



Thomas
Brühlmann
5. Auflage

Arduino Praxiseinstieg

Behandelt Arduino UNO R4 und R3



UNO R4
MINIMA

IOREF

RESET

3.3V

5V

GND

VIN

POWER

A0

A1

A2

komplett
in Farbe

Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

Ihr mitp-Verlagsteam



Thomas Brühlmann

Arduino Praxiseinstieg



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 978-3-7475-0662-2

5. Auflage 2024

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2024 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, Nicole Winkel

Sprachkorrektur: Petra Heubach-Erdmann

Coverbild: Produktbild Arduino UNO R4 © 2023, arduino.cc

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

Inhaltsverzeichnis

1	Einleitung	9
1.1	5. Auflage	9
1.2	Aufbau des Buchs	9
1.3	Mehr Informationen	12
1.4	Weitere Quellen	12
1.5	Danksagung	13
2	Arduino-Plattform	15
2.1	Am Anfang war der König	15
2.2	Be a Maker	16
2.3	Arduino-Plattform	18
2.4	Hardware	19
2.5	Software	27
2.6	Installation der Software	27
2.7	Get Connected	30
2.8	Arduino-Entwicklungsumgebung	36
2.9	Arduino-Boards	45
2.10	Arduino-kompatible Boards	48
3	Startschuss	49
3.1	Das Arduino-Board	50
3.2	Steckbrett – Experimentieren ohne Löten	58
3.3	Spannung, Strom und Herr Ohm	65
3.4	Widerstand & Co	70
3.5	Programmcode	80
3.6	Testen	86
3.7	Projekt Blink	92
3.8	Projekt Wechselblinker	95
4	Eingänge und Ausgänge	99
4.1	Digitale Eingänge	100
4.2	Digitale Ausgänge	114
4.3	Analoge Welt	117

4.4	Serielle Kommunikation	131
4.5	Drahtlose Kommunikation	152
4.6	Keyboard und Maus mit USB HID (nur Arduino UNO R4)	176
4.7	Projekt: Würfel.	180
5	Sensoren	189
5.1	Sensoren	190
5.2	Projekt Kompass mit Richtungsanzeige	246
5.3	Projekt Gefrierschrankwächter (nur Arduino UNO R3)	254
5.4	Kontaktloses Fieberthermometer	257
6	Aktoren	261
6.1	Relais.	261
6.2	Servos	264
6.3	Motoren	274
6.4	Lasten schalten.	282
6.5	Projekt: Roboter mit Wii-Steuerung.	285
7	Anzeigen	299
7.1	Leuchtdiode (LED)	299
7.2	7-Segment-Anzeigen	309
7.3	LC-Display (LCD)	316
7.4	LC-Display Nokia 3310/5110	327
7.5	OLED-Display.	329
7.6	LED-Matrix	335
7.7	Projekt Geschwindigkeitsmesser fürs Fahrrad	336
7.8	Projekt Schrittzähler	345
8	Datenverarbeitung	351
8.1	Daten speichern	351
8.2	Daten ins Internet senden.	365
8.3	Projekt: Programmierbarer Signalgeber	365
8.4	Sinusgenerator.	372
8.5	Projekt Sollwerteingabe mit Drehgeber	374
9	Erweiterungen	391
9.1	Bibliotheken	391
9.2	Hardware-Erweiterungen (Shields)	400
9.3	Hardware-Adapter	409

10	Arduino im Einsatz	413
10.1	Verbindung zum Internet.	413
10.2	Verbindung zum Internet.	427
10.3	ESP-Module und Arduino.	436
10.4	XML einlesen.	448
10.5	Umweltdaten sammeln.	462
10.6	Projekt Wetterstation (nur Arduino UNO R3).	471
11	Fehlersuche/Troubleshooting	493
11.1	Allgemeines Vorgehen	493
11.2	Fehler in der Schaltung.	493
11.3	Fehler im Programm.	494
11.4	Probleme mit der IDE.	495
11.5	Hallo Arduino-Board.	495
12	DIY Boards und Clones	497
12.1	Boards.	497
12.2	Programmieradapter (USB-Wandler)	504
12.3	Programmierung über ICSP (nur Arduino UNO R3)	507
12.4	Arduino im Miniaturformat mit ATtiny	512
13	Tools für Praktiker	539
13.1	Hardware	539
13.2	Leiterplatten herstellen	559
A	Codereferenz	567
B	Boards	601
C	Bezugsquellen	607
	Stichwortverzeichnis	609

Einleitung

1.1 5. Auflage

Es ist das Jahr 2023 und vor Ihnen liegt die 5. Auflage von »Arduino Praxiseinstieg«. Auch über 10 Jahre nach der Erstveröffentlichung dieses Buch-Titels gehört Arduino weiterhin zu den wichtigsten Microcontroller-Boards für Maker und im Bildungsbereich.

Das Arduino-Projekt hat sich im letzten Jahr auf seine ersten Schritte zurückbesinnt und einen Bausatz mit einem Arduino UNO zum Selberlöten, das »Arduino Make Your UNO Kit« vorgestellt. Jeder Bastler kann sich nun einen eigenen UNO mit konventionellen Bauteilen von Hand zusammenbauen, um die Technik und alle Komponenten zu erlernen.

Im Frühling 2023 hat das Arduino-Team den neuen Arduino UNO R4 angekündigt.

Ende Juni 2023 wurde dann endlich der neue Arduino UNO R4 vorgestellt: das Standardmodell UNO R4 Minima und ein UNO R4 Wifi. Der UNO R4 Minima ist der Nachfolger des bisherigen UNO R3. Der UNO R4 Wifi hat, wie der Name aussagt, eine Wifi-Schnittstelle für Internet- und IoT-Anwendungen. Beide Boards basieren auf einem neuen Microcontroller und bieten eine Menge zusätzlicher Funktionen.

In dieser 5. Auflage werden die beiden neuen UNO R4 ausführlich beschrieben. Das bisherige Standardboard UNO R3 spielt aber weiterhin eine Rolle in der Arduino-Community, da bisher noch nicht alle Bibliotheken und Funktionen auf die neuen Boards umgeschrieben sind. Auch gibt es natürlich noch Tausende von Projekten, Beispielen und Tutorials, die mit einem Arduino UNO R3 betrieben werden.

Darum wird in dieser Auflage bei den Beispielen und Projekten jeweils erwähnt, für welches Board die Code-Beispiele ausgelegt und lauffähig sind.

1.2 Aufbau des Buchs

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 Buchs 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.

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. Zusätzlich wird der UNO R4 Wifi im drahtlosen Einsatz als Webclient und Webserver erklärt. 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. Zusätzlich gibt es noch ein Kapitel zum Download auf www.miptp.de/0661, das Wifi mit ESP8266 für den Arduino UNO R3 beschreibt.

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.

Im Schlussteil und Anhang des Buchs 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. Ein Bonus-Kapitel zum Download unter www.mitp.de/0661 listet alle Stücklisten der Beispiele und Projekte aus dem Buch auf.

1.3 Mehr 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/>

Viele Maker-Projekte, Reviews und Tipps gibt's auf der Maker-Plattform des Autors:

<https://www.555circuitslab.com/>

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

Die E-Mail-Adresse zum Buch lautet:

kontakt@arduino-praxis.ch

Der Twitter bzw. X-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.

1.4 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:

<https://arduino.cc>

Make:Blog:

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

Explore & Learn Adafruit:

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

1.5 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 5. Auflage dieses Buchs mussten sie wieder viele Stunden ohne mich auskommen. Die Skiferien haben sie abermals allein verbracht und auch an den Wochenenden waren viele Stunden für das Schreiben des Buchs verplant. Aber alle haben sehr viel Verständnis gezeigt und mich dabei unterstützt.

Ein Dank geht an die Arduino-Community. Ohne sie wäre dieses Buch nicht entstanden und 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 5. 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 5. Auflage zu schreiben.

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 Elektronikingenieuren 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 Vorstellung des Arduino-Projekts vor fast 20 Jahren hat eine richtige Selbstbau-, DIY- und Makerszene hervorgerufen. In den letzten Jahren haben viele findige Tüftler neue Board-Varianten erstellt und Firmen bieten Produkte, 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.

Aus den Originaldaten des Standardboards des Arduino sind viele neue Projekte und Elektronik-Produkte auf den Markt gekommen und viele auch wieder verschwunden.

Das Arduino-Projekt hat zahlreiche Schüler, Studenten und junge Leute animiert, sich mit Elektronik zu beschäftigen und eine berufliche Karriere im technischen Umfeld zu starten.

Auch geht das Arduino-Projekt nochmals auf seine Ursprünge zurück und hat im Herbst 2022 einen Arduino-Bausatz zum Selberlöten herausgebracht. Mit dem Projekt »Arduino Make Your UNO Kit« wird ein Schritt zurück in die Zukunft gestartet.

<https://blog.arduino.cc/2022/11/03/are-you-ready-to-go-back-to-the-future/>

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 Lüfterfrischerlösung 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.

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 auseinandergelöst, geforscht und gebastelt werden, damit viele spannende und auch lustige Anwendungen entstehen. Wäh-

renddessen 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.



Abb. 2.1: Indianerfigur aus Abfallteilen einer Elektronikproduktion

Elektronik-Basteln als Hobby hat sich zu einem richtigen Markt entwickelt. Viele Online-Elektronikplattformen und Community-Websites bilden die sogenannte Maker-Community und liefern Anleitungen, Ideen, Bausätze und Komponenten für den Selbstbau. Stellvertretend für diese Anbieter sind Adafruit Industries (<https://www.adafruit.com>) und Sparkfun (<https://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 Basteleien 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 (<https://makerfaire.com/>) werden jeweils die kommenden Messen angekündigt. In Europa ist speziell die

Maker Faire Rome (<https://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 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 (<https://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 Ausgänge 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

und ist die Plattform, auf der die ausführbaren Programme erstellt werden, die dann von dem Hardwareteil ausgeführt werden.

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

Gleichzeitig dient die Verbindung auch als Hilfsmittel bei der Fehlersuche. Mithilfe eines seriellen Monitors können die Zustände der Ein- und Ausgänge oder die von den Sensoren gelieferten Daten überwacht werden.

Nach der Entwicklung kann die Verbindung zur Entwicklungsumgebung getrennt werden, und der Arduino kann als unabhängiger Minicomputer mit der Außenwelt kommunizieren.

Falls aber Daten zur Weiterverarbeitung oder zur Anzeige an den Computer gesendet werden müssen, bleibt die serielle Verbindung über den USB-Anschluss bestehen.

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-C-Schnittstelle liefert eine Spannung von 5 Volt über die das Board und die externen Sensoren und Module versorgt werden.

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 seriellen Monitor.

Das Standardboard des Arduino (Stand August 2023) heißt Arduino UNO R4. Alle Erklärungen und Beispiele in diesem Buch beziehen sich auf den Arduino UNO R4 Minima. Einzelne Code-Beispiele funktionieren aber nur mit dem UNO R3, da noch nicht alle Bibliotheken für das gerade herausgekommene Modell R4 umgeschrieben sind.

Bei den Beispielen wird jeweils erwähnt, auf welchem Modell der Code lauffähig ist.

Der Arduino UNO R4 kann man momentan (Stand September 2023) nur über Shop von Arduino kaufen.

Der Arduino UNO R3 war jahrelang das Standard-Board und kann weiterhin von verschiedenen Anbietern (siehe Anhang C, »Bezugsquellen«) bezogen werden.

Zusätzlich gibt es eine große Anzahl von kompatiblen Boards (Clones) bei vielen Händlern und Anbietern. Diese bieten auch Bausätze oder nur die Leiterplatten des UNO R3 an.

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

In den letzten zehn Jahren haben viele innovative Bastler Arduino-Boards in zahlreichen Projekten 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-Programm wird jeweils bei Anschluss der Spannungsversorgung am Microcontroller gestartet.

Der UNO R4 ist mit einem modernen 32-Bit Microcontroller ausgerüstet. Zum aktuellen Zeitpunkt sind von diesem Board noch keine Nachbauten im Internet verfügbar.

2.4.1 Arduino Uno R4

<https://store.arduino.cc/collections/boards/products/uno-r4-minima>

Der Arduino UNO R4 ist das aktuelle Standard-Board des Arduino-Projekts. In Abbildung 2.2 ist das Model UNO R4 Minima abgebildet.

Auf dem Arduino UNO R4 sind verschiedene Stecker und Anschlussbuchsen montiert. Die Position der verschiedenen Anschlussbuchsen ist fix definiert und stellt die Kompatibilität mit früheren und zukünftigen Boards sicher.

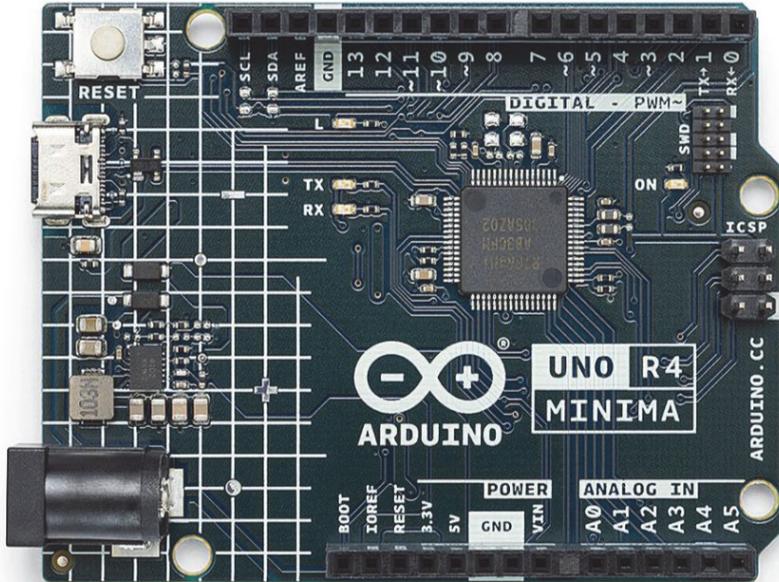


Abb. 2.2: Arduino UNO R4 Minima (Quelle: www.arduino.cc)

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

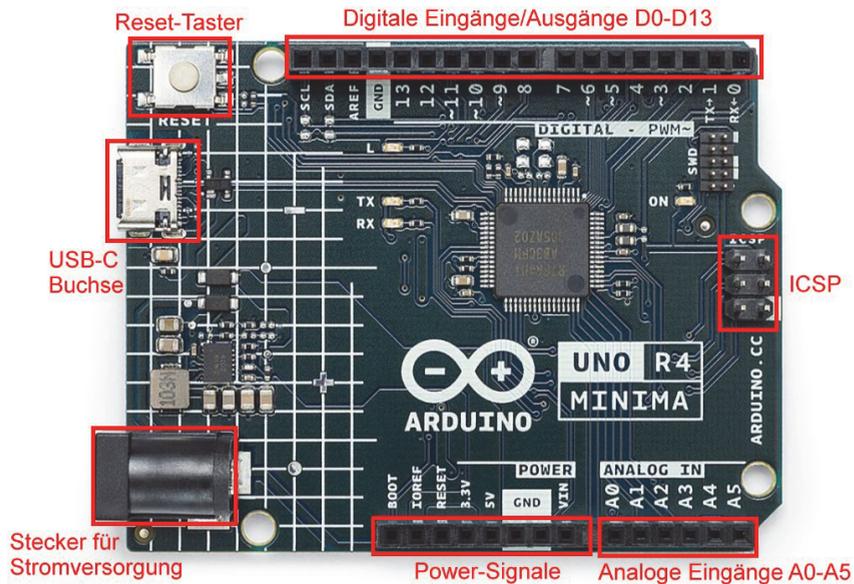


Abb. 2.3: Arduino UNO R4: Anschlussmöglichkeiten

Nachfolgend werden die einzelnen Anschlussmöglichkeiten beschrieben.

USB-Buchse: Der USB-Anschluss (USB-C) 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.

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

Stromversorgung: Mit der Schaltung der Stromversorgung wird die über den 2.1 mm-Hohlstecker zugeführte Versorgungsspannung auf 5 V geregelt.

Microcontroller: Der Microcontroller ist das Gehirn des Arduino UNO, er führt die Programme aus und verarbeitet die Ein- und Ausgangssignale.

SWD-Anschluss: Über den SWD-Anschluss kann der Entwickler einen externen Debugger (J-Link-Debugger) anschließen, um die Funktionen des Microcontrollers zu überwachen und zur vertieften Fehlersuche im System.

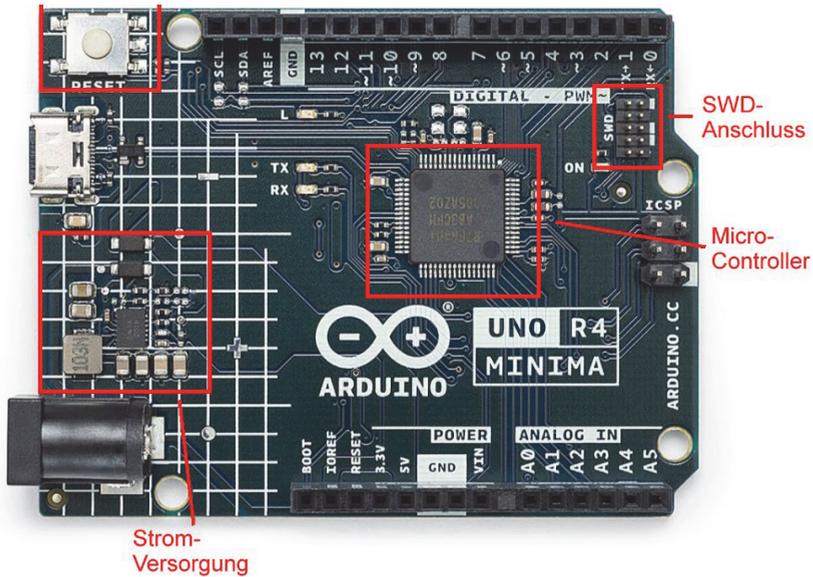


Abb. 2.4: Arduino UNO R4 Minima: Bauteile und Funktionen

In Tabelle 2.1 sind die technischen Daten des Arduino UNO R4 aufgelistet.

Beschreibung	Detailldaten
Microcontroller	R7FA4M1AB3CFM#AA0 (Renesas) 48 MHz Arm Cortex-M4
Spannungsversorgung	6–24 VDC
Betriebsspannung	5 VDC und 3,3 VDC (intern über Spannungsregler generiert)
Digitale Ein-/Ausgänge	14 (D0-D13) davon 6 als PWM (D3, D5, D6, D9, D10, D11)
Analoge Eingänge	6 (A0-A5)
Strom pro digitalem Pin	8 mA DC
Flash Memory	256 KB
SRAM	32 KB
EEPROM	8 KB
Taktfrequenz	48 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 R4

2.4.2 Arduino UNO R4 WIFI

<https://store.arduino.cc/products/uno-r4-wifi>

Gleichzeitig mit dem UNO R4 Minima wurde noch ein zweites Board vorgestellt: der Arduino UNO R4 WIFI (Abbildung 2.5).

Wie die Typenbezeichnung »WIFI« aussagt, hat dieses Board eine WIFI-Schnittstelle eingebaut, die mit einem ESP32 realisiert wurden.

Der Arduino UNO R4 WIFI eignet sich somit als Webserver oder Webclient und für den Einsatz als IOT- oder Sensor-Node mit WLAN-Schnittstelle.

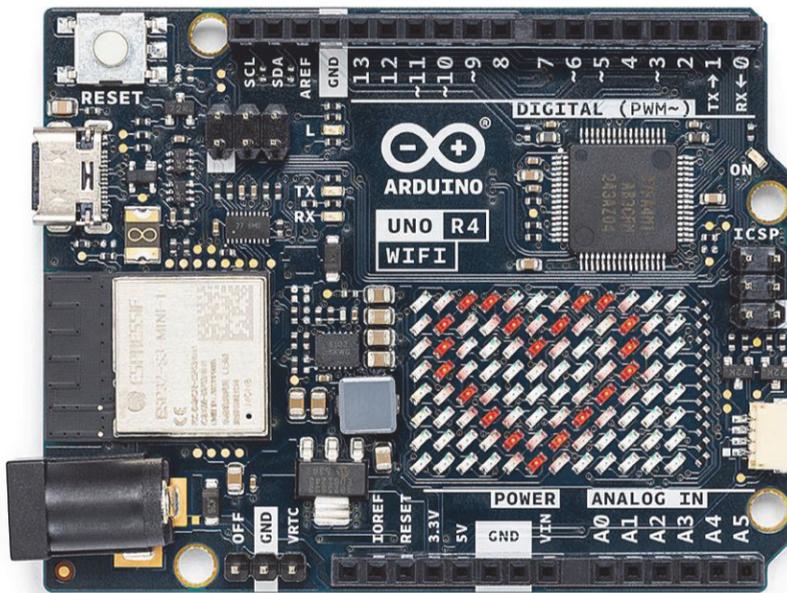


Abb. 2.5: Arduino UNO R4 WIFI (Quelle: www.arduino.cc)

In Kapitel 10 wird die WIFI-Funktionalität des UNO R4 WIFI im Detail erklärt.

2.4.3 Arduino UNO R3

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

Der Arduino UNO R3 ist der weitverbreitete Vorgänger des UNO R4. In Abbildung 2.6 ist das Board abgebildet.

In Abbildung 2.7 sind die Anschlussmöglichkeiten auf der Leiterplatte des Arduino UNO R3 gekennzeichnet.

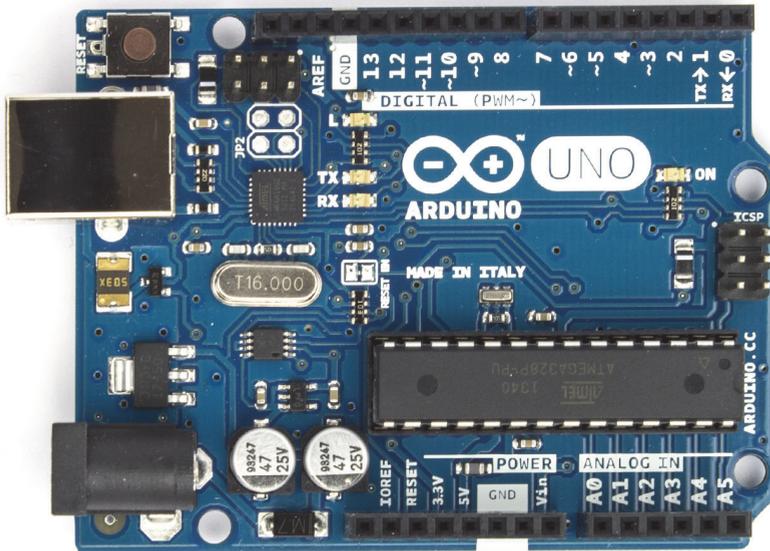


Abb. 2.6: Arduino Uno Rev 3 (Quelle: www.arduino.cc)

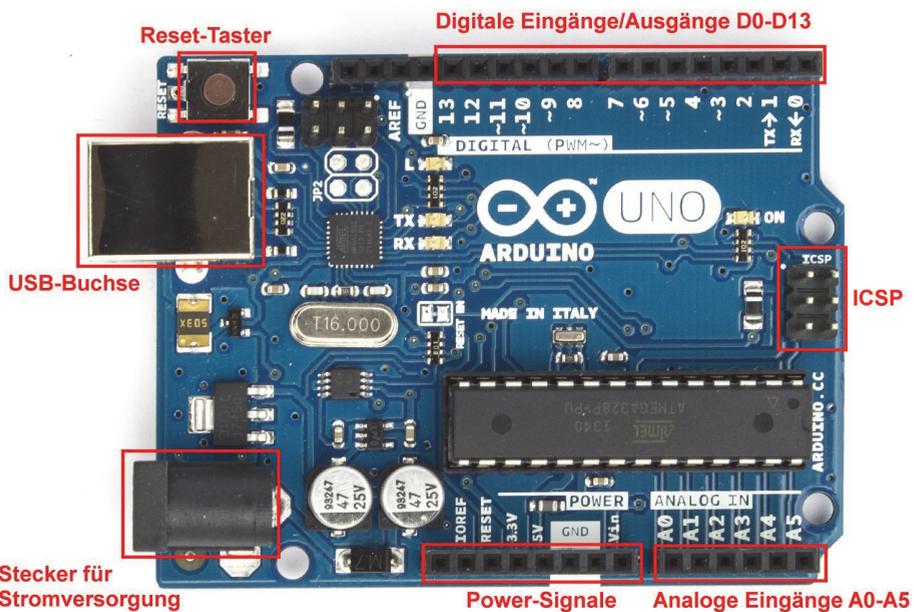


Abb. 2.7: Arduino UNO R3: Anschlussmöglichkeiten

In Abbildung 2.8 sind die Bauteile und Funktionen des UNO R3 abgebildet.

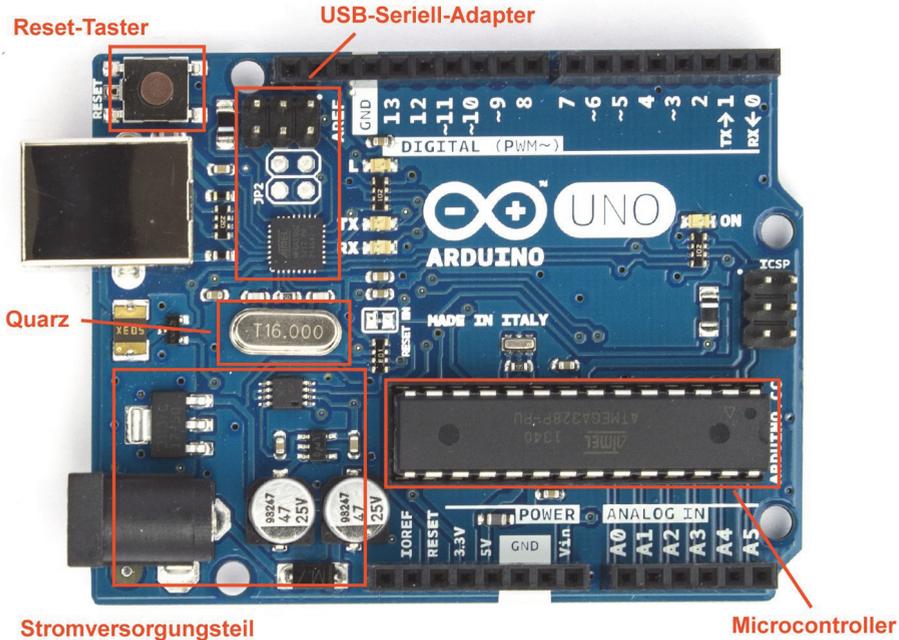


Abb. 2.8: Arduino UNO R3: Bauteile und Funktionen

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

USB-Seriell-Adapter: Dieses 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.

Stromversorgungsteil: Im Stromversorgungsteil wird die Versorgungsspannung, die über den schwarzen 2,1-mm-Klinckenstecker 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 2.2 sind die technischen Daten des Arduino UNO R3 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 2.2: Technische Daten Arduino UNO R3

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, macOS und Linux) verfügbar und wird laufend aktualisiert und erweitert. Mit dem Download und der Installation der Software erhält 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.

2.6 Installation der Software

Im Downloadbereich der Arduino-Website (<https://www.arduino.cc/en/software>) stehen die aktuellen Versionen der Arduino-Entwicklungsumgebungen als Archivdatei zur Verfügung. Zum Zeitpunkt der Drucklegung dieses Buches (Herbst 2023) lautete die aktuelle Programmversion 2.2.1.

2.6.1 Installation

Die nachfolgende Installation bezieht sich auf die Betriebssysteme Windows und macOS. 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 Installationsdatei hat in der aktuellen Programmversion eine Dateigröße von rund von 160 MB.

2.6.2 Installation Windows

Für Windows-Benutzer stehen mehrere Versionen zum Download bereit: die Installationsdatei für Win10/11 als EXE-Datei, ein MSI-Paket und eine ZIP-Version. Nach erfolgreichem Download der Arduino-Umgebung kann die Software ausgeführt oder mit einem Archivprogramm entpackt werden.

Die Variante mit der ausführbaren Installationsdatei (EXE) erfordert Administrationsrechte auf dem Rechner. Bei dieser Variante werden auch alle notwendigen Daten installiert. Die Installationsroutine installiert die Arduino IDE standardmäßig ins Verzeichnis

```
C:\Program Files\Arduino IDE
```

Die ZIP-Version kann ohne Administratorrechte auf den Rechner kopiert, entpackt und ausgeführt werden.

Beim ersten Start der Entwicklungsumgebung kann das Windows-System ein Hinweis bezüglich der Firewall Windows Defender anzeigen. Diese Meldung kann dann mit »Zugriff zulassen« bestätigt werden.

Praxis-Tipp

Für erfahrene Anwender empfehle ich die Installation der Entwicklungsumgebung jeweils in einen eigenen Ordner mit Versionsnummer (C:\Programme\arduino-2-0-3).

Auf diese Art kann in einer Version eine Einstellung oder Erweiterung vorgenommen werden, ohne dass die »produktive« Hauptversion beeinträchtigt ist. Die ist praktisch, wenn man mit verschiedenen Board-Varianten arbeitet und dabei die jeweiligen Board-Pakete verwendet (siehe dazu Kapitel 12).

2.6.3 Installation macOS

Unter macOS verschiebt man das ausführbare Programm nach dem Download ins Verzeichnis /Programme oder /Applications. Die ausführbare Datei heißt `Arduino.app` (Abbildung 2.9).

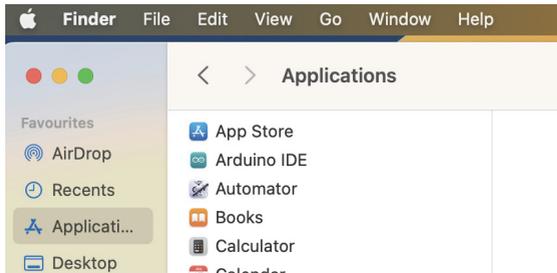


Abb. 2.9: macOS: Ausführbares Programm

Somit ist die Entwicklungsumgebung installiert beziehungsweise auf den Rechner kopiert. Falls die Installation der ZIP-Variante ausgeführt wird, muss noch ein USB-Treiber installiert werden. Für die Installationsvariante ist die Treiber-Installation automatisch erledigt. Unter macOS ist keine Treiberinstallation notwendig.

2.6.4 Installation des USB-Treibers unter Windows 10/11

Der erforderliche Treiber für die USB-Schnittstelle ist bereits bei der Installation der Entwicklungsumgebung mitkopiert worden und liegt im Verzeichnis

```
arduino-verzeichnis\drivers\
```

Um den Treiber zu installieren, muss eine Verbindung mit dem Arduino-Board hergestellt werden. Zu diesem Zweck wird ein USB-Kabel an einem verfügbaren USB-Anschluss des Rechners angeschlossen und mit dem USB-Port auf dem Arduino-Board verbunden. Auf dem Arduino-Board leuchtet die Leuchtdiode »On« und zeigt an, dass das Board mit Spannung versorgt wird.

Sofort startet auf dem Rechner eine Installationsroutine, die versucht, den Treiber für das Arduino-Board zu installieren. Kurze Zeit später meldet die Installationsroutine, dass der Treiber nicht installiert werden konnte.

Der Treiber muss nun manuell nachinstalliert werden.

1. Auf dem Rechner `START|SYSTEMSTEUERUNG` auswählen.
2. `GERÄTE-MANAGER` auswählen.
3. Unter `ANSCHLÜSSE (COM & LPT)` erscheinen der angeschlossene Arduino und der COM-Port.
4. Arduino-Board anklicken, rechte Maustaste.

5. Option TREIBERSOFTWARE AKTUALISIEREN auswählen.
6. Auf die Frage, wie die Treibersoftware gesucht wird, Option AUF DEM COMPUTER NACH TREIBERSOFTWARE SUCHEN auswählen.
7. Oben genannten Treiberpfad (ORDNER/DRIVERS) auswählen.
8. Nach Bestätigung wird die Treiberinstallation gestartet.
9. Anschließend meldet das System die erfolgreiche Installation des Treibers.

Im Geräte-Manager erscheint nun das Arduino-Board mit korrekt installiertem Treiber und der zugeordneten Schnittstelle (Abbildung 2.10)

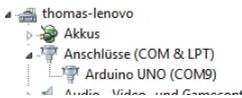


Abb. 2.10: Geräte-Manager mit Arduino Uno und USB-Port

Hinweis

Damit dieser Port im Geräte-Manager angezeigt wird, muss der Rechner mit dem Arduino-Board verbunden sein.

2.6.5 Installation des USB-Treibers unter macOS

Die Installation eines USB-Treibers ist für den Arduino UNO und Arduino Mega 2560 nicht erforderlich. Für ältere Arduino-Boards sowie für alle Arduino-Clones mit einem FTDI-Chip muss der Treiber installiert werden. Der Treiber für den USB-Port kann von der Website des Treiberherstellers geladen werden (<https://www.ftdichip.com/Drivers/VCP.htm>).

Nach einem Doppelklick auf die Treiberdatei startet der Installationsprozess. Befolgen Sie zunächst die Installationsanweisungen und starten Sie zum Schluss den Rechner neu.

2.6.6 Installation des USB-Treibers unter Linux

Der aktuellste Treiber für Linux kann auf der Website des Chip-Herstellers FTDI heruntergeladen werden (<http://www.ftdichip.com/Drivers/VCP.htm>). Nachdem die Archivdatei `ftdi_sio.tar.gz` in einem temporäres Verzeichnis entpackt ist, folgen Sie den Beschreibungen in der README-Datei.

2.7 Get Connected

Mit der Installation der Entwicklungsumgebung (IDE) und des USB-Treibers ist auf dem Rechner bereits alles Nötige für die Entwicklung von Arduino-Anwendungen vorhanden.