMD-Gehäuse und ist fest mit der Platine verbunden. Geht er einmal kaputt, können Sie ihn nicht austauschen und müssen die gesamte Platine ersetzen. Besser geeignet ist daher die normale Version, bei der der Prozessor in einem DIL-Gehäuse auf einem Sockel sitzt. Hier kann der Prozessor einfach ausgetauscht werden, allerdings müssen Sie dann einmalig den Bootloader über den ICSP-Anschluss auf den jungfräulichen Mikrocontroller aufspielen.

#### GEMEINSAMKEITEN ALLER ARDUINOS

Es gibt ein paar Dinge, die alle Arduinos gemeinsam haben: Alle besitzen einen USB-Anschluss, alle haben eine LED, die an Pin 13 angeschlossen ist, alle haben einen Reset-Schalter, alle lassen sich über die Arduino-Entwicklungsumgebung programmieren – und alle sind blau.



## 1.3 Alle Arduino™-Boards auf einen Blick

Heute, nach nur zehn Jahren, bietet Arduino eine fast unüberschaubare Menge an Möglichkeiten, die man ganz individuell und kreativ einsetzen kann. Egal ob man einfach nur den Umgang mit einem Mikrocontroller erlernen möchte, Elektronik mit Digitaltechnik verbinden will oder ein ganz konkretes Projekt im Kopf hat – mit Arduino können Sie fast alles machen.

ARDUINO	PROZESSOR	EINGANGS- SPANNUNG	BETRIEBS- SPANNUNG	CPU- SPEED	AUSGÄNGE DIGITAL
<b>Uno</b>	ATmega328	7-12 V	5 V	16 MHz	20
<b>Due</b>	AT91SAM3X8E	7-12 V	3,3 V	84 MHz	66
<b>Leonardo</b>	ATmega32U4	7-12 V	5 V	16 MHz	32
<b>Mega 2560</b>	ATmega2560	7-12 V	5 V	16 MHz	70
<b>Mega ADK</b>	ATmega2560	7-12 V	5 V	16 MHz	70
<b>Micro</b>	ATmega32U4	7-12 V	5 V	16 MHz	32
<b>Mini</b>	ATmega328	7-9 V	5 V	16 MHz	22
<b>Nano</b>	ATmega168 ATmega328	7-9 V	5 V	16 MHz 16 MHz	22
<b>Ethernet</b>	ATmega328	7-12 V	5 V	16 MHz	20
<b>Esplora</b>	ATmega32U4	7-12 V	5 V	16 MHz	
<b>ArduinoBT</b>	ATmega328	2,5-12 V	5 V	16 MHz	20
<b>Fio</b>	ATmega328P	3,7-7 V	3,3 V	8 MHz	22
<b>Pro (168)</b>	ATmega168	3,35-12 V	3,3 V	8 MHz	20
<b>Pro (328)</b>	ATmega328	5-12 V	5 V	16 MHz	20
<b>Pro Mini</b>	ATmega168	3,35-12 V 5-12 V	3,3 V 5 V	8 MHz 16 MHz	20
<b>LilyPad</b>	ATmega168V ATmega328V	2,7-5,5 V	2,7-5,5 V	8 MHz	20
<b>LilyPad USB</b>	ATmega32U4	3,8-5 V	3,3 V	8 MHz	9
<b>LilyPad Simple</b>	ATmega328	2,7-5,5 V	2,7-5,5 V	8 MHz	9
<b>LilyPad SimpleSnap</b>	ATmega328	2,7-5,5 V	2,7-5,5 V	8 MHz	9
<b>Yún</b>	ATmega32U4	5 V	5 V	16 MHz	32
<b>Robot (Control, oben)</b>	ATmega32U4	5 V	5 V	16 MHz	*
<b>Robot (Motor, unten)</b>	ATmega32U4	9 V	5 V	16 MHz	4
<b>Intel Galileo V1</b>	Intel Quark SoC X1000	5 V (Gen 1) 7-15 V (Gen 2)	5 V 5 V	400 MHz	20
<b>Zero</b>	ATSAMD21G18		3,3 V	48 MHz	20

DAVON ANALOG	DAVON PWM	AUSGÄNGE ANALOG	EEPROM (KB)	SRAM (KB)	FLASH (KB)	USB	UART
6	6	0	1	2	32	Typ B	1
12	12	2	-	96	512	2 Micro B	4
12	7	0	1	2,5	32	Micro B	1
16	15	0	4	8	256	Typ B	4
16	15	0	4	8	256	Typ B	4
12	7	0	1	2,5	32	Micro B	1
8	6	0	1	2	32	-	-
8	6	0	0,512 1	1 2	16 32	Mini B	1
6	4	0	1	2	32	Typ B	-
-		0	1	2,5	32	Micro B	-
6	6	0	1	2	32	-	1
8	6	0	1	2	32	Mini B	1
6	6	0	0,512	1	16	-	1
6	6	0	1	2	16	-	1
6	6	0	0,512	1	16	-	1
6	6	0	0,512	1	16	-	-
4	4	0	1	2,5	32	Micro B	-
4	5	0	1	2	32	-	-
4	5	0	1	2	32	-	-
12	7	0	1	2,5	32	Micro B	1
*	*	0	1	2,5	32	Typ B	???
1	1	0	1	2,5	32	Typ B	???
6	6	0	8	512	8192	Micro B Micro A	1
6	12	0	16	32	256	2 x Micro B	1



## RESSOURCEN ZUM BUCH ALS DOWNLOAD

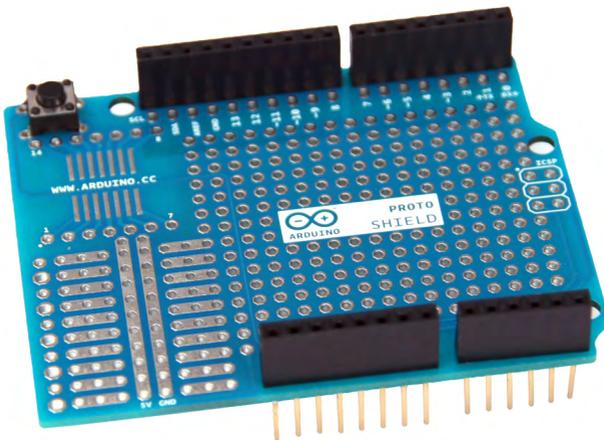
In diesem Buch finden Sie etliche Programme, die Bestandteil der beschriebenen Projekte sind. Natürlich sollen Sie in Zeiten des Internets nicht gezwungen werden, die Programme umständlich abzutippen, wir stellen sie Ihnen daher unter den Internetadressen *buch.cd* und *arduino-handbuch.visual-design.com* vollständig zur Verfügung.

## 1.4 Erweiterungsplatten: Arduino™-Shields

Im Laufe der Zeit hat das Arduino-Team verschiedene Erweiterungsplatten herausgegeben, mit denen Sie die Eigenschaften von fast jedem Arduino erweitern können. Man hat diesen Platinen den Namen Shield gegeben, um sie so von den eigentlichen Arduinos abzugrenzen. Einige dieser Shields stellen wir auf den folgenden Seiten vor.

### 1.4.1 Proto-Shield für selbst gelötete Experimente

Das sicherlich einfachste Shield aus der Arduino-Reihe ist das Proto-Shield. Es ist dazu gedacht, selbst gelötete Experimente auf möglichst einfache Art und Weise durchführen zu können.



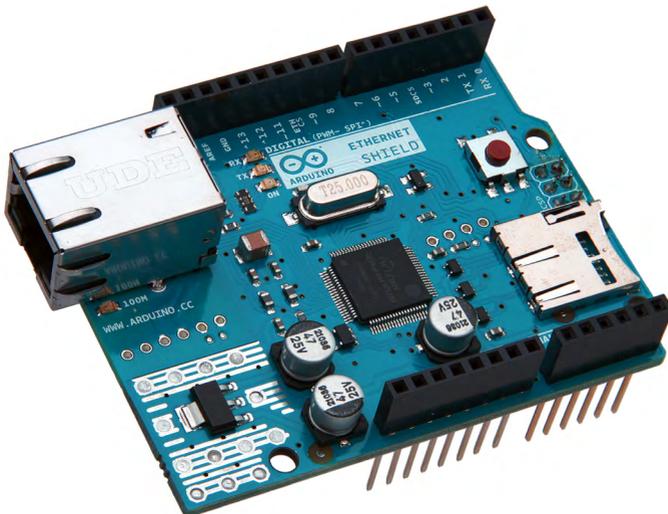
Das Proto-Shield erleichtert die Entwicklung von eigenen Schaltungen.

Die Platine ist ganz im üblichen Arduino-Design gehalten und bietet auf Vorder- und Rückseite Hunderte von Lötunkten, auf denen Sie Ihre Experimente gestalten können. Die Anschlüsse des Shields passen perfekt auf die des Arduino, und man ist nicht gezwungen, eine Stiftleiste umständlich um einen halben Rasterpunkt zu verschieben, was gar nicht so einfach ist. Alle Anschlüsse des Arduino lassen sich bequem über Lötunkte erreichen, die mit den Stiftleisten verbunden sind – Sie müssen also Ihre Kabel nicht umständlich direkt an die Stiftleisten löten. Darüber hinaus besitzt das Shield einen bereits korrekt verbundenen Reset-Schalter, der die Arbeit mit dem Proto-Shield deutlich erleichtert.

Das Proto-Shield ist ein nicht unbedingt notwendiges, aber doch recht praktisches Utensil zum Aufbauen neuer Schaltungen.

### 1.4.2 Ethernet-Shield für lange Verbindungsstrecken

Gerade wenn ein Arduino für die Erfassung von Sensordaten verwendet wird, ist es oft wichtig, diese Daten schnell und unkompliziert an andere Systeme weiterzuleiten. Doch leider gibt es keine USB-Kabel, die länger als 10 m sind, da die USB-Kommunikation nicht für längere Strecken ausgelegt ist. Wenn Sie also Ihren Arduino an einen Computer anschließen möchten, der weiter als 10 m entfernt ist, müssen Sie auf ein anderes System umsteigen – eine Möglichkeit hierzu bietet der Einsatz von Ethernet.



Das Ethernet-Shield kann sich in ein kabelgebundenes Netzwerk einklinken und mit SD-Karten umgehen.

Der Ethernet-Standard wird immer dann verwendet, wenn ein Computer über ein Kabel mit anderen Computern verbunden werden soll. Die Datenübertragungsrate liegt dabei entweder bei 10 MByte/s, bei 100 MByte/s oder bei 1.000 MByte/s, wobei die Kabel üblicherweise bis zu 100 m lang sein dürfen – das reicht sicherlich aus, um vom Dachboden in den Keller zu gelangen. Der Anschluss an das Ethernet-Netzwerk erfolgt über die üblichen RJ-45-Stecker, wobei das Shield auch die Stromversorgung über dieses Kabel erlaubt (Power over Ethernet, PoE).

Auf die Funktionen des Ethernet-Shields kann seitens der Arduino-Entwicklungsumgebung bequem über die Ethernet-Library zugegriffen werden, die all jene Protokolle beinhaltet, die noch nicht durch den Chip des Shields abgedeckt sind. Das Ethernet-Shield verfügt weiterhin über einen SD-Kartenleser, der es ermöglicht, auf die Daten einer handelsüblichen microSD-Karte zuzugreifen. So können beispielsweise Messwerte abgespeichert werden, falls die Ethernet-Verbindung einmal nicht zur Verfügung steht.

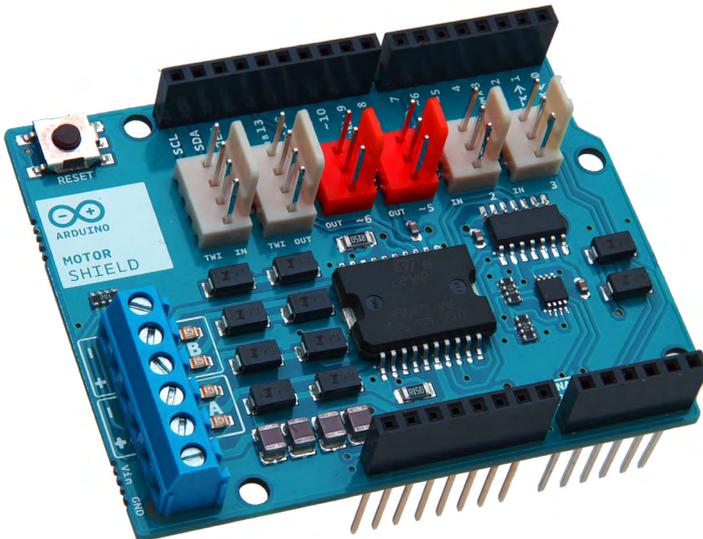
#### LISTE DER TECHNISCHEN DATEN

<b>Chip</b>	Wiznet W5100, 16 KByte Puffer
<b>Geschwindigkeit</b>	10/100 MByte/s
<b>Verwendeter Anschluss</b>	ausschließlich SPI
<b>Anschlüsse</b>	RJ-45-Netzwerkanschluss, SD-Kartenleser
<b>Besonderheiten</b>	bis auf SPI sind alle Anschlüsse durchgeschleift

### 1.4.3 Motor-Shield für die Ansteuerung von Motoren

Wenn Ihnen die bloße Elektronik zu langweilig ist und Sie etwas Action in Ihrem Leben haben möchten, ist das Motor-Shield vielleicht das Richtige für Sie. Es ermöglicht Ihnen, diverse Elektromotoren oder Servomotoren an Ihren Arduino anzuschließen und zu betreiben. Verwendet wird dabei ein L298-Chip, der in der Lage ist, zwei Gleichstrommotoren oder einen Schrittmotor anzusteuern.

Die Motoren können über Schraubanschlüsse direkt auf dem Board angeschlossen werden. Ebenso befinden sich vier dreipolige Anschlüsse auf der Platine, die neben Versorgungsspannung und Masse auch jeweils einen Pin des Arduino zur Verfügung stellen. Zwei davon (weiß) sind für analoge Eingänge reserviert und mit dem A2- sowie dem A3-Pin des Arduino verbunden. Die zwei roten Anschlüsse sind verbunden mit den Pins 5 und 6 des Arduino und sind so auch in der Lage, die PWM-Signale abzugeben. Damit ist es möglich, über diese beiden Anschlüsse auch Servomotoren anzusteuern.



Motoren und Servos kann man mit dem Motor-Shield problemlos ansteuern.

Außerdem hat das Motor-Shield zwei vierpolige Anschlüsse, die als TWI bezeichnet werden. Aus lizenzrechtlichen Gründen verwendet Atmel diese Bezeichnung für einen Bus, der kompatibel zum I<sup>2</sup>C-Bus ist und über den man zahlreiche Sensoren und andere elektronische Bauteile anschließen kann. Einer dieser beiden Anschlüsse ist für Eingänge gedacht, der andere für Ausgänge.

Der Anschluss für die externe Stromversorgung erlaubt es, die angeschlossenen Motoren mit bis zu 2 A pro Kanal anzusteuern. Dabei sollte eine Spannung von 7 bis 12 V verwendet werden, damit der angeschlossene Arduino nicht überlastet wird.

#### LISTE DER TECHNISCHEN DATEN

<b>Chip</b>	L298P
<b>Verwendete Anschlüsse</b>	A0, A1, A2, A3, D3, D5, D6, D8, D9, D11, D12, D13, SCL, SDA
<b>Anschlüsse</b>	2 Gleichstrommotoren (2 A) oder 1 Schrittmotor (4 A), 2 I <sup>2</sup> C-Ports, 2 analoge Eingänge (weiß), 2 digitale Ausgänge mit PWM, Stromanschluss (7-12 V)

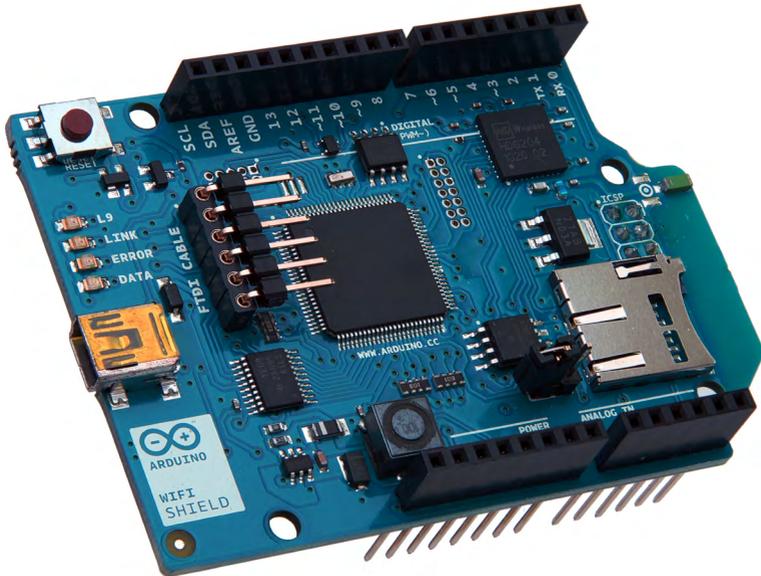
#### 1.4.4 Wi-Fi-Shield für die Drahtlosübertragung

Eine schöne Eigenschaft der Arduinos mit ihren kleinen Mikroprozessoren ist, dass sie relativ wenig Strom verbrauchen. Schon mit einer einfachen 9-V-Batterie ausgestattet, können sie unabhängig von Computern und Steckdosen herumgetragen werden und so beispielsweise Messdaten aufnehmen. Will man solche Daten dann aber an den stationären Computer oder das Internet schicken, hat man wieder das Problem, dass man eine Übertragungsmöglichkeit benötigt, die meist kabelgebunden ist.

#### LISTE DER TECHNISCHEN DATEN

<b>Chip</b>	HDG104, ATmega32UC3
<b>Verwendete Anschlüsse</b>	ICSP, Uno: 7, 10, 11, 12, 13, Mega: 7, 10, 50, 51, 52, 53
<b>Funktion</b>	802.11b/g-Netzwerk, WEP/WPA2-Personal-Verschlüsselung
<b>Anschlüsse</b>	SD-Karten, Mini-USB (Firmware-Update), Jumper (Firmware-Update setzen), FTDI-Schnittstelle zum Debugging

Das Wi-Fi-Shield kann diese Lücke schließen, denn es ermöglicht dem angeschlossenen Arduino, über WLAN zu kommunizieren. So können Daten aus dem Garten in Echtzeit per Funk übertragen werden, ohne dass das Gerät an der langen Leine angebunden ist.



Das Wi-Fi-Shield kommuniziert drahtlos über WLAN.

# INDEX

## A

Addition 58  
Alkaline-Batterie 123  
Ampere 71  
Analoge Uhr 102  
Anode 77  
Arduino  
  Elektronik 68  
  Projekte 92  
Arduino-Befehle 67  
Arduino-Entwicklungs-  
  umgebung 23  
Arduino-Software  
  Apple Mac-Installation 27  
  Download 23  
  Windows-Installation 24  
Arduino-Supportseite 32  
Arduino Uno 10  
Arduino-Websites 67  
Array 61  
ASCII 59  
ATmega16U2 13  
ATmega328 11  
Ausgang 37

## B

Basis 80  
Befehle 32  
Betriebssystem 23  
Bleifreies Lötén 91  
Bool 49  
Booleans 48  
Breadboard 81  
Bug 67  
byte 43

## C

C 22  
C# 22  
Compiler 22,31  
C-Programmierung 32

## D

Debugger 62  
Debugging 67  
delay 38  
delayMicroseconds() 44  
digitalWrite 38  
Diode 76  
double 44  
Download  
  Arduino-Software 23  
  Source Code 16

## E

Elektrischer Leiter 71  
Elektromotor 18  
Elektroniklötzzinn 83  
Emitter 80  
Entwicklungsumgebung 23  
  einrichten 29  
Erweiterungsplatinen 16  
Ethernet-Shield 17

## F

FALSCH 49  
false 49  
Farad 78  
Farbcode  
  Widerstand 73

Feld 61  
Firmware 144  
float 44  
Flussmittel 91  
for 46  
for-Schleife 47  
Funktion 36, 54  
Funktionsaufruf 54

## G

Gleitkommazahl 43

## H

Halbschritt 106  
Heileiter 95  
HIGH 38

## I

I<sup>2</sup>C 146  
ICSP-Anschluss 12  
if 48  
if-Abfrage 48  
Initialisierung 46  
Initialisierungsprozess 35  
Inkrementierung 46  
Integer 43, 46  
int-Zahl 44  
Isolierung 88

## K

Kabel 87  
Kalte Ltstelle 90  
Kaltleiter 95

Kathode 77  
Kollektor 80  
Kommandos 32  
Kommentare 33  
Kompilierung 22  
Kondensator 77  
Kupferlitze 87

## L

Lautsprecher 94  
LCD-Textdisplay 133  
LED 13  
Lilon-Akkus 124  
Linux 23  
Litze 87  
Lochrasterplatine 82  
long 43  
loop 36  
loop-Bereich 57  
Lten 82  
    Vorsichtsmanahmen 89  
Ltfett 91  
Lthilfe  
    helfende Hand 86  
Ltkolben 82  
Ltschwamm 84  
Ltstation 83  
Ltwasser 91  
Ltzinn 83  
LOW 38  
Luftdruck 68, 146  
Luftfeuchtigkeit 145  
Lufttemperatur 145

## M

Memory-Effekt 123  
Messwandler 93  
Milliampere 71  
Motor-Shield 18, 108

## N

New Line 65  
NiMH-Akkus 123  
NPN-Transistor 81  
NTC-Widerstand 95

## O

ODER 49  
Ohm 73  
Ohm'sches Gesetz 75  
Omega 73  
OS X 23  
OUTPUT 37

## P

Pin 13 13, 37  
pinMode 37  
PNP-Transistor 81  
Potenzialausgleich 69  
Potenziale 69  
Potenziometer 135  
Programm  
  hochladen 31  
  installieren 39  
  kompilieren 31  
Programmiersprache C 22

Programmierung 32  
  Befehle 32  
  Funktion 36  
  Initialisierungsprozess 35  
  Kommandos 32  
  Kommentare 33  
  Programmiersprache C 32  
Projekt  
  Analoger Temperatursensor 92  
  Analoge Uhr 102  
  LCD-Textdisplay 133  
  Wetterstation 143  
Proto-Shield 16  
Prozessor 32  
PTC-Widerstand 95  
Pulsweitenmodulation 42

## Q

Quelltext 22

## R

Rechnen  
  Ohm'sches Gesetz 75  
  Spannung 74  
  Stromstärke 74  
Reset-Knopf 12  
Reset-Schalter 13  
RoHS 91  
RS232-Schnittstelle 62  
Rückgabeparameter 54  
Rückschlagventil 76

**S**

Schleife 45  
Schrittmotor 104, 108, 115  
Schrittmotortreiber 107, 108  
Schrumpfschlauch 88  
Seitenschneider 85  
Sensor 117, 145, 146  
Serial.print() 64  
Serieller Monitor 63  
Serielle Schnittstelle 63  
Servomotor 18  
setup-Bereich 36  
Shields 16  
Silberdraht 88  
SMD-Gehäuse 13  
Source Code 22  
    Download 16  
Spannung 68  
Spannungsteiler 101  
Stahlwolle 87  
String 59  
Strom 68  
Stromkreis 72

**T**

Temperatursensor 92  
Test 46  
Texteditor 22  
Thermistor 94, 96  
Transistor 79  
true 49

**U**

Überdruck 77  
UND 49  
Unterdruck 77  
USB-Anschluss 13  
USB-Schnittstelle 62

**V**

Variable 40  
Variablentypen 43  
Verstärker 80  
void setup() 36

**W**

WAHR 49  
wartezeit 42  
Webserver 144  
Wetterstation 143  
Widerstand 72  
    Farbcode 73  
Wi-Fi-Shield 20  
Windows 23  
WLAN 21, 148

**Z**

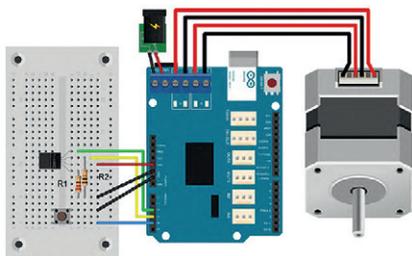
Zange 85  
zeichenkette 62



# SCHNELLEINSTIEG ARDUINO™

ALLE  
QUELLCODES  
STETHEN  
KOSTENLOS  
UNTER  
[WWW.BUCH.CD](http://WWW.BUCH.CD)  
BEREIT.

Mikrocontrollerprogrammierung war Ihnen bisher zu kompliziert, mit Elektronik haben Sie auch nichts am Hut, dann sind Sie hier richtig! Ob Internet der Dinge, Physical Computing oder einfach schnell das Garagentor per Smartphone öffnen, der Arduino™ ist die richtige Plattform, um auch Einsteigern einen schnellen Erfolg zu ermöglichen. Lernen Sie hier alles, was Sie für eigene Projekte benötigen, die am Ende funktionieren und Ihnen Spaß machen.



Wenig Theorie, dafür viel Praxis: dokumentiert mit Quellcode, Schaltbildern und Screenshots.

## Know-how für den Einstieg: Board, C-Programmierung und Elektronikwissen für die Praxis

Arduino™-IDE installieren, notwendige C-Grundlagen lernen und den ersten Sketch auf den Arduino™ laden – die ersten beiden Kapitel führen Sie in die Grundlagen der Nutzung von Arduino™ ein. Praktisch lernen Sie danach die Grundlagen der Elektronik kennen, damit Sie auch Ihre eigenen Schaltungen aufbauen können. Sollten Sie bisher noch nicht mit einem Lötkolben gearbeitet haben, bekommen Sie in Kapitel 4 die Tipps der Profis.

## Projekte zeigen die Vielfalt von Arduino™

Mit dem notwendigem Hintergrundwissen, Schaltungsaufbau und Quellcode werden fünf unterschiedliche Arduino™-Projekte Schritt für Schritt umgesetzt. Bauen Sie die Projekte auf und lernen Sie von den Profis, nutzen Sie diese Projekte direkt als Basis für Ihre eigenen Vorhaben. Vom LCD-Textdisplay bis zur eigenen Wetterstation ist für jeden etwas dabei.



Vorstellung der wichtigsten Boards und Shields.

## Aus dem Inhalt:

- Arduino™-Boards und -Shields
- Aufbau und Ablauf eines C-Programms
- Entwicklungsumgebung installieren und einrichten
- Sketch entwickeln und hochladen
- Elektronikwissen für eigene Projekte
- Lötén wie die Profis
- Analoger Temperaturwarner
- Analoge Steampunkuhr
- Rotierende LED-Blume



9 783645 604277

Besuchen Sie  
unsere Website  
[www.franzis.de](http://www.franzis.de)

FRANZIS