



Sensoren im Einsatz mit Arduino



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 für Vervielfältigungen, 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

Sensoren im Einsatz mit Arduino



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 978-3-95845-151-3

1. Auflage 2017

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2017 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.

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: © Thomas Brühlmann

Satz: III-satz, Husby, www.drei-satz.de

Inhaltsverzeichnis

	Einleitung	11
1	Arduino-Plattform	15
1.1	Das Arduino-Board	15
1.1.1	Praxisbeispiel: Arduino-Minimalschaltung	19
1.2	Entwicklungsumgebung (IDE)	20
1.3	Bibliotheken	23
1.4	Shields	26
1.4.1	Praxisbeispiel: Arduino mit Proto-Shield und Display-Shield	26
1.5	Steckbrett (Breadboard)	28
2	Warm & kalt	31
2.1	Temperatursensor NTC (Thermistor)	31
2.1.1	Praxisbeispiel: Temperaturmessung mit NTC	32
2.2	Analoger Temperatursensor LM35	35
2.2.1	Praxisbeispiel: Raumtemperaturmessung mit LM35	36
2.3	Serieller Temperatursensor DS1820	38
2.3.1	Praxisbeispiel: Temperaturmessung über seriellen Bus mit DS1820	39
2.4	Auswahl eines Temperatursensors	43
3	Licht & Farbe & Bild	45
3.1	Lichtabhängiger Widerstand (LDR)	45
3.1.1	Praxisbeispiel: Lichtmesser mit LDR	46
3.1.2	Praxisbeispiel: Nachtlampe mit LDR	47
3.2	Infrarotanwendungen mit IR-Sensor	50
3.2.1	Praxisbeispiel: Infrarot-Fernbedienung	52
3.2.2	Praxisbeispiel: LED einschalten mit IR-Fernbedienung	54
3.3	Farben erkennen mit Farb-Sensor	58
3.3.1	Praxisbeispiel: Sensor erkennt Farben	58
3.4	UV-Strahlung messen mit UV-Sensor	60
3.4.1	Praxisbeispiel: UV-Index-Monitor	63
3.5	Bilderfassung mit Kamera	68
3.5.1	Praxisbeispiel: Überwachungskamera	69

4	Distanz & Bewegung	81
4.1	Ultraschall-Sensor	81
	4.1.1 Praxisbeispiel: Abstandsmesser mit Ultraschall-Sensor	82
	4.1.2 Praxisbeispiel: Abstandsmesser für die Garage	86
4.2	Bewegungsmelder PIR-Sensor	89
	4.2.1 Praxisbeispiel: Raum-Bewegungsmelder	91
4.3	Neigung erfassen mit Tilt-Sensor	93
	4.3.1 Tilt-Sensor AT407	94
	4.3.2 Praxisbeispiel: Garagentor-Wächter	94
	4.3.3 Tilt-Sensor RPI-1031	97
	4.3.4 Praxisbeispiel: Neigungsmesser mit Neigungsrichtungs- erkennung	99
4.4	Tongebber Piezo-Sensor	102
	4.4.1 Praxisbeispiel: Piezo als Klopf-Sensor	102
5	Kräfte	107
5.1	Biegung messen mit Flex-Sensor	107
	5.1.1 Praxisbeispiel: Flex-Sensor-Testschaltung	108
	5.1.2 Praxisbeispiel: Candy-Schrank-Wächter	110
5.2	Druckempfindlicher Sensor – Force-Sensor (FSR)	114
	5.2.1 Praxisbeispiel: Druck messen mit FSR-Sensor	115
5.3	Berührungslose Eingabe	117
	5.3.1 Praxisbeispiel: Touch-Keyboards	119
	5.3.2 Praxisbeispiel: Q-Touch-Sensor	120
	5.3.3 Praxisbeispiel: Mini-Keyboard	123
6	Ort	127
6.1	Zeig mir Norden – Kompass	127
	6.1.1 Praxisbeispiel: Kompass mit HMC5883	129
	6.1.2 Praxisbeispiel: Digitaler Kompass mit LED-Anzeige	130
6.2	Position ermitteln mit GPS-Modul	136
	6.2.1 Praxisbeispiel: GPS-Daten abfragen	136
	6.2.2 Praxisbeispiel: Anzeige GPS-Position	139
7	Gase	143
7.1	Elektronische Nasen – Gas-Sensoren	143
7.2	MQ2 – Gas und Rauch	144
	7.2.1 Praxisbeispiel: Gase messen	145
7.3	MQ3 – Alkohol-Sensor	146
	7.3.1 Praxisbeispiel: Alkohol-Messgerät	147

8	Elektrische Phänomene	149
8.1	Elektrischen Strom messen	149
	8.1.1 Praxisbeispiel: Strommessung mit Shunt	149
	8.1.2 Praxisbeispiel: Strommessung mit High-Side-Messmodul	151
8.2	Messung einer elektrischen Spannung	154
	8.2.1 Praxisbeispiel: Spannungen von 0 bis 5 Volt messen	154
	8.2.2 Praxisbeispiel: Spannungen von 5 bis 30 Volt messen	156
8.3	Hall-Sensor	160
	8.3.1 Praxisbeispiel: Überwachungskontakt mit Hall-Sensor	161
9	Haus & Garten	165
9.1	Temperatur & Luftfeuchtigkeit	165
	9.1.1 Umweltsensor DHT11/22	165
	9.1.2 Praxisbeispiel: Wetterstation mit DHT-Sensor	166
	9.1.3 Umweltsensor SHT31	171
	9.1.4 Praxisbeispiel: Ansteuerung des SHT31	173
9.2	Sensoren für Bodenfeuchte-Messung	177
	9.2.1 Praxisbeispiel: Chirp-Feuchtesensor	178
9.3	Laut und leise – Geräusche erfassen	184
	9.3.1 Praxisbeispiel: Lärmmesser mit optischer Anzeige	184
9.4	Stromverbrauch messen (Gleichstrom)	189
	9.4.1 Praxisbeispiel: Einfache Strommessung mit ACS712-Sensor	190
9.5	Stromverbrauch messen (Wechselstrom)	194
	9.5.1 Praxisbeispiel: Kontaktlose Messung von Strom und Leistung mit SCT-013-Sensor	197
	9.5.2 Praxisbeispiel: Energy-Monitor-Board	201
9.6	Fensterkontakt-Sensor	202
	9.6.1 Praxisbeispiel: Fenster mit Fensterkontakt überwachen	202
10	Mensch	209
10.1	Herzschlag messen	209
	10.1.1 Praxisbeispiel: Herzschlag messen mit Infrarot-Sensor	210
10.2	Hautwiderstand messen	213
	10.2.1 Praxisbeispiel: Richtig oder falsch mit Lügendetektor	214
	10.2.2 Praxisbeispiel: Messwerte darstellen mit serielltem Plotter	216
11	Datenübertragung	219
11.1	Serielle Übertragung	219
	11.1.1 Praxisbeispiel: Serieller Monitor	220

11.2	Drahtlos mit RF-Modul (433 MHz)	221
11.2.1	Praxisbeispiel: Daten drahtlos senden mit 433-MHz-RF-Kit	221
11.2.2	Praxisbeispiel: 433-MHz-Sender	223
11.2.3	Praxisbeispiel: 433-MHz-Empfänger.	226
11.2.4	Praxisbeispiel: Drahtloser Temperatursensor LM35	230
11.3	Bluetooth	237
11.3.1	Praxisbeispiel: Bluetooth-Anwendungen mit 1Sheeld	237
11.3.2	Praxisbeispiel: 1Sheeld – Erste Anwendung	242
11.3.3	Praxisbeispiel: Fernsteuerung	243
12	Daten anzeigen & speichern	249
12.1	Elektronische Lampe – Leuchtdiode (LED)	249
12.1.1	Praxisbeispiel: Ansteuerung der Leuchtdiode	250
12.1.2	Praxisbeispiel: Mini-Lichtelement mit LED	251
12.2	Viele Farben mit RGB-LED	252
12.2.1	Praxisbeispiel: Ansteuerung einer RGB-Leuchtdiode	254
12.3	LED-Streifen mit Neopixel	256
12.3.1	Praxisbeispiel: Farbmuster mit LED-Streifen	258
12.4	Balkenanzeige mit LED	262
12.4.1	Praxisbeispiel: 10-Segment-Balkenanzeige	264
12.5	Daten und Messwerte anzeigen mit Display	267
12.5.1	Praxisbeispiel: Parallele LCD-Ansteuerung	269
12.6	Daten speichern auf SD-Karte.	272
12.6.1	Praxisbeispiel: Datenlogger mit SD-Karte.	273
12.6.2	Praxisbeispiel: Uhrzeit mit DS1307.	277
12.6.3	Praxisbeispiel: Datenlogger mit Zeitstempel	281
12.7	Datenspeicher EEPROM	285
12.7.1	Praxisbeispiel: Daten ins EEPROM schreiben	285
12.7.2	Praxisbeispiel: Daten aus dem EEPROM lesen	286
12.7.3	Praxisbeispiel: Daten aus dem EEPROM löschen	288
12.8	Internet-Plattformen – Internet of Things (IoT)	289
12.9	Sensordaten bei ThingSpeak.	290
12.9.1	Praxisbeispiel: Lichtmesswerte an ThingSpeak senden.	293
12.10	Lokale IoT-Datenbank mit MySQL-Datenbank	297
12.10.1	Praxisbeispiel: Webserver installieren.	297
12.10.2	Praxisbeispiel: MySQL-Datenbank verwalten.	299
12.10.3	Praxisbeispiel: Sensordaten-Erfassung	299
12.10.4	Praxisbeispiel: Daten erfassen mit PHP	303
12.10.5	Praxisbeispiel: Sensordaten in Webbrowser darstellen	305
12.10.6	Praxisbeispiel: Sensordaten als Liniengrafik darstellen.	307
12.10.7	Praxisbeispiel: Umweltdaten sammeln und an Webserver senden	311

13	Sensor-Shield	317
13.1	Das Sensor-Shield	317
13.2	Shield-Schaltung	318
13.3	Anschlussmöglichkeiten	320
13.4	Anschlussbelegung	323
13.5	Anwendungsmöglichkeiten	324
	13.5.1 Praxisbeispiel: Ansteuerung RGB-LED	324
	13.5.2 Praxisbeispiel: Analogwert-Monitor mit RGB	326
13.6	Bezugsquellen	332
14	Sensor-Board	333
14.1	Sensor-Board	333
	14.1.1 Praxisbeispiel: Grundaufbau Sensor-Board	336
	14.1.2 Praxisbeispiel: Programmierung des Sensor-Boards	337
14.2	Low-Power-Betrieb	341
	14.2.1 Praxisbeispiel: Stromverbrauch reduzieren auf dem Arduino-Board	341
	14.2.2 Praxisbeispiel: Sensor-Board im Low-Power-Betrieb mit Low-Power-Bibliothek	343
	14.2.3 Praxisbeispiel: Sensor-Board im Low-Power-Betrieb mit JeeLib-Bibliothek	344
	Stichwortverzeichnis	347



Einleitung

Sensoren sind die Augen und Fühler jedes Arduino-Boards und erlauben dem kleinen Elektronik-Board die Erfassung von vielen verschiedenen Messwerten rund um Mensch & Umwelt und Haus & Hof. Die von den Sensoren erfassten Messwerte können vom Arduino-Board weiterverarbeitet und dargestellt werden.

Wenn Sie dieses Buch in der Hand halten, möchten Sie vermutlich etwas tiefer in die faszinierende Welt der Sensoren eintauchen und praktische Lösungen und Projekte realisieren.

Dieses Buch über Sensor-Anwendungen mit Arduino richtet sich an Einsteiger, die bereits etwas Erfahrung mit dem Arduino-Board gesammelt haben und nach neuen Ideen und praktischen Projekten suchen.

Mit den vielen im Handel erhältlichen Sensoren, ein paar Erweiterungsplatinen und etwas Fantasie können Sie sich ein eigenes Netzwerk an Sensoren zur Erfassung Ihrer Umwelt aufbauen.

Aufbau des Buches

Der Aufbau des Buches ist so ausgelegt, dass Sie auch einzelne Kapitel überspringen können.

In **Kapitel 1** wird das Arduino-Projekt vorgestellt und anschließend das Arduino-Standardboard Arduino Uno und die Entwicklungsumgebung beschrieben. Der Anwender lädt sein erstes Programm auf den Arduino und lernt die Hard- und Software-Erweiterungen (Shields und Bibliotheken) kennen. Am Ende des Kapitels wird das praktische Breadboard (Steckbrett) beschrieben.

Kapitel 2 beschreibt Temperatursensoren. Temperatursensoren gehören zu den wichtigsten Sensoren. In diesem Kapitel werden verschiedene Typen beschrieben.

Die Themen Licht, Farbe und Bild sind Inhalt von **Kapitel 3**. Mit einem Fotowiderstand kann die Raumhelligkeit ermittelt werden. Unsichtbares Licht kann von einem Infrarot-Sensor erfasst werden. Diese Sensoren werden im Projekt Fernsteuerung eingesetzt. Farbsensoren sind komplexe Sensoren, die einzelne Farben unterscheiden können. Ein UV-Sensor ist ein nützlicher Sensor für die Erfassung der Strahlung der Sonne. Der UV-Index-Monitor zeigt dabei den aktuellen Index auf einer RGB-Leuchtdiode an. Mit einer seriellen Kamera wird eine Überwa-

chung realisiert. Die Bilder der Überwachungskamera können anschließend auf dem Smartphone betrachtet werden.

In **Kapitel 4** werden Ultraschall- und Bewegungssensoren beschrieben. Ein Abstandsmesser in der Garage kann dabei Schaden am Fahrzeug beim Einparken verhindern. Mit einem einfachen Tilt-Sensor kann man die Bewegungsrichtung ermitteln. Ein einfacher Piezo-Sensor wird als Klopfsensor verwendet.

Mit einem Flex-Sensor wird in **Kapitel 5** der Süßigkeitenschrank überwacht. Diese flexiblen Sensoren eignen sich für viele Anwendungen im Haus. Mit einem Force-Sensor wird der Druck auf eine Fläche ermittelt und mittels Berührungssensoren wird ein Mini-Keyboard realisiert.

In **Kapitel 6** werden mit einem Kompass-Sensor und einem GPS-Modul Richtung und Position erfasst. Ein elektronischer Sensor mit LED-Anzeige ist ein praktisches Tool für die nächste Wanderung.

Mit Gas-Sensoren werden in **Kapitel 7** verschiedene Stoffe in der Luft erfasst. Ein Alkoholmesser sagt Ihnen etwas über den Alkoholgehalt in Ihrem Körper.

In **Kapitel 8** werden die elektrischen Phänomene Strom und Spannung mit dem Arduino-Board erfasst und gemessen. Mit einem Hallsensor wird ein Magnetfeld erfasst. Diese stammt vom Magneten eines Überwachungskontakts.

In **Kapitel 9** wird eine kleine Wetterstation mit Umweltsensoren aufgebaut. Ein Feuchtesensor ermittelt die Trockenheit im Blumentopf. Mit Stromsensoren werden die Stromaufnahmen von Verbrauchern gemessen. Dank einer kontaktlosen Strommessung kann nun auch der Stromverbrauch des Kühlschranks überwacht werden. Zur Sicherheit im Haus dient ein Kontakt-Sensor, der am Fenster oder der Tür montiert werden kann.

Das **Kapitel 10** untersucht den Menschen und beschreibt Sensoren zur Erfassung von Herzschlag und Hautwiderstand. Ein lustiges Experiment mit einem Lügen-detektor bringt Unterhaltung in die Familie.

In **Kapitel 11** werden Sensordaten seriell über Kabel und drahtlos übermittelt. Ein drahtloser Temperatursensor kann die aktuelle Temperatur von einem entfernten Standort übermitteln. Mit einem Bluetooth-Modul wird eine drahtlose Fernsteuerung für das Fernsehgerät im Wohnzimmer realisiert.

Kapitel 12 beschreibt verschiedene Anzeige-Elemente wie LED, RGB-LED und LC-Display. Anschließend werden Möglichkeiten zur Speicherung von Sensordaten auf einer SD-Karte und EEPROM beschrieben. Webbasierte Speicherung von Daten mittels IoT-Plattform oder auf einer lokalen MySQL-Datenbank schließen dieses Kapitel ab.

Im **Kapitel 13** wird ein Sensor-Shield vorgestellt, das für viele Sensor-Anwendungen eingesetzt werden kann. Auf einer integrierten RGB-LED kann ein analoger Spannungswert als Farbmuster angezeigt werden.

In **Kapitel 14** wird ein Sensor-Board vorgestellt, das für batteriebetriebene, kompakte Sensor-Anwendungen verwendet werden kann. Die Leiterplatte des Boards ist als Open Source freigegeben und kann für eigene Zwecke verwendet werden.

Mehr Informationen

Weitere Informationen zu den Sensor-Anwendungen im Buch sind auf meiner Website erhältlich.

<http://555circuitslab.com>

<http://facebook.com/555circuitslab>

Im Downloadbereich finden Sie alle Beispielscripts, Ergänzungen und Erweiterungen. Für Anmerkungen und Anregungen können Sie mit mir per E-Mail oder Twitter Kontakt aufnehmen.

E-Mail: maker@555circuitslab.com

Twitter: <https://twitter.com/arduinopraxis>

Weitere Informationen zum Thema Arduino und laufend neue Projekte beschreibe ich in meinem Blog.

<http://arduino-praxis.ch>

Eine detaillierte Einführung finden Sie in meinem bekannten Arduino-Buch »Arduino Praxiseinstieg«.

<https://mitp.de/070>

Danksagung

Ich möchte mich ganz herzlich bei meiner Frau Aga und meinen Jungs Tim und Nik bedanken, dass sie mir wiederum die Zeit gegeben haben, dieses Buchprojekt zu realisieren. Die Familie musste wieder viele Stunden auf mich verzichten und hat erneut die Skiferien ohne mich verbracht.

Vielen Dank auch meinem Hardwarelieferanten, der Firma Boxtec (<http://shop.boxtec.ch>), die mich mit Musterbauteilen unterstützt hat und immer ein Ohr für meine Bedürfnisse an Elektronik-Komponenten hat. Vielen Dank, lieber Christoph.

Zum Schluss möchte ich mich bei meiner Lektorin Sabine Schulz vom mitp-Verlag bedanken. Es war wieder eine nette und produktive Zusammenarbeit.

Im Januar 2017

Thomas Brühlmann

Arduino-Plattform

Das Arduino-Projekt hat sich aus einem internen Universitätsprojekt zu einer weltweit verbreiteten Plattform für die Ausbildung und den Hobbybereich entwickelt. Seit dem Start des Projekts vor über zehn Jahren ist der Begriff »Arduino« quasi der Startschuss für den immer noch anwährenden Maker-Boom.

Makerspaces, Vereine und Einrichtungen für Ausbildung und Hobby bieten Kurse und Workshops an. Tausende von Bastlern haben ihre Projekte online dokumentiert oder bieten Video-Anleitungen auf bekannten Streaming-Plattformen an.

Mit Recht kann man sagen, dass das Arduino-Projekt-Team mit seinem kleinen Board bei vielen jungen Menschen das Interesse für das Thema Technik geweckt hat.

Seit dem ersten Arduino-Board sind etliche neue Modelle auf den Markt gekommen. Einige Modelle haben sich nicht durchgesetzt, andere haben sich zum Standard entwickelt.

Im Herbst 2016 haben sich die Mitglieder des Kern-Teams des Arduino-Projekts nach einem über zwei Jahre dauernden Clinch wieder versöhnt. An der World Maker Faire in New York wurde die Wiedervereinigung bekannt gegeben und bis Ende 2016 soll eine gemeinnützige Arduino-Foundation gegründet werden.

Dank der vielen Bastler, Maker und Programmierer entstehen laufend neue Projekte, Bibliotheken und Funktionen für das Arduino-Board, die anschließend von den Anwendern weltweit verwendet werden.

1.1 Das Arduino-Board

Das Arduino-Board ist eine Leiterplatte mit aufgelöteten elektronischen Bauteilen. Die zentrale Komponente oder das Gehirn des Boards ist der schwarze Baustein oder »Chip«, der Microcontroller. Der Microcontroller führt die Arduino-Programme, Sketche genannt, aus und verarbeitet die Eingangs- und Ausgangssignale. Das auf den Arduino hochgeladene Programm wird im internen Flash-Speicher gespeichert.

Das Arduino-Board wird über den USB-Anschluss oder über ein externes Netzgerät mit Spannung versorgt.

Arduino Uno

<https://www.arduino.cc/en/Main/ArduinoBoardUno>

In Abbildung 1.1 ist das Standardboard der Arduino-Baureihe, Arduino Uno genannt, zu sehen.

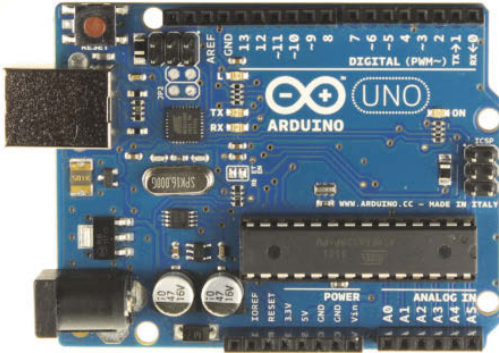


Abb. 1.1: Arduino Uno (Quelle: arduino.cc)

Das Modell Arduino Uno, Rev. 3 ist momentan das Standard-Board und wird als Basis für die Experimente und Projekte in diesem Buch eingesetzt.

Auf dem Arduino Uno sind verschiedene Stecker und Anschlussbuchsen montiert, die unterschiedliche Funktionen ausführen.

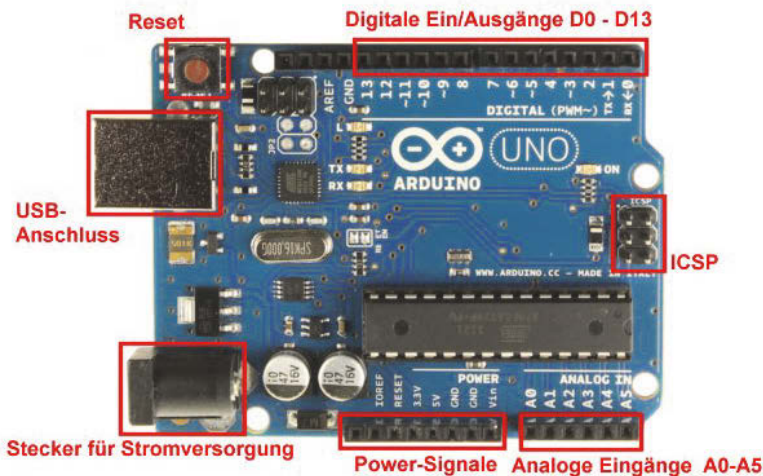


Abb. 1.2: Arduino Uno – Anschlussmöglichkeiten

Die rot markierten Anschlussmöglichkeiten aus Abbildung 1.2 haben folgende Funktionen:

■ **Stecker für Stromversorgung**

Der 2,1-mm-Anschluss, in der Praxis auch Jack-Adapter genannt, dient zum Anschluss eines externen Netzgeräts oder einer Batterie zur Stromversorgung. Erfolgt die Spannungsversorgung über diesen Anschluss, wird die Stromversorgung über den USB-Anschluss elektronisch deaktiviert.

Falls zusätzliche Energie für die Versorgung von angeschlossenen Sensoren, Motoren oder Relais benötigt wird, empfiehlt sich der Einsatz der Spannungsversorgung über diesen Stecker.

■ **USB-Anschluss**

USB-Anschluss (Typ B) für die Kommunikation des Arduino-Boards mit dem angeschlossenen Rechner. Über den USB-Anschluss wird ein Arduino-Programm auf das Arduino-Board geladen. Gleichzeitig kann über diesen Anschluss das Board mit Spannung versorgt werden.

■ **Reset-Taster**

Durch Betätigung des Reset-Tasters kann der Microcontroller auf dem Arduino-Board zurückgesetzt werden.

■ **Digitale Ein/Ausgänge D0 bis D13**

Über die obere Reihe der Buchsenleisten können die digitalen Ein- und Ausgänge D0 bis D13 angesteuert werden.

■ **ICSP**

Die 6-polige Stiftleiste mit der Bezeichnung ICSP (In-Circuit Serial Programming) wird für die Programmierung mit einem externen Programmieradapter verwendet.

■ **Analoge Eingänge A0 bis A5**

Buchsenleiste für den Anschluss von sechs analogen Eingangssignalen. Die Eingangssignale dürfen im Bereich von 0 bis 5 Volt liegen.

■ **Power-Signale**

Buchsenleiste mit den Spannungsversorgungen 3,3 V und 5 V. Über diese Buchsenleiste werden meistens die externen Sensoren und Schaltungen auf dem Steckbrett mit Spannung versorgt.

Die technischen Daten des Arduino-Boards sind in der nachfolgenden Tabelle 1.1 aufgelistet.

Bezeichnung	Detailldaten
Microcontroller	ATmega328
Spannungsversorgung	6–20 VDC (empfohlen 7–12 VDC)
Betriebsspannung	5 VDC und 3,3 VDC (intern über Spannungsregler generiert)

Tabelle 1.1: Arduino Uno – technische Daten

Bezeichnung	Detailldaten
Digitale Ein-/Ausgänge	14 (D0–D13 , davon 6 als PWM-Ausgänge)
Analoge Eingänge	6 (A0–A5), Auflösung 10 Bit
Strom pro digitalem Pin	20 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
Resetschalter	ja
Onboard-ICSP-Stecker	ja
Abmessungen Board (L x B)	70 x 53 mm

Tabelle 1.1: Arduino Uno – technische Daten (Forts.)

Arduino-Modelle

Neben dem Arduino Uno sind im Arduino-Projekt weitere Arduino-Boards verfügbar, die zusätzliche Funktionen oder andere Bauformen aufweisen. Zu erwähnen sind dabei die Modelle:

- Arduino Mega – Arduino-Board mit bis zu 54 digitalen Ein- und Ausgängen
- Arduino Zero – Arduino-Board mit 32-Bit-Microcontroller
- Arduino Mini – Arduino-Board mit ATmega328 und kleiner Bauform
- Arduino MKR1000 – kompakter Arduino Zero mit WiFi-Anschluss

Alle originalen Modelle der Arduino-Familie sind über die Arduino-Website erhältlich.

<https://www.arduino.cc/en/Main/Products>

Arduino-kompatible Boards

Auf dem Markt gibt es neben den originalen Arduino-Boards eine große Anzahl an Arduino-kompatiblen Boards. Diese Modelle basieren auf der gleichen Technik wie die originalen Boards und haben oftmals die gleichen Abmessungen wie die Originale. Diese sogenannten Arduino-Clones unterscheiden sich durch die Farbe der Leiterplatten, Anschlussstecker und die Boardbezeichnung.

Auf Online-Plattformen und in den großen asiatischen Online-Shops gibt es aber auch viele Boards, die als Originale angeboten werden, obwohl dies Kopien sind.

Die Entwicklung und der Verkauf von Arduino-kompatiblen Boards durch Maker und Bastler wird vom Arduino-Projekt unterstützt, solange diese Boards eindeutig als Arduino-Clones erkennbar sind.

1.1.1 Praxisbeispiel: Arduino-Minimalschaltung

Für einfache Anwendungen oder Testaufbauten kann sich jeder Maker sein Arduino-Board auf einem Steckbrett aufbauen. Die minimalste Schaltung für ein Arduino-Board benötigt nur etwa zehn Bauteile.

Die minimale Schaltung des Arduino zeigt der Stromlaufplan aus Abbildung 1.3. Bei dieser Minimalschaltung ist zu beachten, dass keine Spannungsregelung vorhanden ist. Die Spannungsversorgung muss von einem externen Netzteil oder von einem USB-Anschluss mit stabilen 5 V geliefert werden.

Der Programmupload über den USB-Anschluss erfordert zusätzlich einen externen FTDI-Adapter. Ein FTDI-Adapter ist ein USB/Seriell-Wandlermodul, das auf einem Arduino Uno standardmäßig integriert ist. Die Minimalschaltung kann ohne diesen Wandler nicht direkt über das USB-Kabel mit dem Rechner verbunden werden.

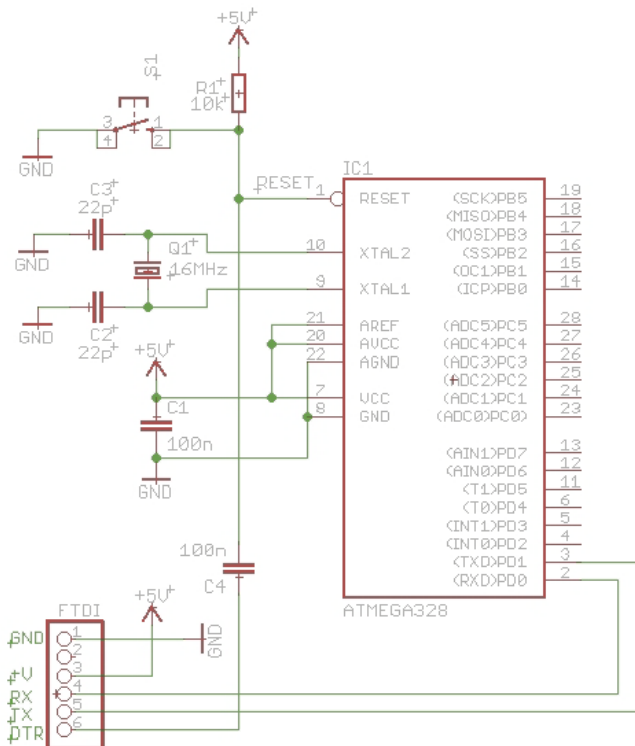


Abb. 1.3: Minimalschaltung Arduino

Stückliste (Minimalschaltung Arduino)

- 1 Microcontroller ATmega328 mit Arduino Bootloader (IC1)
- 1 Quartz 16 MHz (Q1)
- 1 Widerstand 10 kOhm (R1)
- 2 Kondensatoren 22 pF (C2, C3)
- 2 Kondensatoren 100 nF (C1, C4)
- 1 Reset-Taster (S1)
- 1 Stiftleiste 6-polig (Stecker FTDI)
- 1 Steckbrett
- Jumper-Wire

In Abbildung 1.4 ist der Aufbau auf dem Steckbrett dargestellt.

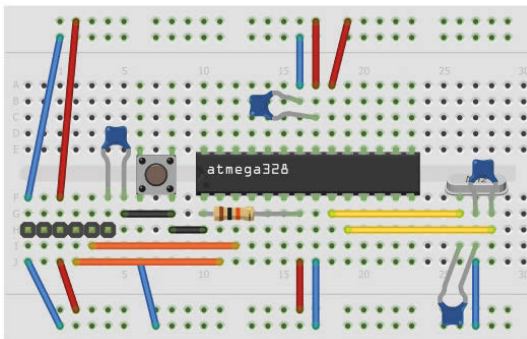


Abb. 1.4: Arduino-Minimalschaltung – Steckbrettaufbau

1.2 Entwicklungsumgebung (IDE)

Neben dem Hardware-Teil, dem Arduino-Board, gehört zum Arduino-Projekt die kostenlose Arduino-Entwicklungsumgebung. Diese Entwicklungsumgebung, in der Praxis auch als IDE bezeichnet, erlaubt das Erstellen, Testen und Hochladen von Arduino-Sketchen auf das Arduino-Board.

Die Arduino-Entwicklungsumgebung ist ein Java-Programm. Die Software ist für die drei gängigen Betriebssysteme Windows, Mac OS X und Linux verfügbar. Sie wird laufend erweitert und ist momentan in der Version 1.6.12 verfügbar.

Durch den in der Einleitung erwähnten Streit unter den Arduino-Projekt-Mitgliedern, dieser ist in der Zwischenzeit ja beigelegt, gibt es zwei Arduino-Websites, die eine Arduino-Software mit unterschiedlicher Versionsnummer anbieten. In diesem Buch wird die Version von arduino.cc beschrieben und genutzt.

Installation

Die Software für die Entwicklungsumgebung steht auf der Downloadseite der Arduino-Website zum kostenlosen Download bereit.

<https://www.arduino.cc/en/Main/Software>

Windows

Windows-Benutzer nutzen den praktischen Installer, der neben der Entwicklungsumgebung gleichzeitig den notwendigen USB-Treiber installiert.

Mac OS X

Für Mac-Benutzer wird die Entwicklungsumgebung als ZIP-Datei bereitgestellt. Nach Download und Entzippen kann die Anwendung in einen beliebigen Ordner kopiert und dann ausgeführt werden.

Linux

Linux-Anwender laden sich das passende Paket auf den Rechner und folgen den Schritten der Anleitung (<https://www.arduino.cc/en/Guide/Linux>).

Inbetriebnahme

Nach dem erfolgreichen Installieren der Entwicklungsumgebung kann sie gestartet werden. Nun wird das Arduino-Board mit einem USB-Kabel mit dem Rechner, auf dem die Entwicklungsumgebung installiert wurde, verbunden.

Für die Kommunikation zwischen dem Rechner und dem Arduino muss in der Entwicklungsumgebung das richtige Arduino-Board und der COM-Port ausgewählt werden. Die Einstellungen finden Sie unter WERKZEUGE|BOARD beziehungsweise WERKZEUGE|PORT (Abbildung 1.5).

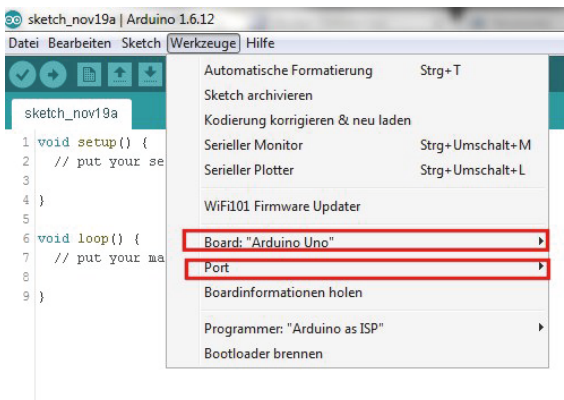


Abb. 1.5: Arduino-Entwicklungsumgebung – Auswahl Board und Port

Mit der korrekten Auswahl der beiden Optionen steht das Arduino-Board für den ersten Test bereit.

Arduino-Sketch »Blink«

Das erste Testprogramm, im Software-Umfeld ein »Hello World«-Programm, nennt sich in der Arduino-Entwicklungsumgebung *Blink* und ist unter DATEI|BEISPIELE|BASICS als Programm BLINK abgelegt.

Das Programm *Blink* ist, wie der Name sagt, ein Blink-Programm, das den Arduino-Ausgang D13 im Sekundentakt ein- und ausschaltet. Für den ersten Test muss nun Blink auf das Arduino-Board geladen werden. Dazu wird das Icon mit dem Pfeil (Hochladen) angeklickt (Abbildung 1.6).

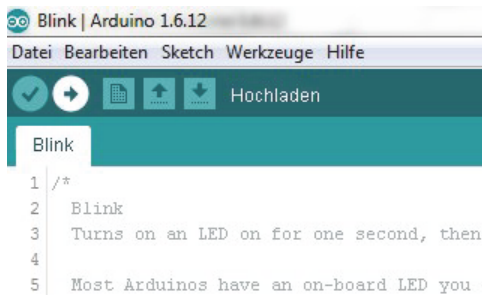


Abb. 1.6: Arduino-Entwicklungsumgebung – Upload »Blink«

Mit dem Anklicken der Upload-Funktion wird der Arduino-Sketch kompiliert und auf das Board hochgeladen.

Falls ein Fehler im Sketch erkannt wird, stoppt die Kompilierung und meldet den Fehler. Wurde ein falsches Board ausgewählt oder die Kommunikation über den COM-Port bricht ab, so wird dies auch mit einer Fehlermeldung ausgegeben.

Bei einem erfolgreichen Hochladen von Blink startet auf dem Arduino Uno das Programm. Auf dem Board wird dies durch die blinkende Leuchtdiode mit der Bezeichnung L sichtbar.

Gratulation, das erste Programm läuft!

Programmstruktur

Das Blink-Beispiel zeigt, wie der grundsätzliche Aufbau eines Arduino-Programms aussieht. Die minimalste Struktur besitzt eine Setup-Funktion `setup()` und ein Hauptprogramm `loop()`.

```
void setup() // Programmstart
{
  // Anweisungen
}

void loop() // Hauptschleife
{
  // Anweisungen
}
```

Die Setup-Funktion ist zwingend notwendig und wird beim Programmstart einmalig aufgerufen. In dieser Funktion werden Grundeinstellungen und Variablen-deklarationen vorgenommen.

Das Hauptprogramm `loop()` wird nach dem einmaligen Ausführen der Setup-Funktion nun endlos durchlaufen, bis eine Spannungsunterbrechung die Ausführung stoppt.

Diese Programmstruktur finden Sie bei jedem Arduino-Sketch.

1.3 Bibliotheken

Bibliotheken oder Libraries (in der Einzahl Library) sind in sich geschlossene Programme zur Funktionserweiterung der Arduino-Anwendungen. Die Bibliotheken stellen Funktionen zur Verfügung, die im Arduino-Programm angesprochen werden können.

Der Vorteil dieser Bibliotheken ist, dass der Anwender auf fertig entwickelte Funktionen zugreifen kann, ohne dass er die gewünschte Funktion neu programmieren muss.

Mit der Arduino-Entwicklungsumgebung werden eine ganze Anzahl Standard-Bibliotheken, wie beispielsweise die Ethernet-Library, die in Kapitel 12 benötigt wird, mitgeliefert. Bei der Installation der Bibliothek werden meist mehrere Beispiel-Programme oder Sketche mitgeliefert.

Diese Beispiele können unter `DATEI|BEISPIELE` aufgerufen werden. In Abbildung 1.7 werden die Beispiele der Ethernet-Bibliothek `Ethernet` aufgelistet.

Bibliotheken können über den Bibliotheksverwalter gesucht und aktualisiert werden (Abbildung 1.8).

Neue Bibliotheken können als ZIP-Datei über die Funktion `.ZIP-BIBLIOTHEK HINZUFÜGEN` in die Entwicklungsumgebung geladen werden (Abbildung 1.9).

Kapitel 1

Arduino-Plattform

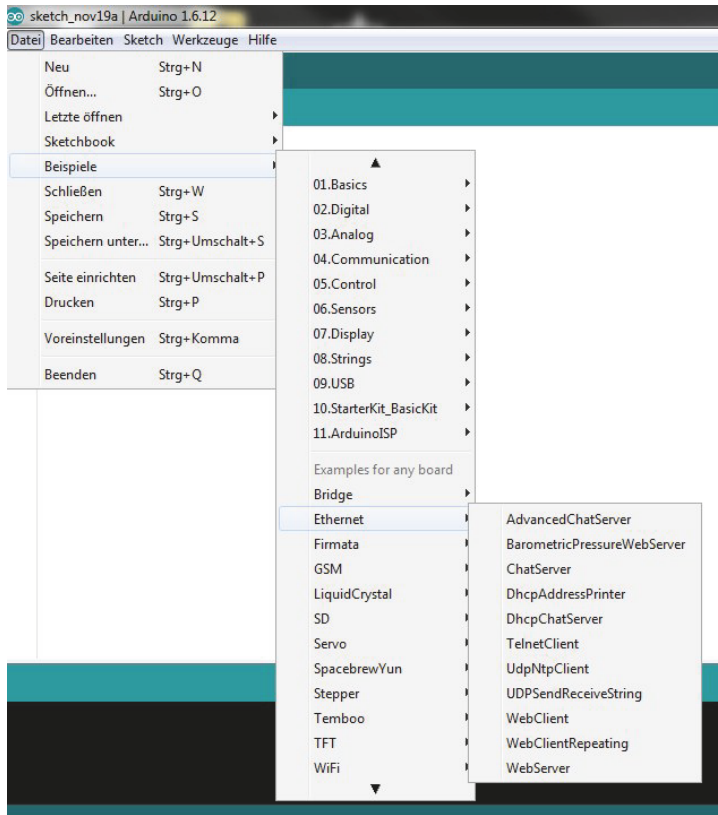


Abb. 1.7: Bibliotheken – Auflistung Beispiele

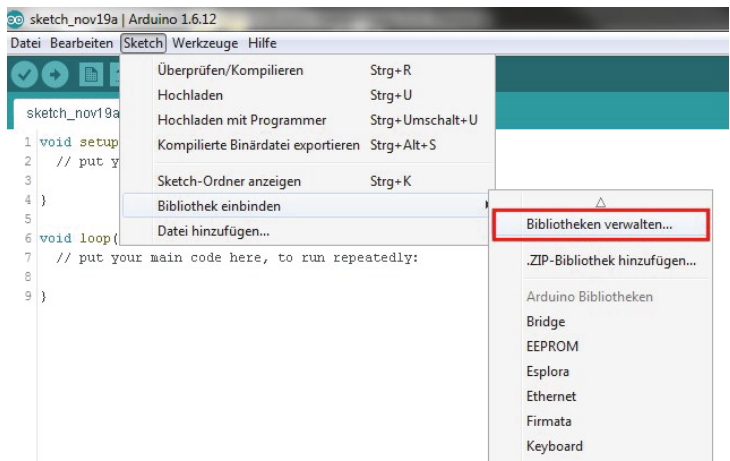


Abb. 1.8: Bibliotheken verwalten

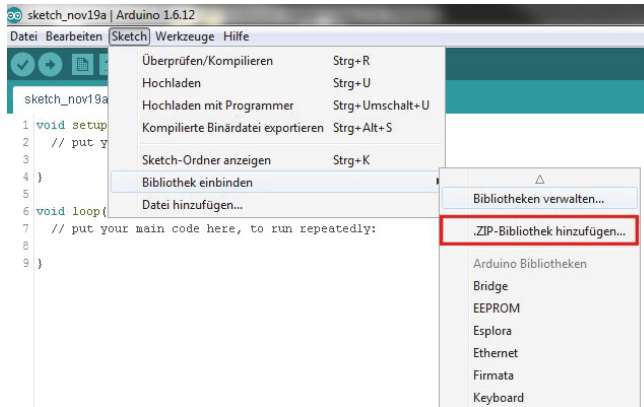


Abb. 1.9: Bibliotheken als ZIP hinzufügen

Zusätzlich zum Import über die Entwicklungsumgebung können die Arduino-Bibliotheken auch über das Dateisystem hinzugefügt werden. Nach dem Download und Entpacken der ZIP-Datei kopiert man das gesamte Verzeichnis der Bibliothek ins Bibliotheken-Verzeichnis `\libraries` der aktuellen Arduino-Entwicklungsumgebung (Abbildung 1.10).

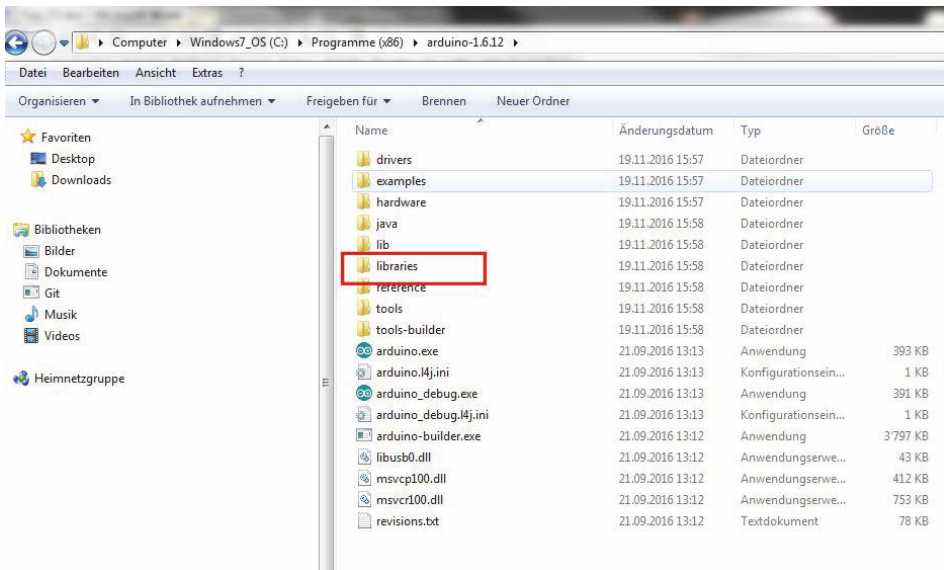


Abb. 1.10: Bibliotheken-Ordner in Arduino-Entwicklungsumgebung

Nach dem Einfügen der neuen Bibliothek muss die Entwicklungsumgebung neu gestartet werden. Beim Hinzufügen einer Bibliothek über den ZIP-Upload ist ein Neustart der Entwicklungsumgebung nicht nötig.

1.4 Shields

Erweiterungsplatinen, im Arduino-Umfeld »Shields« genannt, ermöglichen die Funktionserweiterung des Arduino-Boards durch Zusatzfunktionen und Schaltungen. Ein Shield ist eine Erweiterungsplatine, die auf das Arduino-Board aufgesteckt wird. Die Verbindung zum Arduino erfolgt dabei über Stiftleisten, die mit den Buchsenleisten auf dem Arduino-Board verbunden werden. Diese »Sandwich-Montage« erlaubt das Aufstecken mehrerer Erweiterungsplatinen, wobei natürlich beachtet werden muss, dass die einzelnen Signalleitungen nicht mehrfach genutzt werden.

1.4.1 Praxisbeispiel: Arduino mit Proto-Shield und Display-Shield

Signal D10 wird auf dem Proto-Shield für die Ansteuerung einer Leuchtdiode verwendet. Auf dem Display-Shield muss darum für die Ansteuerung des LC-Displays eine andere, freie Datenleitung verwendet werden.

In Abbildung 1.11 ist ein Proto-Shield auf ein Arduino-Board gesteckt. Für die Verbindung zum Arduino wurden in diesem Fall die sogenannten »Header-Leisten« verwendet, die die oben genannte Sandwich-Montage erlauben.

Auf das Proto-Shield kann in diesem Fall ein weiteres Shield aufgesteckt werden.

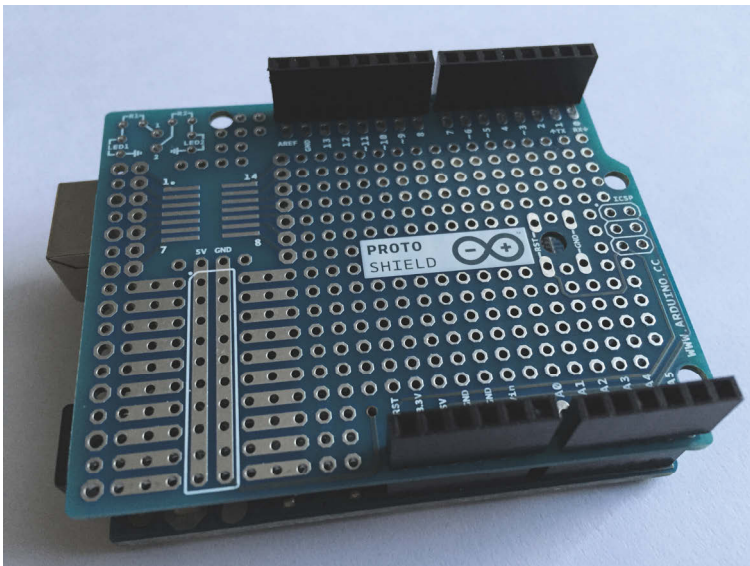


Abb. 1.11: Proto-Shield auf Arduino

Das Proto-Shield ist die einfachste Variante eines Shields und ermöglicht dem Anwender, Zusatzkomponenten und Schaltungen aufzulöten. Die Verbindungen zwischen den Bauteilen müssen selber erstellt werden. Das Proto-Shield beinhaltet also keine eigentliche Schaltung, sondern nur Platz und Lötanschlüsse für eigene Schaltungsaufbauten. Für erste Prototypen einer Funktion sind Proto-Shields die geeignete Lösung, um einen stabilen Aufbau zu realisieren. Es empfiehlt sich, immer eine Anzahl Proto-Shields am Lager zu haben.

Ich empfehle das Protonly-Protoshield PCB von Boxtec.

<http://shop.boxtec.ch/protonly-protoshield-pcb-p-41152.html>

Das Protonly-Protoshield wurde für die maximale Anzahl an Löt pads entwickelt und besitzt keine Zusatzschaltungen wie Reset-Taster oder LEDs (Abbildung 1.12).

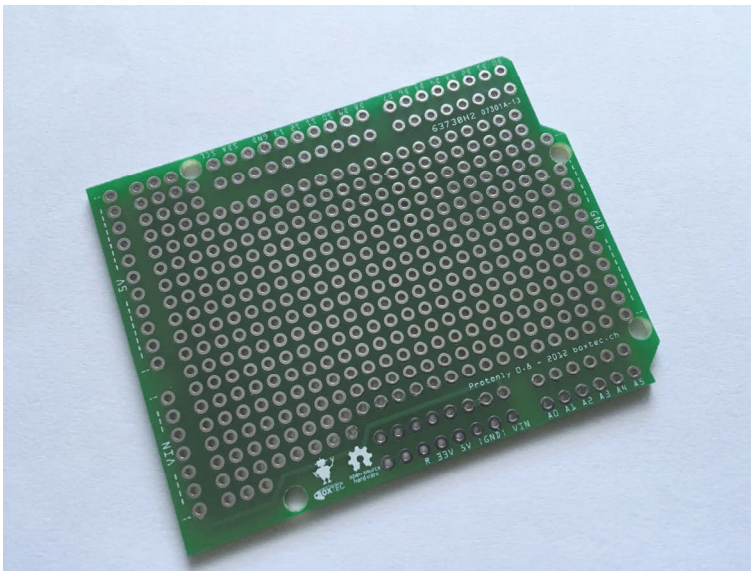


Abb. 1.12: Protonly-Protoshield

Der freie Platz auf dem Proto-Shield kann auch für die Aufnahme eines Steckbretts genutzt werden.

Abbildung 1.13 zeigt die Verwendung des Proto-Shields als flexibles Steckbrett.

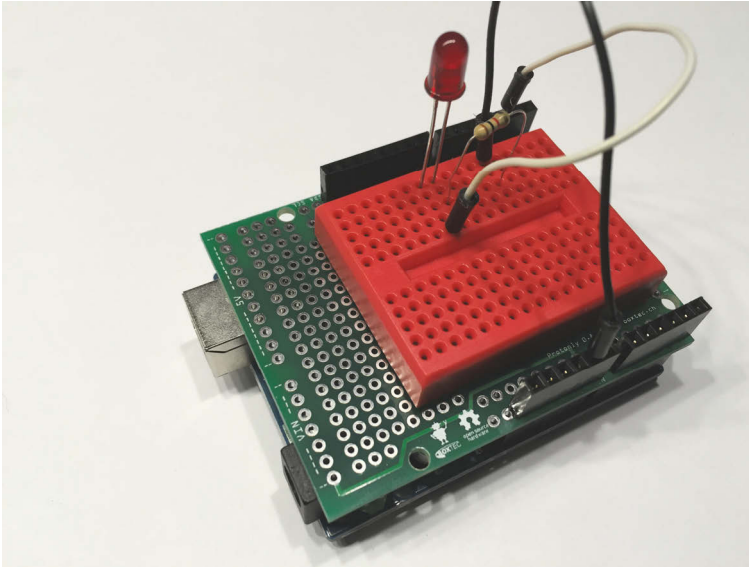


Abb. 1.13: Proto-Shield mit Steckbrett

Auf dem Markt gibt es mittlerweile viele verschiedene Shields für alle möglichen Anwendungsfälle.

In Kapitel 13 beschreibe ich ein von mir entwickeltes Sensor-Shield.

Zu den bekanntesten Standard-Shields gehören:

Ethernet-Shield

<https://www.arduino.cc/en/Main/ArduinoBoardEthernet>

Motor-Shield

<https://www.arduino.cc/en/Main/ArduinoMotorShieldR3>

Datalogger-Shield

<https://www.adafruit.com/products/1141>

1.5 Steckbrett (Breadboard)

Ein Steckbrett (engl. Breadboard) ist eine Art Leiterplatte mit Steckanschlüssen, auf das man ohne Löten handelsübliche elektronische Bauelemente aufstecken kann. Die einzelnen Steckkontakte sind dabei miteinander verbunden und erlauben einen schnellen Schaltungsaufbau ohne Lötarbeit.

Breadboards gibt es in verschiedenen Baugrößen und Farben (Abbildung 1.14).

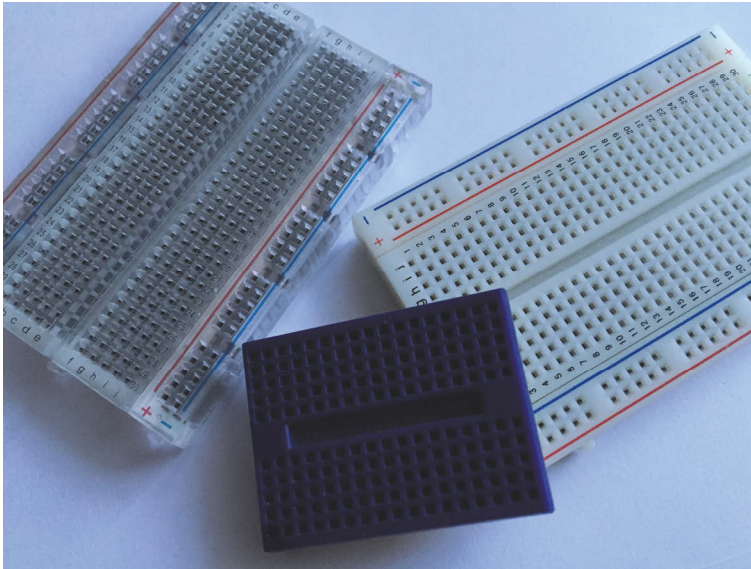


Abb. 1.14: Breadboard verschiedener Baugrößen und Farben

Bei den meisten Steckbrettern sind jeweils fünf Kontakte in vertikaler Richtung miteinander verbunden. Auf größeren Steckbrett-Varianten gibt es zusätzlich horizontale Kontakte, die für die Verteilung der Spannungsversorgung verwendet werden.

In Abbildung 1.15 sind links ein Steckbrett mittlerer Größe und rechts ein Mini-Breadboard zu sehen. Beim Steckbrett links sind oben und unten die horizontalen Kontakte für die Spannungsversorgung angeordnet. Dazwischen befinden sich die vertikalen Kontakte für die Bauteile. Das Mini-Breadboard besitzt nur vertikale Anschlusskontakte für Bauteile.

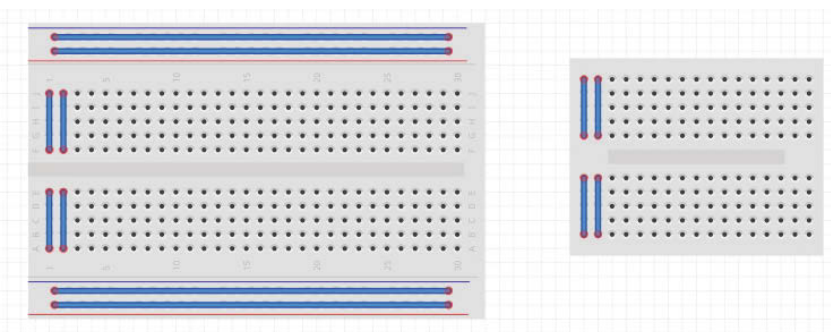


Abb. 1.15: Kontaktanordnung bei Breadboards

Hinweis

Steckbretter oder Breadboards sind für den lötfreien Einsatz ausgelegt. Das Löten von Anschlussdrähten direkt im Steckbrett ist nicht zu empfehlen, obwohl dies in vielen Internet-Videos und Tutorials gezeigt wird. Durch die Wärmeentwicklung kann der Kunststoff des Steckbretts weich werden und die Kontakte verschieben. Es besteht dann die Gefahr eines Kurzschlusses zwischen den Kontakten.