



# Sensoren im Einsatz mit Arduino

# Inhaltsverzeichnis

	<b>Einleitung</b> .....	11
<b>1</b>	<b>Arduino-Plattform</b> .....	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
<b>2</b>	<b>Warm &amp; kalt</b> .....	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
<b>3</b>	<b>Licht &amp; Farbe &amp; Bild</b> .....	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

<b>4</b>	<b>Distanz &amp; Bewegung</b> .....	<b>81</b>
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	Tongeber Piezo-Sensor .....	102
4.4.1	Praxisbeispiel: Piezo als Klopff-Sensor .....	102
<b>5</b>	<b>Kräfte</b> .....	<b>107</b>
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
<b>6</b>	<b>Ort</b> .....	<b>127</b>
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
<b>7</b>	<b>Gase</b> .....	<b>143</b>
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

<b>8</b>	<b>Elektrische Phänomene</b> .....	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
<b>9</b>	<b>Haus &amp; Garten</b> .....	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
<b>10</b>	<b>Mensch</b> .....	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 seriellm Plotter ..	216
<b>11</b>	<b>Datenübertragung</b> .....	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	<b>Daten anzeigen &amp; speichern</b> .....	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

<b>13</b>	<b>Sensor-Shield</b> . . . . .	<b>317</b>
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
<b>14</b>	<b>Sensor-Board</b> . . . . .	<b>333</b>
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
	<b>Stichwortverzeichnis</b> . . . . .	<b>347</b>



# 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 Süßigkeitenschränk 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 Lügendetektor 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](mailto: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.

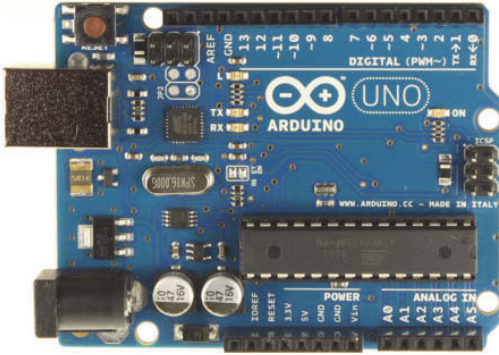


Abb. 1.1: Arduino Uno (Quelle: arduino.cc)

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.

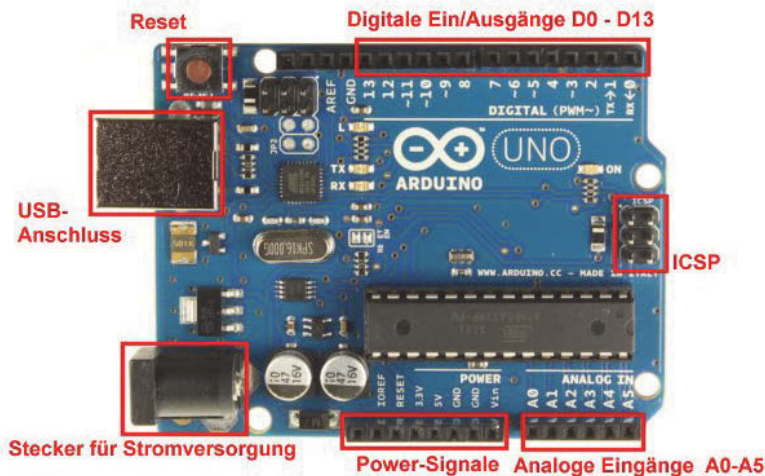


Abb. 1.2: Arduino Uno – Anschlussmöglichkeiten

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	Datendetails
Digitale Ein-/Ausgänge	14 ( <b>D0–D13</b> , davon 6 als PWM-Ausgänge)
Analoge Eingänge	6 ( <b>A0–A5</b> ), 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 Programmpupload 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.

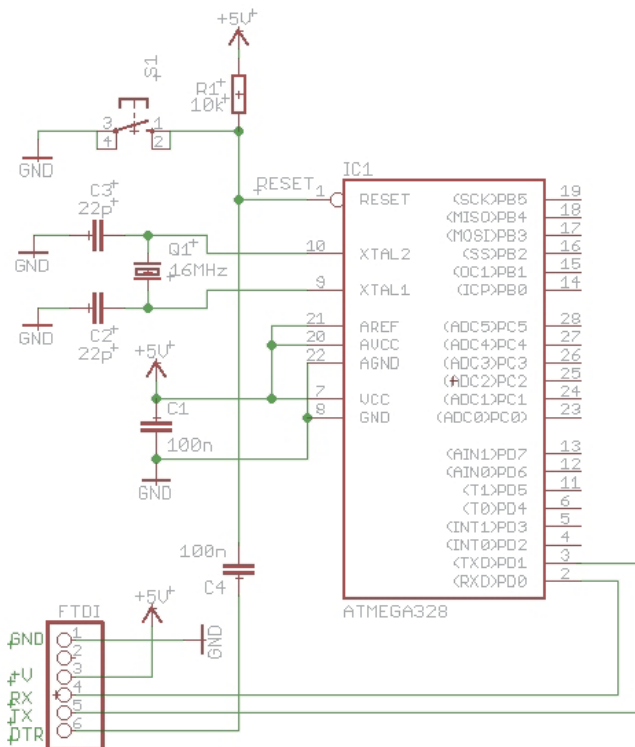


Abb. 1.3: Minimalschaltung Arduino

## 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.

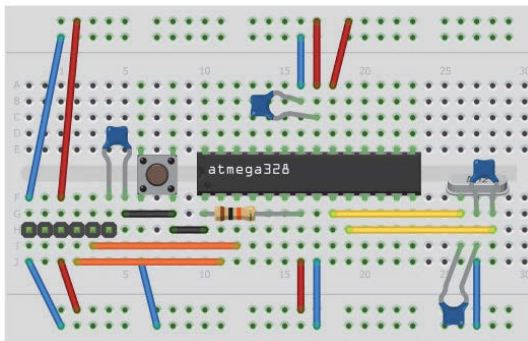


Abb. 1.4: Arduino-Minimalschaltung – Steckbrettaufbau

## 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).

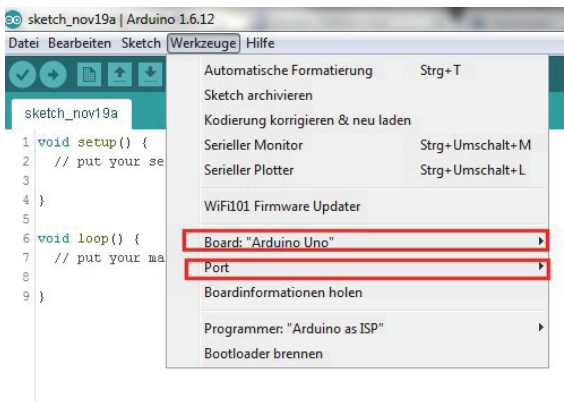


Abb. 1.5: Arduino-Entwicklungsumgebung – Auswahl Board und Port



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).

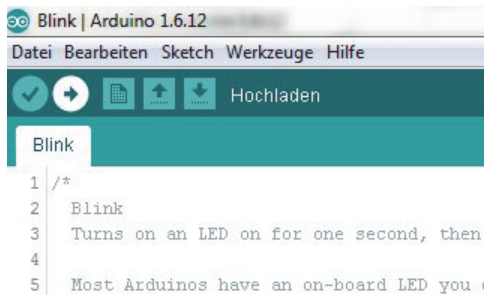


Abb. 1.6: Arduino-Entwicklungsumgebung – Upload »Blink«

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).

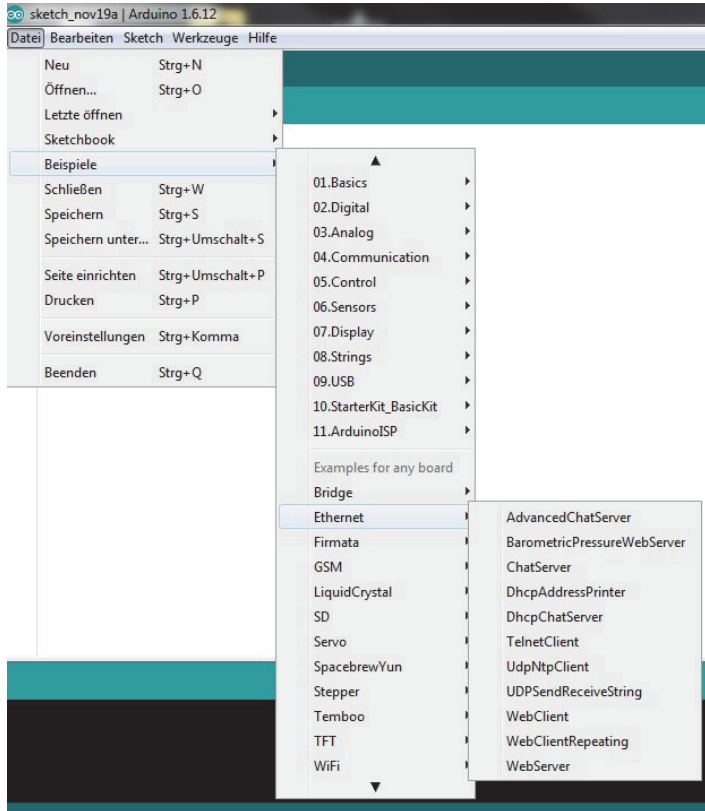


Abb. 1.7: Bibliotheken – Auflistung Beispiele

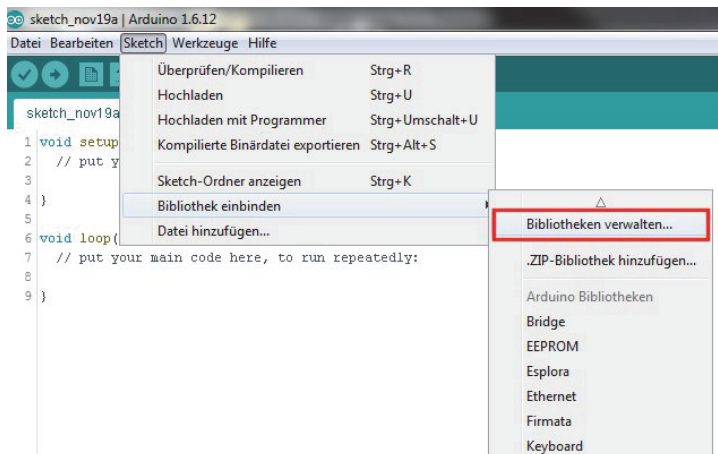
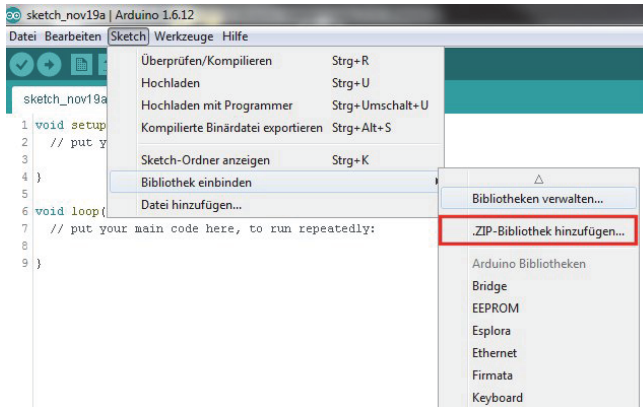
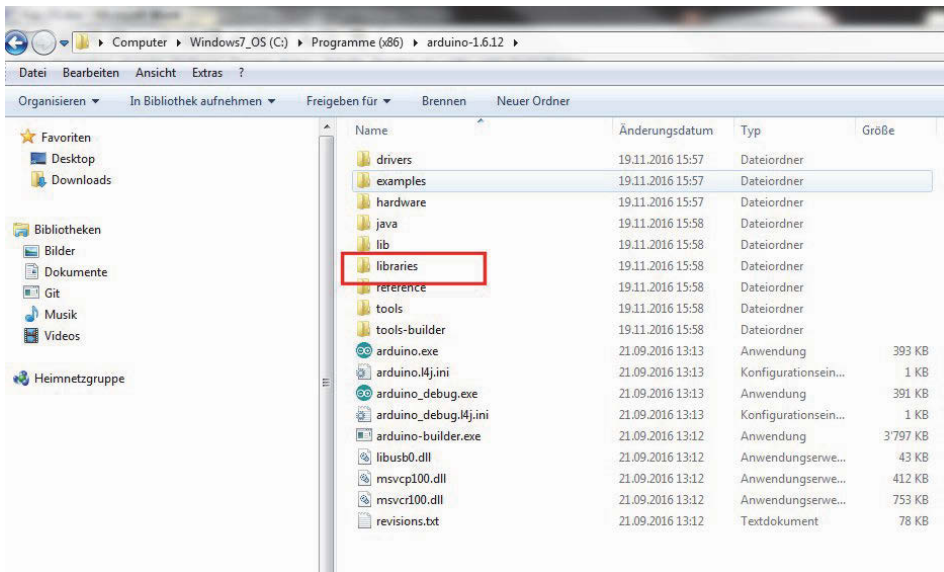


Abb. 1.8: Bibliotheken verwalten



**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.

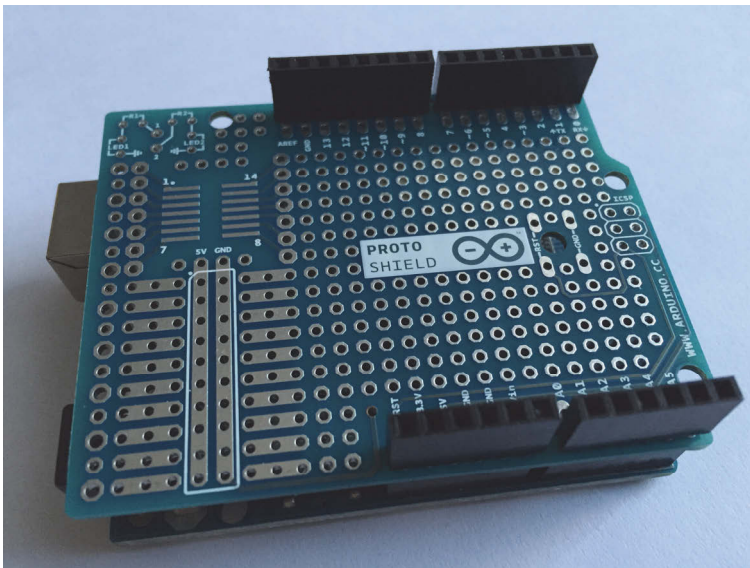


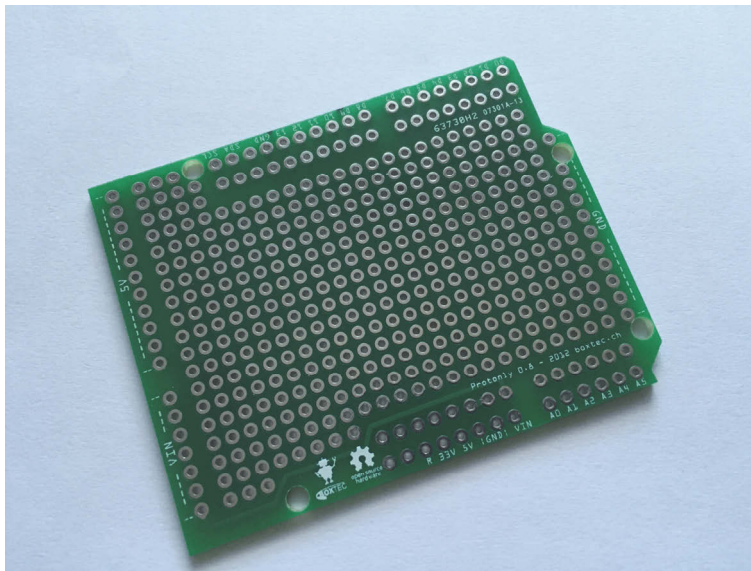
Abb. 1.11: Proto-Shield auf Arduino

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>

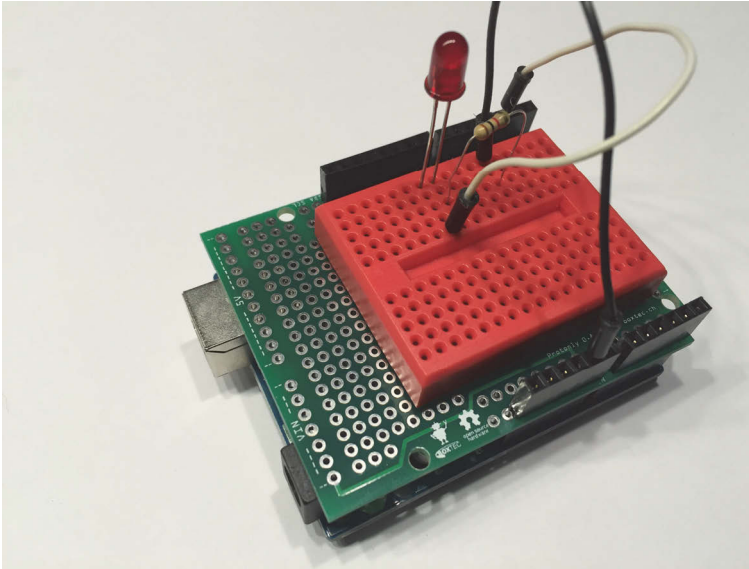
Das Protonly-Protoshield wurde für die maximale Anzahl an Löt pads entwickelt und besitzt keine Zusatzschaltungen wie Reset-Taster oder LEDs (Abbildung 1.12).



**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**

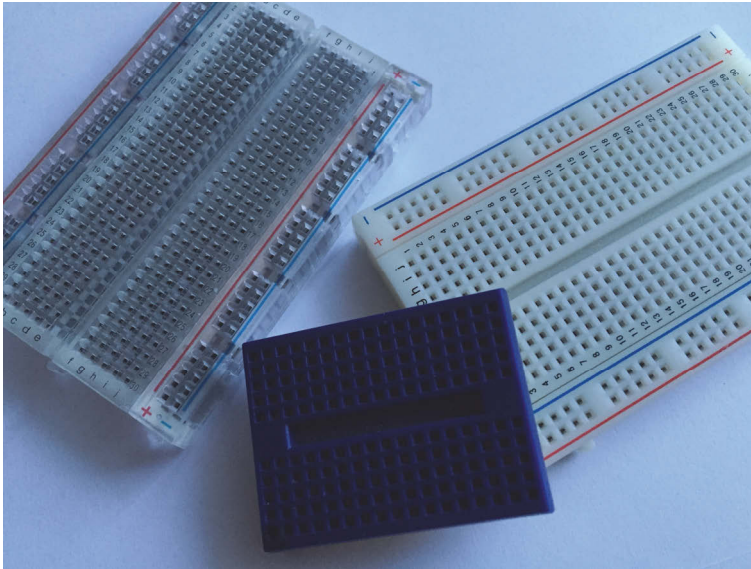
<https://www.adafruit.com/products/1141>

## **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.



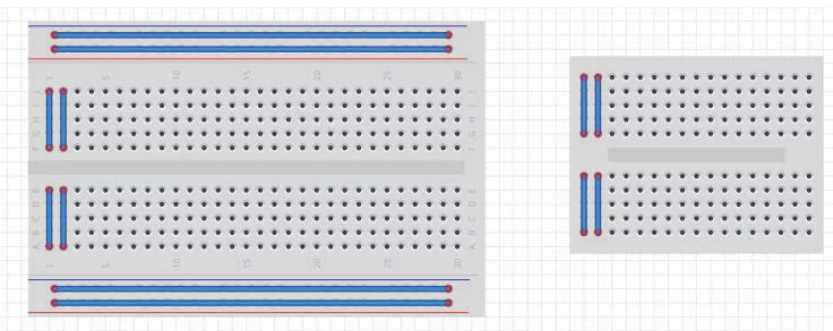
Breadboards gibt es in verschiedenen Baugrößen und Farben (Abbildung 1.14).



**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:** Kontakthanordnung 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.

# Warm & kalt

Das Erfassen von warm und kalt mittels Temperaturmessungen gehört zu den alltäglichen Aufgaben von Geräten, die wir im Haushalt oder Büro verwenden. Die Funk-Wetterstation zeigt die Außentemperatur an. Der Backofen zeigt die Backtemperatur und unsere Smartphones liefern Wetterinformationen, die von Wetterstationen gemessen wurden.

Sensoren für die Temperaturmessung sind recht günstige Bauteile und eignen sich ideal für die Erfassung der Umwelt mittels Arduino-Board.

## 2.1 Temperatursensor NTC (Thermistor)

Thermistoren sind einfache Temperatursensoren, die ihren Widerstandswert abhängig von der Temperatur verändern. Der Begriff NTC bedeutet *Negative Temperature Coefficient*, was darauf hindeutet, dass dieser Sensor einen negativen Temperaturkoeffizienten besitzt. In der Praxis heißt das, dass bei Kälte der Widerstandswert hoch ist und bei Wärme niedrig. Die Widerstandsänderung ist also genau umgekehrt zur Temperaturänderung.

Diese Widerstandstemperaturen werden darum oft einfach als NTC-Sensor oder Heißleiter bezeichnet. Neben dem NTC gibt es auch einen PTC-Thermistor. Bei PTC ist der Temperaturkoeffizient positiv. Der Widerstandswert vergrößert sich bei Wärme.

NTC-Thermistoren sind günstige Temperatursensoren und werden mit verschiedenen Widerstandswerten und in unterschiedlichen Bauformen geliefert. Der im Datenblatt angegebene Widerstandswert ist immer der Wert bei 25 Grad Celsius. Die NTC-Sensoren eignen sich für Temperaturmessungen im Bereich von –50 bis 200 Grad Celsius.

In Abbildung 2.1 sind verschiedene Bauformen von NTC-Sensoren zu sehen.

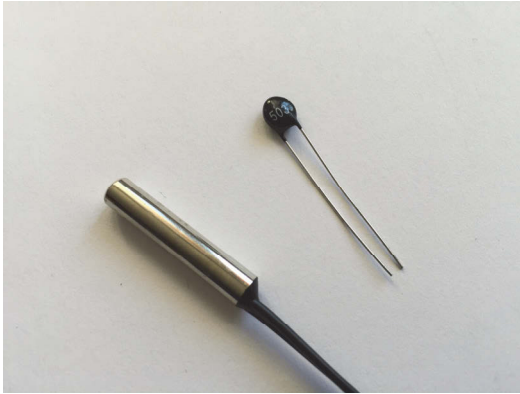


Abb. 2.1: NTC-Thermistoren

### 2.1.1 Praxisbeispiel: Temperaturmessung mit NTC

Für die Temperaturmessung mit einem NTC wird nur noch ein zusätzlicher Widerstand benötigt. In Abbildung 2.2 ist der Schaltungsaufbau auf dem Steckbrett dargestellt.

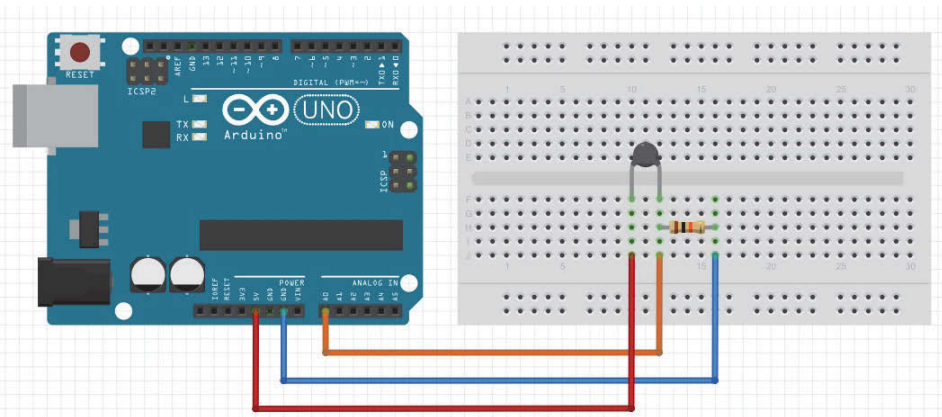


Abb. 2.2: Temperaturmessung mit NTC – Steckbrett-Aufbau

Der NTC wird zusammen mit dem Widerstand in einer Spannungsteiler-Schaltung eingesetzt. Die am Widerstand gemessene, analoge Spannung wird über die orange Signalleitung auf den analogen Eingang A0 des Arduino geführt.

#### Stückliste (NTC-Thermistor)

- 1 Arduino-Board
- 1 Steckbrett

- 1 NTC 10 kOhm
- 1 Widerstand 10 kOhm
- Jumper-Wire

Aus dem Analogwert an A0 kann nun mittels einer komplexen Berechnungsformel die aktuelle Temperatur ermittelt werden. Die Berechnung wird als sogenannte »Steinhart-Hart-Gleichung« bezeichnet.

Für die Berechnung wird die Mathematik-Bibliothek `math.h` benötigt und ins Programm eingebunden.

```
#include <math.h>
```

Die Berechnung benötigt den Widerstandswert des NTC bei 25 Grad C.

```
// Widerstandswert bei 25 Grad C  
int Wid25Grad= 10000;
```

Nun werden die Variablen für den analogen Eingang und die Temperaturmessung deklariert. Der analoge Messwert wird an A0 eingelesen.

```
int valTemp;  
// analoger Eingang  
int TempPin = 0;  
// Wert Temp in Celsius  
float tempC;
```

Im `setup()` wird die serielle Schnittstelle für die Ausgabe vorbereitet.

```
void setup()  
{  
  // Start serielle Ausgabe  
  Serial.begin(115200);  
}
```

Das Hauptprogramm liest laufend den analogen Wert an A0 ein, speichert ihn in der Variablen `valTemp` und übergibt den gemessenen Wert anschließend an die Funktion `ThermistorC(valTemp)`.

```
void loop()  
{  
  valTemp = analogRead(TempPin);  
  tempC = ThermistorC(valTemp);  
  Serial.print("Temperatur: ");
```

```
Serial.print(tempC);  
Serial.println(" C");  
// Warten 1 Sekunde  
delay(1000);  
}
```

Die Umrechnungsfunktion `ThermistorC(valTemp)` rechnet den Wert in eine absolute Temperatur um und gibt diese in Grad Celsius zurück.

Zum Test wird nun der Temperaturwert seriell ausgegeben.

Listing 2.1 zeigt das komplette Programm für die Temperaturmessung mit einem NTC.

```
#include <math.h>  
  
// Widerstandswert bei 25 Grad C  
int Wid25Grad= 10000;  
// analoger Messwert  
int valTemp;  
// analoger Eingang  
int TempPin = 0;  
// Wert Temp in Celsius  
float tempC;  
  
void setup()  
{  
    // Start serielle Ausgabe  
    Serial.begin(115200);  
}  
  
void loop()  
{  
    valTemp = analogRead(TempPin);  
    tempC = ThermistorC(valTemp);  
    Serial.print("Temperatur: ");  
    Serial.print(tempC);  
    Serial.println(" C");  
    // Warten 1 Sekunde  
    delay(1000);  
}  
  
double ThermistorC(int RawADC)  
{  
    double Temp;
```

```
Temp = log(((10240000/RawADC) - Wid25Grad));  
Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp *  
Temp * Temp));  
// Umrechnung Kelvin / Grad Celsius  
Temp = Temp - 273.15;  
return Temp;  
}
```

**Listing 2.1:** Temperaturmessung mit NTC

## Anwendungsbeispiele

- Einfache Raumtemperaturmessung
- Temperaturmessung im Terrarium
- Aquariumwächter

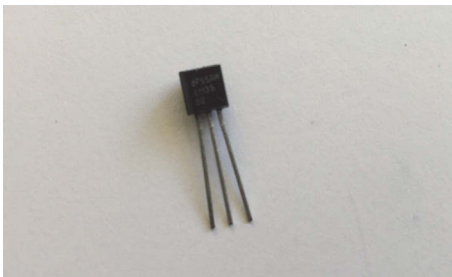
## 2.2 Analoger Temperatursensor LM35

Der Temperatursensor LM35 ist ein kompakter Sensor und liefert als Ausgangssignal eine lineare Ausgangsspannung von 10 Millivolt (mV) pro Grad Celsius. Bei 25 Grad Celsius liefert der Sensor somit eine Spannung von 250 Millivolt, die direkt über einen analogen Eingang des Arduino eingelesen werden kann.

Für Temperaturmessungen im Bereich von 0 bis 100 Grad sind keine zusätzlichen Bauteile nötig. Der Sensor wird mit einer Spannung im Bereich von 4 bis 30 Volt versorgt. An seinem analogen Ausgang kann direkt die gemessene Spannung ausgelesen werden.

Der analoge Temperatursensor LM35 kann somit auch für Anwendungen ohne Microcontroller eingesetzt werden, da er keine Linearisierung mittels Berechnungsformel oder Elektronik-Schaltung benötigt.

In Abbildung 2.3 ist ein LM35 mit der Anschlussbelegung im Gehäuse TO-92 dargestellt.



**Abb. 2.3:** Temperatursensor LM35

# Stichwortverzeichnis

## Numerisch

- 1Sheeld
  - erste Anwendung 242
  - Keypad Shield 238
- 1Sheeld (Bibliothek) 240, 245
- 1-Wire-Protokoll 38
- 433-MHz-Modul 222
  - Empfänger 222
  - Sender 222
- 4-Bit-Modus 269

## A

- A3144 (Hall-Sensor) 160
- Abstandsmelder 81
- ACS712 (Breakout-Board) 153
- ACS712 (Strom-Messmodul) 189
- Adafruit-VC0706-Bibliothek 75
- Alkohol 146
- Alkoholmessung 147
- AM23xx 168
- Amperemeter 149
- Analoger Eingang 17
- Analogwert-Monitor 326
- Anode 249
- Apache 297
- Arduino Mega siehe Arduino-Board
- Arduino Mini siehe Arduino-Board
- Arduino MKR1000 siehe Arduino-Board
- Arduino Uno siehe Arduino-Board
- Arduino Zero siehe Arduino-Board
- Arduino-Board 15
- Arduino-Clone 18
- Arduino-Foundation 15
- Arduino-Minimalschaltung 19
- Arduino-Programm 15
- Arduino-Projekt 15
- AT407 (Tilt-Sensor) 94
- AT42QT1070 (Touch-Sensor) 117
- ATmega328
  - Portregister ansteuern 89

- Aufheizphase 144
- AVRISP mkII (Programmieradapter) 339

## B

- Balkenanzeige 262
- Batteriebetrieb 341
- Bedienungselement 117
- Berührung 117
- Berührungssensor 107
- Betrieb
  - parasitärer 39
- Betriebsspannung 342
- Bewegung 81
- Bewegungsänderung 93
- Bewegungsmelder 81, 89, 91
- Bewegungsrichtung erfassen 97
- Bibliothek 23
  - 1Sheeld 240, 245
  - Dallas 39
  - EEPROM 285
  - Emonlib 198
  - HMC5883L 130
  - IRremote 52
  - JeeLib 342
  - LiquidCrystal 268, 270
  - Low-Power 342
  - Mathematik 33
  - MotionDetect 75
  - Neopixel 260
  - One-Wire 39
  - Q-Touch 121
  - SD-Card 274
  - SHT-Sensoren 174
  - SoftwareSerial 76
  - ThingSpeak 294
  - VirtualWire 223, 227
- Bibliotheksverwalter 23
- Bild 45
- Bitmuster 88
- Blink 22
- Bluetooth 219, 237

Bluetooth 4.0 237  
 Bodenfeuchte 177  
 Breadboard 28  
 Breitengrad 139  
 Buchsenleiste 17, 26  
 Bus  
     serieller 38  
 Buzzer 123

## C

Candy-Schrank-Wächter siehe Flex-Sensor  
 Chirp-Feuchtesensor 178  
     Register 180  
 Cloud 289  
 Common Anode 252  
 Common Cathode 252  
 COM-Port 21

## D

Dallas-Library 39  
 Dämmerungsschalter 46  
 Datalogger-Shield 284  
 Datenkommunikation 222, 223  
 Datenlogger 274  
 DHT11 (Umweltsensor) 165  
     technische Daten 165  
 DHT22 (Umweltsensor) 165  
     technische Daten 165  
 DHT-Sensor 165  
 Digitaler Ein/Ausgang 37  
 Digitaler Kompass 130  
 Dimmer 46  
 Display 267  
 Distanz 81  
 Distanzmessung 83  
 Drahtlose Übermittlung 219  
 Drahtlose Übertragung 221  
 Drahtloser Temperatursensor 235  
 DS1307 (Uhrenbaustein) 277  
     Lithium-Zelle 278  
 DS1820 (serieller Temperatursensor) 38

## E

Echosignal 82  
 EEPROM 18, 285  
     ATmega328 285  
 EEPROM (Bibliothek) 285  
 Elektrische Kapazität messen 178  
 Elektrizität 149  
 Emonlib (Bibliothek) 198

Energie-Messadapter 200  
 Energie-Monitor  
     Schaltung 197  
 Energy-Monitor-Board 201  
 Entwicklungsumgebung 20  
     Installation 21  
     Linux 21  
     Mac OS X 21  
     Windows 21  
 Erweiterung Platine siehe Shield  
 Ethernet-Shield 293

## F

Farbe 45  
 Farb-Sensor 58  
 Fensterkontakt 202  
     Magnet 202  
 Fensterkontakt-Sensor 202  
 Fernbedienung 50, 52  
 Fernsteuerung 243  
 Fingersensor 215  
 FlashAir 70  
 Flex-Sensor 107  
     Widerstandsänderung 110  
 Flüssigkristallanzeige siehe LC-Display  
 Force-Sensor 114  
 Fotowiderstand 45  
     lichtabhängiger Widerstand 45  
 FSR siehe Force-Sensor  
 FTDI-Adapter 19  
 Funk-Wetterstation 31

## G

Gas 143  
 Gasleck 144  
 Gas-Sensor 143  
     Übersicht 143  
 Geräusch 184  
 Gesetz  
     ohmsches 149  
 GPS 136  
 GPS-Modul 136  
 Grove 319  
 Grove-Q-Touch-Sensor 119

## H

Haartrockner 200  
 Hall-Sensor 149, 160  
     A3144 160



Hautwiderstand 213  
 Ohmmeter 214  
 HC-SR04 (Ultraschall-Sensor) 81  
 Messbereich 82  
 HD44780 (Display-Controller) 268  
 Header-Leiste 26  
 Herzschlag 209  
 Herzschlagsensor 209  
 High-Side siehe Strommessung  
 High-Side-Strommessung 152  
 Hintergrundbeleuchtung 269  
 HMC5883L 127  
 HMC5883L (Bibliothek) 130  
 HMC5883L (Kompassmodul) 129

## I

ICSP 17  
 IDE siehe Entwicklungsumgebung  
 IFTTT (Online-Plattform) 290  
 INA169 (Strommessung) 152  
 verdrahten 153  
 INA219 (Strommessung) 152  
 In-Circuit Serial Programming 17  
 Infrarot-Sensor 50  
 Internet of Things 289  
 IR-Empfänger 51  
 IRremote (Bibliothek) 52  
 IR-Sensor 50

## J

Jack-Adapter 17  
 JeeLib (Bibliothek) 342

## K

Kamera 68  
 Kapazität  
 elektrische Kapazität messen 178  
 Kapazitive Veränderung 117  
 Kathode 249  
 Key-Nummer  
 abfragen 124  
 Klinkenstecker 201  
 Klopfsignal 102  
 Kompass 127  
 digitaler 130  
 Kompass-Shield 131  
 Kontaktlose Strommessung 195  
 Krokodilklemme 215

## L

Längengrad 139  
 Lärmmesser 185  
 Schaltschwellen 186  
 LC-Display 267  
 parallele Ansteuerung 268  
 serielle Ansteuerung 268  
 LDR siehe Fotowiderstand  
 LED 249  
 Grundschialtung 250  
 Schaltzeichen 250  
 LED-Streifen 256  
 Leistungswiderstand 149  
 Leiterplatte 15  
 Leuchtdiode 249  
 Library 23  
 Licht 45  
 Lichtänderung 46  
 LiquidCrystal (Bibliothek) 268, 270  
 LM35 (analoger Temperatursensor) 35  
 LM35 (drahtloser Temperatursensor) 230  
 Grundschialtung 231  
 Logfile 276  
 loop() 22  
 Loudness-Sensor 185  
 Low Power 341  
 Low-Power (Bibliothek) 342  
 Luft 143  
 Luftfeuchtigkeit 165  
 Lügendetektor 214

## M

Magnetstreifen 161  
 Mathematik-Bibliothek 33  
 Mensch 209  
 Microcontroller 15  
 Mini-Keyboard 123  
 ML8511 (UV-Sensor) 61  
 Monitor  
 serieller 219  
 MotionDetect (Bibliothek) 75  
 MQ2 (Gas-Sensor) 144  
 MQ3 (Alkohol-Sensor) 146  
 MySQL-Datenbank 297  
 Tabellen erstellen 300

## N

Nachtlampe 48  
 Nagelsensor 178

Negative Temperature Coefficient 31  
 Neigungsmesser 99  
 Neigungssensor 93  
 Neopixel 256  
     adressierbare RGB-LEDs 256  
     Schutzbeschaltung 259  
 Neopixel (Bibliothek) 260  
 NMEA 137  
 Nordrichtung 130  
 NTC 31  
 NTC-Sensor siehe NTC

## O

Ohmsches Gesetz 149  
 One-Wire-Bibliothek 39  
 Operating Mode 240  
 Ort 127

## P

Parasitärer Betrieb 39  
 Pegelwandler 74  
 PHP 297  
 phpmyadmin 299  
 Piezoeffekt 102  
 Piezo-Sensor 102  
 Ping (Ultraschall-Sensor) 85  
 PIR-Sensor 89  
     Wärmeänderung 89  
 Plotter  
     serieller 216  
 PortD siehe Portmanipulation  
 Portmanipulation 87  
 Position 127  
 Programmstruktur 22  
 Protonly-Protoshield 27  
 Proto-Shield 26, 131  
 PTC 31  
 Puls-Sensor 210  
 Pulsweitenmodulation siehe PWM  
 PWM 18, 48

## Q

Q-Touch-Bibliothek 121

## R

RGB Color Wheel 326  
 RGB-LED 324  
 RGB-Matrix 256  
 RPI-1031 (Tilt-Sensor) 94

## S

Schmalbandfunk 221  
 Schnittstelle  
     serielle 220  
 Schrumpfschlauch 237  
 SCT-013-030 (Wechselstrom-Messwandler)  
     195  
 SD-Card (Bibliothek) 274  
 SD-Card-Shield 273  
 SD-Karte 70  
 SD-Karten-Adapter 73  
 Sensirion 171  
 Sensor  
     gerader Zustand 110  
     Piezo 102  
 Sensor-Anwendung 333  
 Sensor-Board  
     Batteriebetrieb 343  
     Bestückungsplan 336  
     Eigenschaften 333  
     Leiterplattendaten 334  
     Programmierung 337  
     Schaltung 334  
     Stückliste 336  
     Varianten 340  
 Sensordaten-Erfassung 299  
 Sensor-Shield 317  
     Anschlussmöglichkeiten 317, 320  
     Open Source 332  
 Serielle Schnittstelle 220  
 Serielle Übertragung 219  
 Serieller Bus 38  
 Serieller Monitor 219  
 Serieller Plotter 216  
 setup() 22  
 Setup-Funktion 22  
 Shield 26  
 SHT31 (Umweltsensor) 171  
     Ansteuerung 174  
     Versorgungsspannungsbereich 172  
 Shunt 149  
 Sketch 15  
 Smartphone 219  
 SoftwareSerial (Bibliothek) 76, 137  
 Spannung 154  
 Spannungsteiler-Schaltung 32  
 Spannungsversorgung 17  
 Speicherkarte 273  
 SPI-Konfiguration 76  
 SRAM 18

Steckbrett 28  
 Steinhart-Hart-Gleichung 33  
 Streifen siehe Flex-Sensor  
 Strom 149  
 Strommessung 151  
   High-Side-Strommessung 152  
   kontaktlose 195  
 Stromverbrauch 333  
   Gleichstrom 189  
   Wechselstrom 194

## T

Taktfrequenz 342  
 TCS34752 59  
 Temperatur 165  
 Temperaturalarm 166  
 Temperaturmessung 31  
 Temperatursensor  
   drahtloser 234  
   DS1820 38  
   LM35 35  
 Thermistor 31  
 ThingSpeak  
   Channel 291  
 ThingSpeak (Bibliothek) 294  
 ThingSpeak (Cloud-Anbieter) 290  
   API Key 293  
   Registrierung 291  
 Tiefpass 95  
 Tilt-Sensor 93  
   AT407 94  
   RPI-1031 97  
 TinyGPS++ (Bibliothek) 138  
 TinyGPSBibliothek  
   ++ 138  
 Tongeber siehe Piezo-Sensor  
 Touch-Keybaord 119  
 Touch-Sensor 117  
   AT42QT1070 117  
 TSOP38238 (IR-Receiver) 51  
 TTL-Serial-JPEG-Cam 68

## U

Übermittlung  
   drahtlose 219  
 Übertragung  
   drahtlose 221  
   serielle 219  
 Überwachungskamera 70, 74, 75  
 Uhrenbaustein  
   DS1307 277  
 Ultraschall 81  
 Ultraschall-Sensor 81  
 Umweltgeräusch 184  
 Upload Mode 240  
 USB-Anschluss 17  
 USB-Kabel 21  
 USBTinyISP (Programmieradapter) 339  
 UV Sensor 60  
 UV-Index 60, 63, 66  
 UV-Index-Monitor 63

## V

Veränderung  
   kapazitive 117  
 VirtualWire (Bibliothek) 223  
 VirtualWire-Bibliothek 227

## W

W5100 (Ethernet-Controller) 293  
 Webserver 297  
 Wechselstrom 194  
 Wechselstrom-Messwandler 195  
 Widerstand der Haut 213  
 Widerstandsnetzwerk 265  
 WiFi-Speicherkarte 70  
 Wildkamera 80  
 WS2812 (adressierbare LEDs) 257

## X

Xively (Cloud-Anbieter) 290