



mitp

Thomas
Brühlmann

4. Auflage



Arduino Praxiseinstieg

**komplett
in Farbe**

Arduino |

Neuerscheinungen, Praxistipps, Gratiskapitel,
Einblicke in den Verlagsalltag –
gibt es alles bei uns auf Instagram und Facebook



[instagram.com/mitp_verlag](https://www.instagram.com/mitp_verlag)



[facebook.com/mitp.verlag](https://www.facebook.com/mitp.verlag)



Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere fürervielfältigungen, Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Thomas Brühlmann

Arduino

Praxiseinstieg



mitp

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.

Bei der Herstellung des Werkes haben wir uns zukunftsbewusst für umweltverträgliche und wiederverwertbare Materialien entschieden.

Der Inhalt ist auf elementar chlorfreiem Papier gedruckt.

ISBN 978-3-7475-0055-2

4. Auflage 2019

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 – 079

Telefax: +49 7953 / 7189 – 082

© 2019 mitp Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Arduino is a registered Trademark of the Arduino Team (<http://arduino.cc/en/Main/FAQ>)

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Sabine Schulz

Sprachkorrektorat: Petra Heubach-Erdmann

Coverbild: Produktbild Arduino Uno Rev.3 © 2019, arduino.cc

Satz: Reemers Publishing Services GmbH, Krefeld

Druck: Medienhaus Plump GmbH, Rheinbreitbach

Inhaltsverzeichnis

1	Einleitung	13
1.1	Aufbau des Buches	14
1.2	Downloads und weiterführende Informationen	16
1.3	Weitere Quellen	16
1.4	Danksagung	17
2	Arduino-Plattform	19
2.1	Am Anfang war der König	19
2.2	Be a Maker	20
2.3	Arduino-Plattform	23
2.4	Hardware	23
2.4.1	Arduino Uno/Genuino Uno	25
2.5	Software	28
2.6	Installation der Software	29
2.6.1	Installation des USB-Treibers unter Windows 7 und Windows 10	30
2.6.2	Installation des USB-Treibers unter Windows XP	31
2.6.3	Installation des USB-Treibers unter Mac OS X	33
2.6.4	Installation des USB-Treibers unter Linux	33
2.7	Get Connected	33
2.7.1	Verbindungskabel	34
2.7.2	Verbindung und »Blink«	34
2.7.3	Projekt Blink	38
2.8	Arduino-Entwicklungsumgebung	39
2.8.1	Voreinstellungen	39
2.8.2	Aufbau Entwicklungsumgebung	41
2.8.3	Menü- und Symbolleiste	41
2.8.4	Editor	44
2.8.5	Ausgabefenster	44
2.9	Arduino-Boards	46
2.9.1	Arduino Leonardo	46
2.9.2	Arduino Due	46
2.9.3	Arduino Yun	47
2.9.4	Arduino Mega 2560	48

2.9.5	Arduino Nano	49
2.9.6	Arduino Mini	49
2.9.7	Arduino LilyPad	51
2.10	Arduino-kompatible Boards	51
3	Startschuss	53
3.1	Das Arduino-Board	54
3.1.1	Stromlaufplan	54
3.1.2	Microcontroller – Das Gehirn	56
3.1.3	Anschlussbelegung	56
3.1.4	Stromversorgung	58
3.2	Steckbrett – Experimentieren ohne Löten	59
3.2.1	Spannungsversorgung auf dem Steckbrett	64
3.3	Spannung, Strom und Herr Ohm	68
3.4	Widerstand & Co.	72
3.4.1	Widerstand	74
3.4.2	Potentiometer	74
3.4.3	Kondensator	75
3.4.4	Diode	76
3.4.5	Leuchtdiode	76
3.4.6	Transistor	78
3.4.7	Integrierte Schaltung (IC)	79
3.4.8	Relais	81
3.4.9	Schalter	81
3.4.10	Taster	82
3.5	Programmcode	82
3.5.1	Integer, Typen und Variablen	83
3.5.2	Struktur	86
3.6	Testen	88
3.6.1	Serieller Monitor	88
3.6.2	Code-Debugging	91
3.7	Projekt Blink	93
3.8	Projekt Wechselblinker	96
4	Eingänge und Ausgänge	101
4.1	Digitale Eingänge	102
4.1.1	Pin als Eingang setzen	102
4.1.2	Digitalen Eingang lesen	104
4.1.3	Digitalen Eingang entprellen	109
4.1.4	Hohe Eingangssignale	113

4.2	Digitale Ausgänge	114
4.2.1	Ausgang setzen und ausgeben	115
4.2.2	Praxis-Tipp: Status eines Ausganges lesen	116
4.3	Analoge Welt	117
4.3.1	Analoge Signale einlesen	118
4.3.2	Analoge Signale ausgeben	124
4.4	Serielle Kommunikation	128
4.4.1	Serielle Schnittstelle (RS232)	128
4.4.2	Schnittstellenerweiterung	133
4.4.3	I ² C/2-Wire (Two-Wire)	135
4.5	Drahtlose Kommunikation	148
4.5.1	433-MHz-Kommunikation	148
4.5.2	Daten übertragen mit RFM12B Transceiver	156
4.6	Projekt: Würfel	172
5	Sensoren	179
5.1	Sensoren	180
5.1.1	LDR (Fotowiderstand)	180
5.1.2	NTC/PTC	182
5.1.3	Integrierte Temperatursensoren	186
5.1.4	Pt100 und Thermoelemente	204
5.1.5	Feuchtesensoren	217
5.1.6	Kombinierte Umweltsensoren	221
5.1.7	Schaltersensoren	233
5.1.8	Abstandssensoren	234
5.1.9	Beschleunigungssensor	235
5.1.10	Kompass	238
5.1.11	Hall-Sensor	245
5.2	Projekt Kompass mit Richtungsanzeige	251
5.3	Projekt Gefrierschrankwächter	258
5.4	Kontaktloses Fieberthermometer	261
6	Aktoren	265
6.1	Relais	265
6.2	Servos	268
6.2.1	Analoge Temperaturanzeige	274
6.2.2	Servos als Motoren für Miniroboter	277
6.3	Motoren	278
6.4	Hohe Lasten schalten	285
6.5	Projekt: Roboter mit Wii-Steuerung	289

7	Anzeigen	301
7.1	Leuchtdiode (LED)	301
7.1.1	Konstantstromquelle mit Transistor	302
7.1.2	Konstantstromquelle mit Spannungsregler	303
7.1.3	Helligkeit steuern	304
7.1.4	LED als Berührungssensor	305
7.1.5	Jetzt wird es hell.	309
7.2	7-Segment-Anzeigen	310
7.2.1	Port Expander über den I ² C-Bus	314
7.3	LC-Display (LCD)	317
7.3.1	Paralleles LC-Display	317
7.3.2	Serielle LC-Displays	320
7.4	LC-Display Nokia 3310/5110	326
7.5	OLED-Display	328
7.6	LED-Matrix	333
7.7	Projekt Geschwindigkeitsmesser fürs Fahrrad	334
7.8	Projekt Schrittzähler	343
8	Datenverarbeitung	349
8.1	Daten speichern	349
8.1.1	Daten im ATmega-Controller speichern	349
8.1.2	Daten in externem EEPROM ablegen	351
8.1.3	Daten auf SD-Karte speichern	354
8.2	Daten ins Internet senden	360
8.3	Projekt: Programmierbarer Signalgeber	360
8.4	Sinusgenerator	365
8.5	Projekt: Digitales Netzteil	366
8.5.1	Hochspannungsteil	367
8.5.2	Leistungsteil	369
8.5.3	Ansteuerungsteil	370
8.5.4	Spannungsversorgungen	378
8.5.5	Aufbau des Geräts	380
8.6	Projekt Sollwerteingabe mit Drehgeber	396
8.6.1	Projekt-Erweiterung – Ausgabe als Analogwert	405
9	Erweiterungen	411
9.1	Bibliotheken	411
9.1.1	Ethernet-Bibliothek	412
9.1.2	Wire-Bibliothek	412

9.1.3	SoftwareSerial	414
9.1.4	TinyGPS-Bibliothek	416
9.1.5	NMEA	417
9.1.6	PString-Bibliothek	419
9.1.7	TextFinder-Bibliothek	419
9.1.8	Matrix-Bibliothek	419
9.1.9	LiquidCrystal-Bibliothek (LCD)	420
9.1.10	JeeLib	420
9.1.11	Stepper-Bibliothek	420
9.1.12	Wii Nunchuk	422
9.2	Hardware-Erweiterungen (Shields)	422
9.2.1	Protoshield	422
9.2.2	Protonly-Protoshield	424
9.2.3	Floweronly-Protoshield	424
9.2.4	Ethernet Shield	425
9.2.5	Datalogger und GPS Shield	427
9.2.6	Adafruit Motor Shield	427
9.2.7	DFRobot Motor Shield	427
9.2.8	Diduino-MsMot-Shield	428
9.2.9	Keypad Shield	428
9.2.10	Wave Shield	428
9.2.11	SD Card Shield	429
9.2.12	Nano Shield	429
9.2.13	Lithium Backpack	429
9.2.14	ITEAD Power Shield	430
9.2.15	RFM12B Shield	430
9.2.16	CC3000 WiFi Shield	430
9.2.17	Schraubklemmen-Shield	431
9.3	Hardware-Adapter	432
9.3.1	RFM12B-Breakout-Board	432
9.3.2	Wii-Nunchuk-Adapter	433
10	Arduino im Einsatz	435
10.1	Verbindung zum Internet	435
10.1.1	Netzwerkverbindung	436
10.1.2	Arduino als Webserver	441
10.1.3	Der Arduino als Webclient	444
10.1.4	Eingänge und Ausgänge übers Internet steuern	448
10.1.5	Wireless Ethernet (WiFi)	451

10.2	ESP-Module und Arduino	466
10.2.1	Integration in Arduino	466
10.2.2	ESP8266	470
10.2.3	ESP8266-ESP12E	471
10.2.4	Wemos D1 Mini	471
10.2.5	WiFi-Bibliothek für ESP8266	475
10.3	Heute schon getwittert?	479
10.4	Arduino mails	483
10.4.1	Mail direkt versenden	483
10.4.2	Mail via PHP-Skript versenden	485
10.5	XML einlesen	489
10.5.1	XML lesen mit TextFinder	490
10.5.2	Wetterdaten von OpenWeatherMap abfragen	495
10.6	RSS einlesen	501
10.7	You got mail	507
10.8	Umweltdaten sammeln	511
10.9	Projekt Wetterstation	519
11	Fehlersuche/Troubleshooting	541
11.1	Allgemeines Vorgehen	541
11.2	Fehler in der Schaltung	541
11.3	Fehler im Programm	542
11.4	Probleme mit der IDE	542
11.5	Hallo Arduino-Board	543
12	DIY Boards und Clones	545
12.1	Boards	545
12.1.1	Minimalschaltung Arduino	545
12.1.2	Bare Bone Breadboard Arduino	547
12.1.3	Really Bare Bone Board (RBBB)	549
12.1.4	Helvetino	550
12.1.5	Sippino	551
12.1.6	RFBoard	551
12.2	Programmieradapter (USB-Wandler)	552
12.2.1	Anschlussbelegung FTDI	554
12.3	Programmierung über ICSP	555
12.3.1	Der Arduino als Programmiergerät (Arduino ISP)	557
12.4	Arduino im Miniaturformat mit ATtiny	560
12.4.1	tinyAVR und Arduino	561

12.4.2	Boardpakete für ATtiny	562
12.4.3	(Blink-)Schaltung mit ATtiny	569
12.4.4	Programmierung des ATtiny-Microcontrollers	571
12.4.5	AVR-Programmer	576
12.4.6	Projekt Selbstbau-Programmieradapter.	579
12.4.7	Projekt ICSP-Breakout-Board fürs Steckbrett	581
12.4.8	Projekt Windlicht.	582
12.4.9	Prototypen-Board für ATtiny84.	586
13	Tools für Praktiker	589
13.1	Hardware	589
13.1.1	Steckbrett und Kabel	589
13.1.2	Lochrasterplatinen.	590
13.1.3	LötKolben und Lötzinn	592
13.1.4	Zangen	593
13.1.5	Biegelehre	593
13.1.6	Multimeter	594
13.1.7	Oszilloskop – Spannung sichtbar machen	596
13.2	Software.	599
13.2.1	Schaltungsaufbau mit Fritzing.	599
13.2.2	Eagle CAD	603
13.2.3	KiCad	604
13.2.4	Oszilloskop mit Arduino.	605
13.3	Leiterplatten herstellen.	606
13.3.1	Datenformat Gerber	607
13.3.2	Gerber-Daten aus Fritzing.	608
13.3.3	Gerber-Daten aus Eagle.	609
13.3.4	Gerber-Daten prüfen	610
13.3.5	Leiterplatten von OSH Park	611
A	Codereferenz.	613
A.1	Programmstruktur	613
A.2	Aufbau einer Funktion	614
A.3	Konventionen	615
A.4	Datentypen	618
A.5	Datentypkonvertierung.	625
A.6	Variablen & Konstanten	625
A.6.1	Variablen	625
A.6.2	Konstanten	626

A.7	Kontrollstrukturen	627
A.8	Mathematische Funktionen	630
A.9	Zufallszahlen.	633
A.10	Arithmetik und Vergleichsfunktionen	633
A.11	Funktionen	635
A.11.1	Digitale Ein- und Ausgänge	635
A.11.2	Analoge Ein- und Ausgänge	636
A.11.3	Tonausgabe	637
A.11.4	Interrupts	637
A.12	Zeitfunktionen	639
A.13	Serielle Kommunikation	640
B	Boards	645
B.1	Vergleich der Board-Varianten.	645
B.2	Anschlussbelegung Microcontroller	646
C	Bezugsquellen.	649
C.1	Bezugsquellen und Lieferanten.	649
D	Listings	651
D.1	Wii-Nunchuk-Funktionsbibliothek (Kapitel 5)	651
D.2	Mailchecker (Kapitel 8).	655
E	Migration zu Arduino 1.0.	659
F	Stücklisten.	663
	Stichwortverzeichnis	669

Einleitung

Vor Ihnen liegt die 4. Auflage von »Arduino Praxiseinstieg« und die Erfolgsgeschichte von Arduino geht weiter. Auch im Jahr 2019 gehört die Arduino-Plattform zur ersten Wahl, wenn es um einen Einstieg in die Programmierung von Microcontrollern geht.

In der Zwischenzeit sind viele Microcontroller-Boards auf den Markt gekommen und auch wieder verschwunden. Der Arduino Uno, das Standardboard für den Einsteiger, Maker und Bastler ist weiterhin das Zugpferd im Arduino-Projekt.

Das Thema »Internet of Things«, abgekürzt IoT, wird auch immer wichtiger für die Arduino-Community. Neue Boards mit WiFi und die fantastischen ESP-Module unterstützen viele Projekte beim drahtlosen Einsatz.

Dieses Praxisbuch soll ein Führer beim Einstieg in das Thema Arduino sein und den Leser dabei unterstützen, Schritt für Schritt Hard- und Software kennenzulernen, um die ersten Erfolgserlebnisse feiern zu können. Die hier zum Einsatz kommende Art des Mixes von Hard- und Software wird »Physical Computing« genannt. Physical-Computing-Projekte haben meist einen künstlerischen Hintergrund und verbinden den Menschen mit den digitalen Systemen. Die Verbindung zwischen Mensch oder Umwelt und digitalen Systemen wird auch in vielen Selbstbau- oder Bastelprojekten umgesetzt. Dabei geben Eingabelemente und Sensoren Signale an Computersysteme weiter, in unserem Fall das Arduino-Board, und diese verarbeiten die Informationen in Form einer Reaktion, beispielsweise einer Ausgabe auf ein Display oder das Ansteuern einer Lampe, eines Motors oder eines Servos.

Bestimmt sind viele Inhalte und Beispiele in diesem Buch auch an anderen Orten in ähnlicher Form zu finden. Wichtige und grundlegende Informationen müssen trotzdem erwähnt werden, damit das Verständnis für den nächsten Schritt vorhanden ist.

Darum gibt es im Buch auch viele Verweise auf Lösungen und Beispiele, die bereits realisiert wurden. Einige sagen jetzt vielleicht, dass man für eine Linkliste normalerweise kein Geld bezahlt. Aber oft ist es schwierig, das Richtige in der großen Masse von Informationen im Internet zu finden. Gute Ideen sollten auch erwähnt werden, da sie wieder neue Ideen für eigene Anwendungen generieren.

1.1 Aufbau des Buches

Am besten arbeitet man das Buch der Reihe nach durch, da die einzelnen Kapitel aufeinander aufbauen. In den ersten Kapiteln werden zunächst die wichtigsten Grundlagen der Hard- und Software des Arduino-Projekts beschrieben. Wer dieses Wissen bereits erworben hat, kann diese Kapitel natürlich überspringen.

In **Kapitel 2** wird kurz über die Entstehung des Arduino berichtet und die Idee vom Basteln, Testen und Ausprobieren erläutert. Das Ganze ist aber kurz gehalten, da das Ziel dieses Buches die praktische Arbeit ist. Im Anschluss beginnt dann der praktische Teil. Sie lernen zuerst die Hardware und die verschiedenen Arduino-Boards kennen. Begriffe werden erklärt und dann startet schon die eigentliche Installation der Software, der Entwicklungsumgebung. Nach erfolgreicher Installation und Verbindung mit der Arduino-Plattform schließt das Kapitel mit dem ersten Programm, dem Testprogramm Blink. Es folgt eine Einführung in die Oberfläche der Entwicklungsumgebung, und schon ist sie für die ersten Programme bereit.

Kapitel 3 startet mit einem Hardwareteil, in dem Sie die Arduino-Boards und den Schaltungsaufbau mit dem Steckbrett kennenlernen. Anschließend werden die Begriffe Strom, Spannung und Widerstand sowie die wichtigsten elektronischen Bauelemente erklärt. Als Nächstes werden die wichtigsten Begriffe rund um den Programmcode erklärt: Was ist eine Variable und wie ist eine Funktion aufgebaut? Nach der Einführung in die Struktur der Arduino-Programme wird das Testen und Debuggen des Programmcodes beschrieben. Hier wird aufgezeigt, wie man den seriellen Monitor zur Fehlersuche nutzen kann.

Digitale Eingänge lesen und Ausgänge schalten sind die nächsten Schritte in **Kapitel 4**. Die erste Leuchtdiode wird zum Leuchten gebracht. Im Anschluss befassen wir uns mit der analogen Welt. Es werden grundsätzliche Themen wie die Auflösung von Analogwandlern erklärt. Das erste richtige Programm liest die Sensorspannung eines Temperatursensors ein und gibt den Wert im seriellen Monitor aus. Nach dem Einlesen von analogen Werten werden analoge Signale mittels Pulsweitenmodulation ausgegeben.

Das nächste Thema in Kapitel 4 ist die serielle Kommunikation über die serielle Schnittstelle (RS232). Es werden Daten ausgegeben und eingelesen. Anschließend wird der Datentransfer über einen 2-Draht-Bus (I²C-Bus und Two-Wire-Bus) beschrieben. Daten werden von einem Master zum Slave versendet, wodurch ein Miniatur-Servo gesteuert wird. Weiter werden praktische I²C-Anwendungen wie ein serieller Sensor und eine busfähige Uhrenanwendung realisiert. Zum Schluss wird noch eine drahtlose Kommunikation mittels 433-MHz-Technologie erklärt. Ein Projekt, in dem ein elektronischer Würfel aufgebaut wird, schließt dieses Kapitel ab.

In **Kapitel 5** werden Sensoren wie ein Fotowiderstand und viele Temperatursensoren sowie weitere Sensoren zur Umwelterfassung beschrieben. Danach folgt die

Beschreibung sonstiger Sensoren. Mit einem Beschleunigungssensor wird eine kleine Wasserwaage realisiert. Als Kapitelprojekte werden ein elektronischer Kompass, ein Gefrierschrankwächter und ein kontaktloses Fieberthermometer realisiert.

Als Nächstes folgen in **Kapitel 6** die Aktoren wie Relais und Servo. Bei einer Servoanwendung wird ein Servo zu einer analogen Temperaturanzeige umgebaut. Nun lernt der Leser die verschiedenen Motoren und deren Ansteuerung kennen. Die Erläuterung der Frage, wie man hohe Lasten schaltet, schließt das Thema Aktoren ab. Als Kapitelprojekt wird ein kleiner Roboter realisiert. Der kleine Roboter kann dabei mittels einer bekannten Komponente aus dem Spielboxbereich gesteuert werden, dem Wii Nunchuk, dem Joystick für die Wii-Spielkonsole.

Kapitel 7 behandelt die verschiedenen Anzeigeelemente. Es wird erklärt, wie man diese ansteuert und wie man die Helligkeit regeln oder fest begrenzen kann. Beispiele von Ansteuermöglichkeiten für LC-Displays und LED-Matrix und Beispiele mit Nokia-Display und OLED-Display runden das Thema Anzeigen ab. Ein Geschwindigkeitsmesser fürs Fahrrad und ein Schrittzähler für Jogger schließen dieses Kapitel ab.

Kapitel 8 beschreibt die verschiedenen Arten der Datenverarbeitung. Es werden Daten in ein EEPROM und auf eine SD-Karte geschrieben. In einem praktischen Projekt wird ein programmierbarer Signalgeber zur Erzeugung analoger Signale realisiert. Dabei lernen Sie eine neue Variante der Ansteuerung von Digitalports kennen. Als Abschlussprojekt wird ein Netzteil aufgebaut, das über eine Folientastatur bedient wird.

In **Kapitel 9** werden verschiedene Softwarebibliotheken zur Erweiterung der Arduino-Funktionalität vorgestellt. Der zweite Teil dieses Kapitels behandelt die Hardwareerweiterungen, »Shields« genannt. Einige wichtige und nützliche Shields werden etwas genauer betrachtet. Zum Abschluss dieses Kapitels wird eine praktische Lösung zur Herstellung eines Wii-Adapters vorgestellt.

Kapitel 10 beschreibt verschiedene praktische Arduino-Lösungen und wie man mittels Ethernet-Erweiterung mit dem Arduino kommunizieren kann. Wir werden einen E-Mail-Checker realisieren und unsere Mailbox abfragen. Eine Anwendung wird Meldungen an Twitter senden. Zum Abschluss dieses Kapitels werden Sensordaten gesammelt und an ein Auswertungstool übertragen. Im Schlussprojekt dieses Kapitels wird eine kleine Wetterstation aufgebaut, die das aktuelle Wetter und die Wettervorhersage für den kommenden Tag anzeigt.

Fehlersuche und Troubleshooting sind die Themen in **Kapitel 11**. Es wird gezeigt, wie man die eigene Schaltung oder das neu erstellte Programm zum Laufen bringt.

Kapitel 12 beschreibt die verschiedenen Arduino-Clones und wie man sich einen minimalen Arduino auf dem eigenen Steckbrett aufbauen kann.

In **Kapitel 13** werden verschiedene Werkzeuge beschrieben, die bei den Arduino-Projekten nützlich und hilfreich sind. Neben Steckbrett, LötKolben und Zangen werden auch Messgeräte wie Multimeter und Oszilloskop erläutert. Im Teil zum Thema Softwaretools lernen Sie Programme zur Schaltplan- und Leiterplattenerstellung kennen. Abschließend wird die auf einem Arduino-Board basierende Variante eines Oszilloskops beschrieben.

Im Schlussteil und Anhang des Buches finden Sie eine Codereferenz, eine Board-Übersicht, Informationen über Bezugsquellen sowie Anpassungen, die bei der Migration von älteren Arduino-Anwendungen auf die aktuelle Version 1.x nötig sind. Das letzte Kapitel im Anhang listet alle Stücklisten der Beispiele und Projekte aus dem Buch auf.

1.2 Downloads und weiterführende Informationen

Weitere Informationen zu den Anwendungen und Beispielen im Buch sind auf der Buchwebsite erhältlich:

<http://arduino-praxis.ch>

Die Beispielskripte stehen im Downloadbereich zur Verfügung.

<http://arduino-praxis.ch/download/>

Für Anmerkungen oder Anregungen zu diesem Thema und die Kontaktaufnahme mit dem Autor stehen die Kontaktseite der Buchwebsite, E-Mail und Twitter zur Verfügung.

Die E-Mail-Adresse zum Buch lautet:

kontakt@arduino-praxis.ch

Der Twitter-Account lautet:

<http://twitter.com/arduinopraxis> oder der User @arduinopraxis.

Im Blog zum Buch werden laufend neue und interessante Projekte sowie Produktvorstellungen aus der Arduino-Welt publiziert.

1.3 Weitere Quellen

Die größte Quelle für weitere Fragen zum Arduino ist natürlich das Internet. Zu fast jedem Problem gibt es bereits realisierte Lösungen oder Ansätze.

Für Arduino-Anwender sind folgende Websites ideale und empfehlenswerte Anlaufstellen bei Problemen und Fragen.

Arduino-Website:

<http://arduino.cc/>

Arduino-Forum:

<http://www.arduino.cc/forum/>

Make:Blog:

<http://blog.makezine.com/>

Explore & Learn Adafruit:

<https://learn.adafruit.com/>

1.4 Danksagung

Mein Dank geht vor allem an meine Familie, meine Frau Aga und meine beiden Jungs Tim und Nik. Auch während der Arbeit an der 4. Auflage dieses Buches mussten sie wieder viele Stunden ohne mich auskommen. Die Skiferien haben sie abermals allein verbracht und auch viele Stunden an den Wochenenden waren für das Schreiben des Buches verplant. Aber alle haben sehr viel Verständnis gezeigt und mich dabei unterstützt. Mittlerweile lötet auch mein jüngerer Sohn Nik Prototypen-Leiterplatten sauber zusammen.

Einen Dank möchte ich auch meinem Hardwarelieferanten aussprechen. Die Firma Boxtec (<http://shop.boxtec.ch/>) hat meine Arbeit am Buch mit Musterkomponenten und Vergünstigungen beim Hardwareeinkauf unterstützt. Lieber Christoph, herzlichen Dank dafür.

Ein weiterer Dank geht an die Arduino-Community. Ohne sie wäre dieses Buch nicht entstanden, niemand hätte originelle Lösungen und Lösungsansätze realisiert, die mich zu meinem Buch inspirierten. Viele nette und konstruktive E-Mails und Kommentare zum Buch haben mir Auftrieb für die 4. Auflage geben.

Natürlich möchte ich auch einen Dank an das Arduino-Core-Team aussprechen. Die Realisierung dieses Open-Source-Projekts ist eine Bereicherung für die Hardwaregemeinde. Die Idee einer Open-Source-Plattform, offener Hardware und kostenloser Entwicklungstools ist einfach großartig.

Zum Schluss möchte ich mich wiederum bei meiner Lektorin Sabine Schulz bedanken. Sie hat mir die nötige Zeit gegeben, um diese 4. Auflage mit vielen neuen Projekten zu schreiben.



Abbildung 1.1: Nik lötet eine Prototypen-Platine.

Arduino-Plattform

2.1 Am Anfang war der König

Arduin von Ivrea war in den Jahren 1001 bis 1015 der Markgraf von Ivrea, einer italienischen Stadt in der Region Piemont. Dieser Graf Arduin, in den Jahren 1002 bis 1004 auch König von Italien, ist der indirekte Namensgeber des Arduino-Projekts. Indirekt, weil er das Projekt nicht selbst so getauft hat, dies haben Dozenten am *Interaction Design Institute Ivrea* (IDII) getan. Andere meinen, dass die Bar in der Nähe des Instituts der Namensgeber für das Arduino-Board sei.

Tatsache ist, dass die Dozenten und Studenten vor dem Problem standen, dass es zum damaligen Zeitpunkt keine einfachen und kostengünstigen Microcontroller-Systeme für Designstudenten und deren Design- und Kunstprojekte gab. Zusammen mit Elektronikern wurde daher eine erste Serie von Arduino-Boards produziert. Als Entwicklungsumgebung stand die Processing-Umgebung Pate, und daraus wurde eine Programmiersprache für den Arduino entwickelt.

Durch die Offenheit dieser Plattform – die Entwicklungsumgebung ist kostenlos verfügbar und die Hardware kann man fertig kaufen, oder man stellt sie sich selbst her – hat sich die Arduino-Community schnell vergrößert. Viele Studenten und Bastler haben Lösungen und eigene Boards realisiert und die Projektinformationen und Daten im Internet publiziert.

Die Rechte für die Marke »Arduino« gehören der Firma tinker.it. Der Rest des Arduino-Projekts ist frei verfügbar und kann kostenlos genutzt werden.

Viele findige Tüftler haben neue Board-Varianten erstellt und Firmen bieten einzelne Komponenten oder ganze Bausätze zum Verkauf an, wobei Stromlaufpläne und auch Programme für diese fertigen Produkte und Lösungen bereitgestellt und von den Bastlern weiterverwendet werden dürfen. So verhält es sich beispielsweise auch mit den Daten des Standardboards des Arduino. Auf der Website des Arduino-Projekts sind die technischen Daten des Boards ausführlich beschrieben. Über die verschiedenen Distributoren können interessierte Anwender die fertig aufgebauten und geprüften Standardboards bestellen. Praktisch begabte Anwender, die auch die technischen Kenntnisse und die nötige Infrastruktur besitzen, können die Boards anhand der CAD-Daten selbst herstellen oder sie von einem professionellen Leiterplattenhersteller produzieren lassen.

Die zuvor beschriebene Offenheit hat sich also als Erfolg herausgestellt, und der Name des Königs von Italien ist wieder in aller Munde.

Darüber hinaus bringt diese freie Verfügbarkeit Entwickler immer wieder auf neue Ideen, sodass neue Boards entstehen. Die Präsentation und Vorstellung solcher Lösungen in den verschiedenen Bastler-Foren erhöht natürlich auch die Bekanntheit des Arduino. Selbst die Medien greifen das Thema immer wieder auf. Einzelne Projekte erscheinen sogar in den täglichen Nachrichtensendungen im Fernsehen, in den News-Portalen im Internet und in vielen Zeitungen oder Zeitschriften.

Zu erwähnen ist in diesem Zusammenhang beispielsweise das Projekt Baker Tweet (<http://www.bakertweet.com>). Dieses Projekt informiert die Welt über Twitter darüber, welche frischen Leckereien soeben aus dem Ofen des Bäckers gekommen sind. Und wer kennt nicht das Projekt Botanicalls (<http://www.botanicalls.com>)?! Die Pflanze im Wohnzimmer, die mit einem intelligenten Sensor bestückt ist, meldet über Twitter, dass sie etwas Wasser benötigt. Aus einer originellen Idee entsteht plötzlich ein Produkt, das man als fertiges Gerät oder als Bausatz mit allen Komponenten zum Selbstaufbau erstehen kann.

2.2 Be a Maker

Basteln, Spaß und Lernen ist das Motto bei den Arduino-Anwendungen. Die Anwendungen sollen nämlich nicht nur von Ingenieuren realisiert werden, die die Schaltungsentwicklung an Hochschulen erlernt haben. Die Entwicklung von Anwendungen soll auf praktischem Weg durch Aufbauen und Testen erfolgen. Das ingenieurmäßige Vorgehen durch Planen und Berechnen ist hier nicht das Wesentliche. Eine Lösung sollte schnell als Prototyp aufgebaut sein, und anschließend kann man überprüfen, ob das Resultat auch der Idee entspricht und ob die Anwendung das gewünschte Verhalten aufweist. Natürlich geht dabei auch mal etwas schief, zum Beispiel brennt ein Widerstand durch oder der Servo bleibt am Anschlag stehen.

Das Basteln und Probieren sollte Spaß machen. Dabei sollte der Kreativität freier Lauf gelassen werden. Beim Aufbau der Prototypen können plötzlich wieder neue Ideen entstehen, und vielleicht verändern sich auch die im Voraus angedachten Vorgaben.

Beim Entwickeln von Elektronikprojekten kommen oftmals auch vorhandene Geräte zum Einsatz, die man im Haushalt findet. Einen automatischen Raumlüfterfrischer, den es in verschiedenen Discountergeschäften oder Drogeriemärkten zu kaufen gibt, nutzt man üblicherweise im Haus oder in der Wohnung, um die Luft zu erfrischen. Nimmt man diesen auseinander und versieht ihn mit einer intelligenteren Logik in Form eines Arduino-Boards, entsteht aus einem einfachen Gegenstand ein ganz neues Produkt. Das Resultat ist in diesem Fall ein automatischer Luftbefeuchter für das heimische Terrarium. Mit einem intelligenten Regensensor

kann die Lufterfrischerlösung sogar zu einer internetbasierten Pflanzenbewässerung erweitert werden.

Wie das Beispiel zeigt, kann man für seine Bastelprojekte mit dem Arduino viele Gegenstände aus dem Haushalt verwenden oder diese umbauen und um eine Logik erweitern.



Abbildung 2.1: Indianerfigur aus Abfallteilen einer Elektronikproduktion

Das Basteln und Entwickeln mit Arduino ist eine Sache, die Spaß macht und die Vorstellungskraft und Kreativität fördert. Dabei muss nicht alles im Voraus geplant und berechnet werden. Es soll auseinandergebaut, geforscht und gebastelt werden, damit viele spannende und auch lustige Anwendungen entstehen. Währenddessen gilt es, immer wieder verschiedene Herausforderungen und Hürden zu nehmen. Läuft das Arduino-Programm und funktioniert das Zusammenspiel mit den externen Sensoren und Aktoren, kommt der nächste Schritt, der Aufbau der Lösung in einer stabilen Form.

Elektronik-Basteln als Hobby hat sich zu einem richtigen Markt entwickelt. Viele Online-Elektronikplattformen und Community-Websites bilden die sogenann-

te Maker-Community und liefern Anleitungen, Ideen, Bausätze und Komponenten für den Selbstbau. Stellvertretend für diese Anbieter sind Adafruit Industries (<http://www.adafruit.com>) und Sparkfun (<http://www.sparkfun.com>) zu erwähnen. Beide Firmen bieten auf ihren Websites viele Tutorials und Anleitungen für Arduino-Anwender. Damit man umgehend mit den Experimenten und Bauteilen loslegen kann, liefern sie auch alle nötigen Bauteile und Werkzeuge. In den Onlineshops können sich Bastler alle Teile, vom einzelnen Widerstand über Leiterplatten und Stecker bis zum kompletten Toolkit mit Bauteilen, Zangen und Lötkolben besorgen. Weltweit finden Messen für Maker und Bastler statt, die als Maker Faire bezeichnet werden, auf denen sich Hobbyanwender und Profis treffen und austauschen. Auf der Maker-Faire-Website (<http://makerfaire.com/>) werden jeweils die kommenden Messen angekündigt. In Europa ist speziell die Maker Faire Rome (<http://www.makerfairerome.eu/>) zu erwähnen, auf der sich die Maker aus ganz Europa treffen.

Das Schöne an diesem Hobby ist, dass man in das Elektronik-Basteln nicht viel Geld investieren muss. Viele Teile hat man bereits im Haushalt oder im Hobbykeller. Zu Beginn kauft man sich meist als Grundausstattung ein Arduino-Board, installiert die kostenlose Entwicklungsumgebung, und schon kann man die ersten Tutorials durcharbeiten und originelle Lösungen aufbauen. So könnte beispielsweise eine dimmbare LED-Lampe mit einem Lampengehäuse aus einer leeren Milchflasche das heimische Wohnzimmer verschönern.

Mit den Projekten kommen die Ideen, und die dann noch fehlenden Bauteile, wie beispielsweise einen Infrarotsensor für berührungslose Lichtschalter, kauft man jeweils von Fall zu Fall.

Neben den verschiedenen Anbietern von Tutorials und den nötigen Werkzeugen und Bauteilen fördert auch eine immer größer werdende Community im Internet das Tinkering. Das Arduino-Forum (<https://forum.arduino.cc/>), die erste Adresse bei Problemen oder Fragen rund um Arduino, bietet für jeden Arduino-Bastler, egal ob Anfänger oder Profi, viele Ideen und Unterstützung. Zusätzlich stehen beim Arduino-Playground (<http://playground.arduino.cc/>) viele Anleitungen, Tutorials und Beispielprogramme für eigene Projekte bereit. Im Wiki kann jeder Anwender seine Projekte und Lösungen vorstellen und dokumentieren.

Die Do-it-yourself-Plattform Instructables (<http://www.instructables.com/>) gehört zu den größten Bastel-Portalen im Internet. Diese Selbstbauplattform wird von vielen Bastlern genutzt. Tausende von Usern präsentieren hier ihre Selbstbauprojekte. Genaue Anleitungen, meist durch Bilder oder Videos ergänzt, unterstützen den Bastler beim Nachbau des vorgestellten Projekts. Der Benutzer findet für fast jedes Problem eine Anleitung oder die Beschreibung einer bereits realisierten Lösung. Kompetente Benutzer unterstützen den Anfänger oder Einsteiger bei Fragen und Problemen.

2.3 Arduino-Plattform

Die Arduino-Plattform besteht aus zwei Teilen: einem Hardwareteil und einem Softwareteil.

Der Hardwareteil der Arduino-Plattform besteht aus einem Microcontroller-Board, das mittels Ein- und Ausgangsports die Verbindung zur physischen Welt herstellt. Die digitalen und analogen Eingänge können Daten von Sensoren wie Schaltern, Temperatursensoren oder GPS-Modulen einlesen. Die Ausgänge können Leuchtdioden, Relais, Servos oder Motoren ansteuern.

Die Entwicklungsumgebung, auch IDE (Integrated Development Environment) genannt, ist der Softwareteil. Die Entwicklungsumgebung basiert auf Processing (<http://www.processing.org>) und ist die Plattform, in der die ausführbaren Programme erstellt werden, die dann von dem Hardwareteil ausgeführt werden.

Die »Programmiersprache« ähnelt C++ und ist so ausgelegt, dass man ohne allzu tiefe Programmierkenntnisse Programme erstellen kann.

Die Software selbst wird auf einem Computer installiert und ist für verschiedene Plattformen (Windows, Mac OS und Unix/Linux) verfügbar.

Das erstellte Programm, auch »Sketch« genannt, wird via USB-Port von der Entwicklungsumgebung in den Speicher des Arduino-Boards geladen. Diese physische Verbindung zum Hardwareteil muss während der Entwicklungs- und Testphase immer vorhanden sein. Die Übertragung erfolgt seriell.

Seit Anfang 2015 befinden sich die Arduino-Gründer (Arduino LLC) und die bisherigen Board-Produzenten (Arduino S.R.L.) im Rechtsstreit um die Marke Arduino. Dies hat zur Folge, dass beide Parteien eine eigene Website für das Arduino-Projekt betreiben (arduino.cc und arduino.org). Auf beiden Websites wird die Entwicklungssoftware (IDE) mit einer unterschiedlichen Versionsnummer angeboten. Um das Projekt für die Zukunft zu sichern, hat Arduino LLC, die die Namensrechte für die USA besitzt, die Marke Genuino eingeführt. Arduino-Boards, die außerhalb der USA verkauft werden, führen nun den neuen Markennamen Genuino.

Die im Buch erwähnten Software-Versionen beziehen sich auf die Software von arduino.cc.

2.4 Hardware

Die Hardware des Arduino-Projekts ist eine Leiterplatte, die mit verschiedenen Elektronik-Bauteilen bestückt ist. In der Praxis wird diese Leiterplatte auch als »Board« bezeichnet.

Im Arduino-Projekt stehen verschiedene Boards zur Verfügung, die je nach Anforderungen (Baugröße, Anzahl digitaler Ein- und Ausgänge, Speicherbedarf) verwendet werden können.

Auf jedem Board sind die nötigen elektronischen Komponenten wie Schaltkreise, Widerstände, Kondensatoren etc. platziert. Zusätzlich besitzen die Boards verschiedene Stecker und Schnittstellen für die Verbindung mit der Außenwelt.

Die Spannungsversorgung erfolgt entweder über ein separates, externes Netzteil oder über die angeschlossene USB-Schnittstelle. Die USB-Schnittstelle kann bei 5 Volt Spannung maximal 500 mA liefern. Für kleinere eigene Entwicklungen ist diese Art der Spannungsversorgung meist ausreichend.

Über die USB-Schnittstelle erfolgt neben der soeben beschriebenen Spannungsversorgung auch die Kommunikation mit der Entwicklungsumgebung, sprich das Hochladen von Programmen (Sketche) und die Überwachung via seriellem Monitor.

Das Standardboard des Arduino (Stand Januar 2019) heißt Arduino Uno und wurde bereits im September 2010 auf der New Yorker Maker Faire (<http://makerfaire.com/newyork/2010/>) vorgestellt. Alle Erklärungen und Beispiele in diesem Buch beziehen sich auf den Arduino Uno Rev. 3, es sei denn, es wird konkret eine andere Arduino-Variante erwähnt.

Die meisten Boards können von verschiedenen Anbietern (siehe Anhang C, »Bezugsquellen«) bezogen werden. Hier sind komplette Bausätze (Kits), einzelne, bestückte und geprüfte Boards, oder auch nur Leiterplatten und Komponenten erhältlich.

Zusätzlich gibt es eine ganze Anzahl von kompatiblen Boards (Clones) für verschiedene Anwendungsfälle. Zu erwähnen wären hier das Redboard von Sparkfun (<https://www.sparkfun.com/products/13975>), der Adafruit Metro (<https://www.adafruit.com/product/50>) oder auch der Helvetino von der Boxtec-Community (<https://shop.boxtec.ch/helvetino-kit-v101-p-41829.html>).

Neben den Einzelkomponenten und den komplett aufgebauten Boards besteht für den fortgeschrittenen Elektronikbastler auch die Möglichkeit, ein Board mittels CAD-Daten selbst zu erstellen. Auf der Arduino-Website stehen hierzu die Leiterplattendaten für das kostenlose CAD-Programm Eagle zur Verfügung.

Laufend werden von innovativen Bastlern in neuen Projekten neue und originelle Arduino-Boards realisiert. Da neben den unter Open-Source-Lizenz vertriebenen Boards auch das eigentliche Gehirn des Arduino, der programmierbare ATmega-Microcontroller, frei verfügbar ist, kann man mit einem Microcontroller mit Bootloader und ein paar externen Komponenten schnell einen einfachen Arduino aufbauen. Der Bootloader ist ein kleines Programm, das im Microcontroller gespeichert ist und das Hochladen von Programmen erlaubt. Das Bootloader-Pro-

gramm wird jeweils bei Anschluss der Spannungsversorgung am Microcontroller gestartet.

Zu diesen innovativen Projekten gehört der Breadboard Based Arduino Compatible (BBAC), der für den Aufbau auf einem Steckbrett gedacht ist. Weiter ist die Arduino-Board-Version Paperduino zu erwähnen (<http://lab.guilhermemartins.net/paperduino-prints/>), die ein Stück Karton als Leiterplatte verwendet. Die Verbindung der einzelnen Stifte und Drähte der nötigen Komponenten erfolgt auf der Kartonrückseite mittels dünner Drähte.

2.4.1 Arduino Uno/Genuino Uno

<https://store.arduino.cc/arduino-uno-rev3>

Der Arduino Uno ist das aktuelle Standardboard der Arduino-Baureihe. In Abbildung 2.2 ist die aktuelle Version 3 (Revision 3) abgebildet.

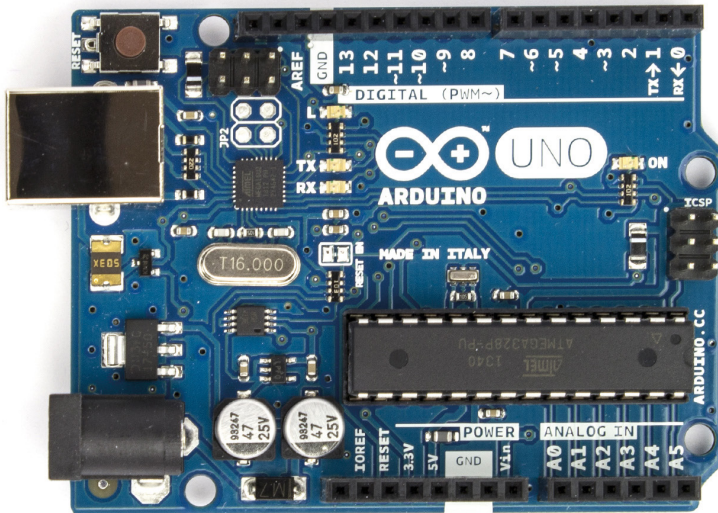


Abbildung 2.2: Arduino Uno Rev 3 (Quelle: www.arduino.cc)

Auf dem Arduino Uno sind verschiedene Stecker und Anschlussbuchsen montiert. Die Position der verschiedenen Anschlussbuchsen ist fix definiert und stellt die Kompatibilität mit früheren und zukünftigen Boards sicher.

In Abbildung 2.3 sind die Anschlussmöglichkeiten auf der Leiterplatte des Arduino Uno gekennzeichnet.

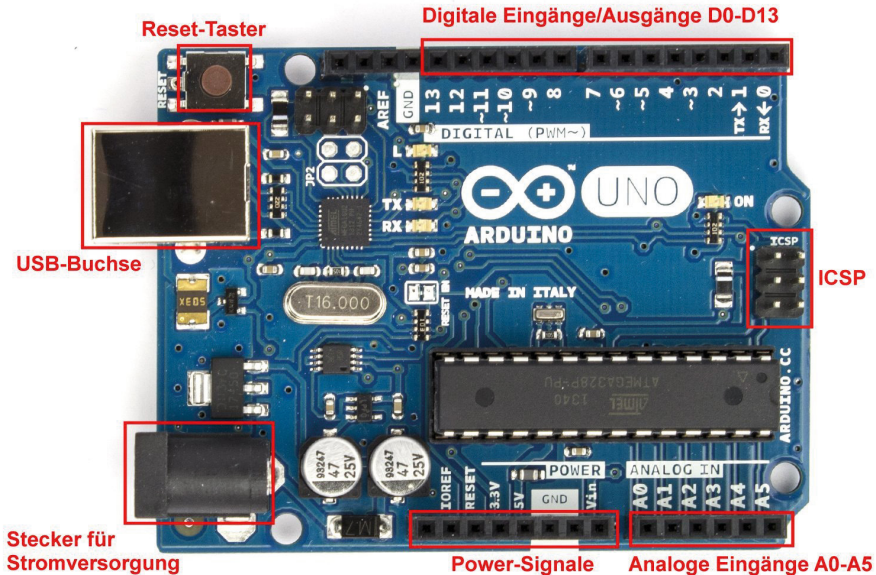


Abbildung 2.3: Arduino Uno: Anschlussmöglichkeiten

Nachfolgend werden die einzelnen Anschlussmöglichkeiten beschrieben.

USB-Buchse: Der USB-Anschluss (Typ B) wird für die Programmierung mit der Entwicklungsumgebung und zur Kommunikation mit dem angeschlossenen Rechner verwendet. Gleichzeitig kann das Board über den USB-Anschluss mit Strom versorgt werden.

Stecker für externe Stromversorgung: Über eine 2,1-mm-Buchse kann das Arduino-Board per externem Steckernetzteil oder Batterie mit Spannung versorgt werden. Die Stromversorgung über den USB-Port wird dabei automatisch deaktiviert.

Vorteilhaft ist vor allem die höhere Leistung, die aus der externen Versorgungsspannung bezogen werden kann, beispielsweise zur Versorgung von Aktoren und Sensoren. Im Falle eines Kurzschlusses wird nicht der USB-Port des angeschlossenen Rechners belastet, sondern der Überlastschutz der externen Stromversorgung.

Steckerleisten für digitale Ein- und Ausgänge (D0-D13): Einreihige Anschlussbuchsen, auf die Erweiterungsplatinen (Shields) oder Steckerleisten gesteckt werden können. Alle vorhandenen digitalen Signale und Schnittstellen sind über diese Anschlussleisten verfügbar.

Steckerleiste für analoge Eingänge (A0-A5): Einreihige Anschlussbuchsen für den Anschluss von 6 analogen Eingangssignalen.

ICSP: Die Stiftreihe für das ICSP (In-Circuit Serial Programming) erlaubt die Programmierung von Sketchen (Programmen) ohne Bootloader mittels eines externen Programmiergeräts, dazu mehr in Kapitel 10.

Neben den Anschlussmöglichkeiten sind auf der Leiterplatte des Arduino Uno Bauteile und Funktionen vorhanden, die in Abbildung 2.4 dargestellt sind.

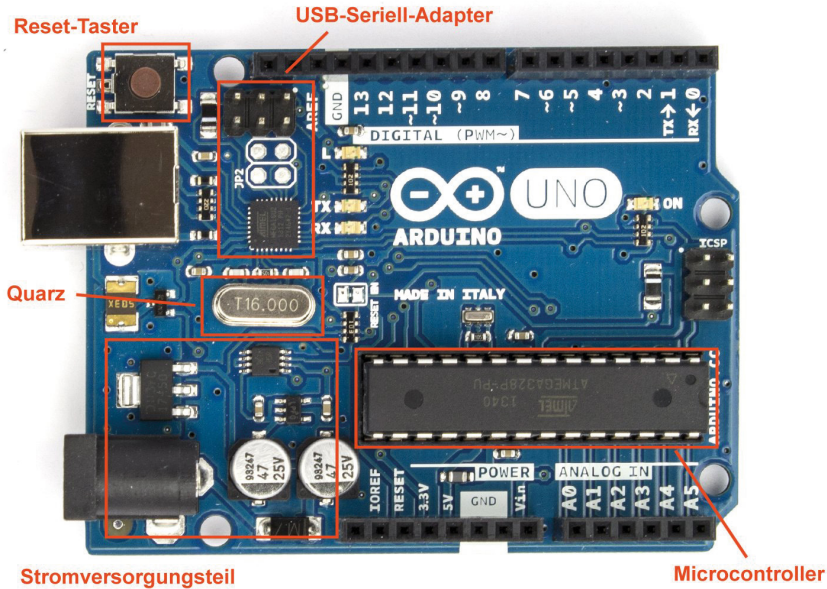


Abbildung 2.4: Arduino Uno: Bauteile und Funktionen

Reset-Taster: Mit dem Reset-Taster kann der Microcontroller zurückgesetzt werden. Nach einem Reset erfolgt ein Neustart des Boards.

USB-Seriell-Adapter: Dieser Schaltungsteil wandelt das USB-Signal in ein serielles Signal um, das der Microcontroller verarbeiten kann. Die seriellen Signale sind an den digitalen Pins D0 (RX) und D1 (TX) verfügbar.

Quarz: Der Schwingquarz ist das Bauteil, das den Takt für den Microcontroller bereitstellt.

Stromversorgungssteil: Im Stromversorgungssteil wird die Versorgungsspannung, die über den schwarzen 2,1-mm-Klinkenstecker zugeführt wird, auf 5 V geregelt. Erfolgt die Spannungsversorgung über den USB-Anschluss, so stellt dieses Schaltungsteil die saubere Umschaltung zwischen USB-Versorgung und externer Versorgung sicher.

Microcontroller: Der Microcontroller ist quasi das Gehirn des Arduino-Boards. Dieser kleine Minicomputer führt die Arduino-Programme aus und verarbeitet die Ein- und Ausgangssignale. Beim Arduino Uno wird ein Microcontroller vom Typ ATmega328 vom Hersteller Atmel eingesetzt.

In Tabelle 2.1 sind die technischen Daten des Arduino Uno aufgelistet.

Beschreibung	Detailldaten
Microcontroller	ATmega328
Spannungsversorgung	7–12 VDC
Betriebsspannung	5 VDC und 3,3 VDC (intern über Spannungsregler generiert)
Digitale Ein-/Ausgänge	14 (davon 6 als PWM-Ausgänge)
Analoge Eingänge	6
Strom pro digitalem Pin	40 mA DC
Flash Memory	32 KB (ATmega328), wobei 0,5 KB vom Bootloader belegt werden
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Taktfrequenz	16 MHz
USB-Schnittstelle	ja
Reset-Taster	ja
Onboard-ICSP-Stecker	ja
Abmessungen Board (L x B)	70 x 53 mm

Tabelle 2.1: Technische Daten Arduino Uno

2.5 Software

Für die Entwicklung der Arduino-Programme muss auf dem lokalen Rechner eine Entwicklungsumgebung installiert werden. Die Arduino-Entwicklungsumgebung ist ein Java-Programm und kann kostenlos von der Arduino-Website heruntergeladen werden. Die Software ist für die drei gängigen Betriebssysteme (Windows, Mac OS X und Linux) verfügbar und wird laufend aktualisiert und erweitert. Mit dem Download und der Installation der Software bekommt man eine komplette Entwicklungsumgebung mit Code-Editor, Dokumentation und einer Anzahl von Beispielen sowie Standardbibliotheken für verschiedene Anwendungen. Die Installation und Konfiguration dieser Entwicklungsumgebung wird im nachfolgenden Abschnitt beschrieben.

Zusätzlich zur Entwicklungsumgebung muss ein spezieller Treiber für die USB-Schnittstelle installiert und konfiguriert werden. Dieser Treiber ermöglicht die serielle Kommunikation des USB-Ports mit dem FTDI-Chip auf dem Arduino-Board. Die Software ist in der Archivdatei der Entwicklungsumgebung enthalten und muss im Anschluss an die Installation der Entwicklungsumgebung installiert werden.

2.6 Installation der Software

Im Downloadbereich der Arduino-Website (<https://www.arduino.cc/en/Main/Software>) stehen die aktuellen Versionen der Arduino-Entwicklungsumgebungen als Archivdatei zur Verfügung. Zum Zeitpunkt der Drucklegung dieses Buches (März 2019) lautete die aktuelle Programmversion 1.8.8.

Die nachfolgende Installation bezieht sich auf die Betriebssystemversionen Windows und Mac OS X. Die Installation unter Linux ist etwas komplizierter und von Distribution zu Distribution leicht unterschiedlich. Auf der Arduino-Website sind detaillierte Installationsanleitungen, Problembeschreibungen und Zusatzinformationen für die Installation auf den einzelnen Linux-Distributionen beschrieben (<http://playground.arduino.cc/Learning/Linux>).

Zum Download klickt man auf den Downloadlink der jeweiligen Betriebssystemversion und speichert die Software in einem temporären Verzeichnis auf dem Rechner ab. Die Archivdateien haben in der aktuellen Programmversion eine Dateigröße von rund von 180 MB.

Für Windows-Benutzer stehen zwei Versionen zum Download bereit, die Installer-Version und ein Paket, das ohne Administrations-Rechte auf dem Rechner installiert werden kann.

Nach erfolgreichem Download der Arduino-Umgebung kann die Software mit einem Archivprogramm entpackt werden, idealerweise in ein Programmverzeichnis.

Für Windows empfiehlt sich die Ablage der Software unter `C:\Programme` oder `C:\Program Files`. Dabei belässt man die Ordnerstruktur, wie sie im Archiv-File war. Die Anwendung ist somit im Verzeichnis `arduino-[Programmversion]` abgelegt.

Unter Mac OS X verschiebt man das ausführbare Programm nach dem Download ins Verzeichnis `/Programme` oder `/Applications`. Die ausführbare Datei heißt `Arduino.app` (Abbildung 2.5).