

Arduino

Projekte für Haus und Garten

- Überblick über die wichtigsten Arduino-Boards.
- Mit einem Arduino-Set den einfachen Einstieg wagen.
- Viele praktische Projekte für Haus & Garten zum Nachbauen.



René Gäbler

Arduino

Projekte für Haus und Garten

Das Einsteigerseminar



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.

Reihe: Das Einsteigerseminar

Titel: Arduino – Projekte für Haus und Garten

Autor: René Gäbler

Lektorat: Petra Kleinwegen

Satz: Petra Kleinwegen

Umschlagsgestaltung: Quick Hsu

EAN/ISBN: 9783828717541

1. Auflage 2019

www.bhv.de

E-Mail: info@bhv.de

Telefon: +49 2253 / 92822 20

Telefax: +49 2253 / 92822 29

© 2019 bhv Verlag GmbH

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.

Inhalt

Einleitung	7
Arduino – Was ist das und wozu ist das gut?	7
Warum es jeder machen kann und nicht nur die Technik-Gurus	8
Ohne E-Technik und Programmierkurs Mikrocontroller verwenden – geht das überhaupt?	8
1. Die Arduino IDE installieren	9
1.1 Was ist die Arduino IDE und wozu braucht man sie?	9
1.2 Die Installation der Arduino IDE auf einem Windows-10-Rechner	11
1.3 Die Installation der Arduino IDE auf einem Mac-OSX-Rechner	13
1.4 Die Installation der Arduino IDE auf einem Linux-Rechner	14
1.5 Was sind Bibliotheken?	15
2. Ein erster Blick auf die Oberfläche der Arduino IDE	17
2.1 Die Arduino IDE starten	17
2.2 Ein erster Blick auf die Arduino IDE	21
3. Ein kurzer Überblick über die wichtigsten Arduinoboards	23
3.1 Was ist ein Arduinoboard und wozu brauche ich das?	23
3.2 Was man bei einem Arduinoboard wissen muss	25
3.3 Die wichtigsten Arduinoboards	28
Arduino Nano	29
Arduino Uno	29
Arduino Uno Rev 3	30
Arduino Mega 2560	30
Arduino Mega ADK	31
Arduino Ethernet	32
Arduino Leonardo	32
Arduino Micro	33
Arduino Esplora	33

Arduino Due	34
Arduino Mini	35
Arduino Flo	35
Gemma	36
Lilypad Arduino	36
LilyPad Arduino USB	37
LilyPad Arduino Simple	37
LilyPad Arduino SimpleSnap	38
Arduino Yun	38
Arduino Yun	39
Arduino Yun Mini	39
Arduino MKR WiFi 1010	40
Arduino MKR Vidor 4000	41
Arduino MKR WAN 1300	41
Arduino ISP	42
Arduino GENIO ZERO	42
Arduino Steuerungsboard	42
Arduino Motorboard	43
Arduino Galileo / Edison	44
3.4 Die wichtigsten Bestandteile eines Arduinobords	44
4. Eine kleine Einführung in die Welt der Elektronik	47
4.1 Warum Sie diese Einführung überspringen können, aber dennoch lesen sollten	47
4.2 Ein Blick in ein Starterset	47
4.3 Die wichtigsten Bauelemente, wie sie funktionieren, was sie tun und wozu man sie braucht	53
Das Steckbrett	53
Die Batterieklemme	56
Das Netzteil	56
Die Steckbrettkabel	57
LEDs	58
RGB-LED	59
Taster	59
Widerstände	60

Der Feuchtigkeitssensor	61
Die Relaiskarte	62
Das Drehpotentiometer	62
Der Temperatursensor	63
Der Bewegungsmelder	65
Der Piezo Speaker	66
5. Mit einem Arduino-Set einen einfachen Einstieg wagen	69
5.1 Was ist ein Arduino-Set und wozu ist dieses gut?	69
5.2 Ein kleiner Überblick über verschiedene Arduino-Sets	71
Sets von Arduino	71
Lernsets aus dem Hause Funduino	72
Sets aus dem Hause Elegoo	73
Und was gibt es noch?	76
6. Eine kleine Einführung in den Umgang mit dem Arduinoboard	79
6.1 So verbinden Sie den Arduino Uno mit Ihrem PC	79
6.2 So verbinden Sie den Arduino Mega 2560 mit Ihrem PC	82
6.3 So verbinden Sie den Arduino Nano mit Ihrem PC	85
6.4 Immer auf den neusten Stand – Aktualisieren Sie die Bibliotheken in der Arduino IDE	87
6.5 So aktualisieren Sie die Boardinformationen in der Arduino IDE	89
6.6 Ein installiertes Boardpaket mit einem Update auf den neusten Stand bringen	91
6.7 Die Arduino IDE aktualisieren	94
6.8 Die Boardinformationen holen	95
6.9 Wozu sind Bibliotheken da und welche brauche ich?	97
6.10 Einen Sketch kompilieren	99
6.11 Einen Sketch aus der Arduino IDE auf das Arduinoboard übertragen und starten	100
6.12 Einen Sketch speichern	101
6.13 Das erste Projekt	104
Mit dem Beispiel-Programm „Blink“ die OnBoard-LED zum Blinken bringen	105

7.	Programmieren für Einsteiger	109
7.1	Gibt es nun doch einen Programmierkurs?	109
7.2	Ein paar grundlegende Dinge zur Arduino IDE	109
7.3	Beispiel-Sketches laden und verwenden	112
7.4	Die Kennzeichnung der Codeinhalte	117
7.5	Die Elemente eines Arduino-Programmes	118
7.6	Die wichtigsten Befehle in einem Sketch	121
	Allgemeine Befehle	122
	Befehle der Bibliothek LiquidCrystal	123
	Befehle der Bibliothek DS3231 (für den RTC)	125
	Befehle der Bibliothek RTCLib	127
8.	Ein Überblick über die Projekte in diesem Buch	129
8.1	Bevor es losgehen kann – Die Einkaufslisten verwenden	129
8.2	Wie die Projekte beschrieben werden	132
8.3	Die Schaltübersichten	134
8.4	Fritzing installieren und verwenden	136
8.5	Schaltpläne mit Fritzing ansehen	139
8.6	Die Projekte im Buch	143
	Erst stecken, später löten – So geht´s	143
	Automatisch Blumengießen mit einem Bodensensor	144
	Eine Uhr selbst bauen	145
	Automatisch Blumen gießen mit einem Timer und dem Feuchtigkeitssensor	146
	Mit dem Bewegungsmelder ungestört bleiben	146
	Temperaturen messen und auf dem LCD anzeigen lassen	147
9.	Arduino-Projekte für Haus, Hobby und Garten	149
9.1	Wie finde ich die richtigen Bauelemente?	149
9.2	Erst stecken, später löten – So geht´s	153
9.3	Automatisch Zimmerpflanzen mit einem Bodensensor gießen	172
	Der Feuchtigkeitssensor	174
	Die Wasserpumpe	177
	Der Adapter	179

Die Relaiskarte	180
Der serielle Monitor	182
Diese Teile benötigen wir für unser Projekt	184
Das Projekt aufbauen	185
Der Sketch	194
10. Arduino-Projekte für Haus, Hobby und Garten, Teil 2	199
10.1 Blumengießen mit einem Timer	199
Arduinoboard vs. Real Time Clock	199
Die Bibliothek für den DS3231 installieren	201
Diese Teile benötigen wir für unser Projekt	206
Das Projekt aufbauen	207
Der Sketch	212
10.2 Eine Uhr selbst bauen	214
Das LC-Display	214
Diese Teile benötigen wir für unser Projekt	218
Eine kleine notwendige Vorarbeit – Die Uhr wird gestellt	220
Verschiedene Bibliotheken – Welche nehme ich und warum?	223
Das LCD anschließen	225
Das LC-Display ausprobieren	229
Der Sketch für unsere Uhr	231
11. Arduino-Projekte für Haus, Hof und Garten	241
11.1 Automatisch Blumen gießen mit einem Timer und einem Bodensensor	241
11.2 Piezo Speaker, Bewegungsmelder, Temperatursensor und Potentiometer – Neue Bauteile kurz vorgestellt	244
11.3 Mit einem Bewegungsmelder ungestört bleiben	246
Diese Teile benötigen wir für unser Projekt	246
Das Projekt aufbauen	247
Der Sketch zum Projekt	250
11.4 Temperaturen messen und auf einem LCD ausgeben	251
Diese Teile benötigen wir für unser Projekt	251
Das Projekt aufbauen	252
Der Sketch zum Projekt	253

Anhang	257
Glossar	257
Wichtige Webadressen	260
Index	263

Einleitung

Mit diesem Buch möchte ich Ihnen den Arduino und seine Möglichkeiten näherbringen. In einem Einstieg erfahren Sie, was der Arduino ist, wozu man ihn verwenden kann und wer diese Möglichkeiten nutzen kann. Sie lernen einige elektronische Grundbegriffe kennen und erfahren, wo Sie die verschiedenen Arduino-Sets erwerben können. Ich stelle Ihnen die Arduino IDE vor und ebenso die verschiedenen Arduinobords, die es im Handel momentan gibt. Sie erfahren, wie Sie den Arduino mit dem PC verbinden und wie Sie die verschiedenen Bauelemente auf den Arduino aufstecken. Nach dem Aufbringen der Bauelemente lernen Sie auch das Schreiben eines Programmes, ohne das es leider nicht geht. Danach endlich geht es an die verschiedenen und sehr spannenden Projekte. Sie werden eine LED zum Blinken bringen, Ihre Blumen automatisch gießen lassen und mit einer Webcam die Katze beobachten können.

Arduino – Was ist das und wozu ist das gut?

Arduino ist ein Mikrocontroller-Board, das mit verschiedenen Bauelementen und einem Programm eine bestimmte Aufgabe erfüllen kann. Arduino ist ein Open-Source-Projekt, das heißt, die Software kann kostenlos aus dem Internet geladen werden und Sie können sie verwenden, ohne befürchten zu müssen, dass Sie irgendwann etwas zahlen müssen. Sie können auch Ihre eigenen Projekte an andere weitergeben oder Projekte aus dem Internet nachbauen.

Das Wichtigste ist: Arduino macht einen Heidenspaß! Sie müssen kein Elektroniker, Programmierer oder Hobbybastler sein. Sie können die im Buch vorgestellten Projekte sehr einfach umsetzen. Sie finden bei jedem Projekt eine Beschreibung, eine Liste der notwendigen Bauteile und ein Programm vor. Letzteres tippen Sie einfach ab. Natürlich erfahren Sie im Buch auch Grundlegendes zur Programmstruktur und zu den Elementen in einem Arduino-Programm. Und ich bringe Ihnen ein paar Elektrotechnik-Bauteile näher und verrate Ihnen, was diese tun und wozu sie gut sind. Und Sie können nicht nur die im Buch vorgestellten Programme und Projekte umsetzen. Jedes Projekt lässt sich erweitern und verbessern. Sie finden zudem im Internet viele spannenden Projekte anderer Arduino-Anwender und können natürlich auch eigene Ideen umsetzen.

Warum es jeder machen kann und nicht nur die Technik-Gurus

Wie bereits gesagt müssen Sie kein Spezialist auf dem Gebiet der Elektrotechnik sein. Sie müssen sich nicht mit dem Entwickeln von Programmen auskennen. Zu einem Arduino-Set gibt es immer eine Steckplatte und viele Bauelemente, mit denen Sie sofort loslegen können. Die Elemente werden auf die Steckplatte aufgebracht. Per USB-Kabel verbinden Sie das Arduino-Board mit Ihrem PC und tippen das Programm ab, das Sie für Ihr Projekt brauchen. Dann wird dieses übertragen und gestartet. Und fertig!

Jedes Projekt im Buch habe ich aufgebaut und geprüft. Jedes Programm wurde auf Herz und Nieren getestet und erst dann im Buch niedergeschrieben. Und damit es noch einfacher geht, finden Sie eine aktuell gehaltene Einkaufsliste und den Programmcode auf meiner Webseite. Den Code können Sie einfach per Copy & Paste auswählen und in Ihre IDE einfügen.

Ohne E-Technik und Programmierkurs Mikrocontroller verwenden – geht das überhaupt?

Ja, das geht! Das ist ja das Geniale am Arduino. Sie können spannende Projekte umsetzen und sehr coole Dinge bauen, ohne dass Sie Elektrotechniker oder Programmierer sind. Es ist schon wichtig zu wissen, was das aufgebaute Dings tut, aber Sie müssen sich nicht vorher mühselig einarbeiten und etwas lernen.

Es kann wirklich jeder machen. Dieses Buch richtet sich an keinen bestimmten Personenkreis und an kein bestimmtes Alter. Sie brauchen keinen Lötkolben und keine E-Technik-Werkstatt oder gar Messgeräte. Einzig und allein ein Arduino-Board, ein paar Bauelemente, die Sie einzeln oder als Set kaufen, und die Arduino-IDE. Diese laden Sie aus dem Internet und installieren sie auf Ihrem PC. Und schon kann es losgehen.

1. Die Arduino IDE installieren

In diesem Kapitel erfahren Sie, wozu Sie die Arduino IDE brauchen und wie Sie diese auf Ihrem Rechner installieren. Die Installation wird für Rechner mit Windows 10, Mac OSX und Linux beschrieben.

1.1 Was ist die Arduino IDE und wozu braucht man sie?

Mit der Arduino IDE erstellen Sie ein Programm und „sagen“ dem Arduino Board, was es zu tun hat. Ein solches Programm ist notwendig, damit der Arduino arbeiten kann. Die verschiedenen Bauelemente werden hier mit einem entsprechenden Programm versorgt.

Die Arduino IDE ist Open Source. Sie können das Programm kostenlos aus dem Internet laden und auf Ihrem Rechner installieren.

Zum Programm gehören auch gleich ein paar Beispiele. Sie zeigen, wie ein Arduino-Programm aussehen kann. Schauen wir uns das einmal an:

```
// the setup function runs once when you press reset or
power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
(HIGH is the voltage level)
  delay(500); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off
by making the voltage LOW
  delay(500); // wait for a second
}
```

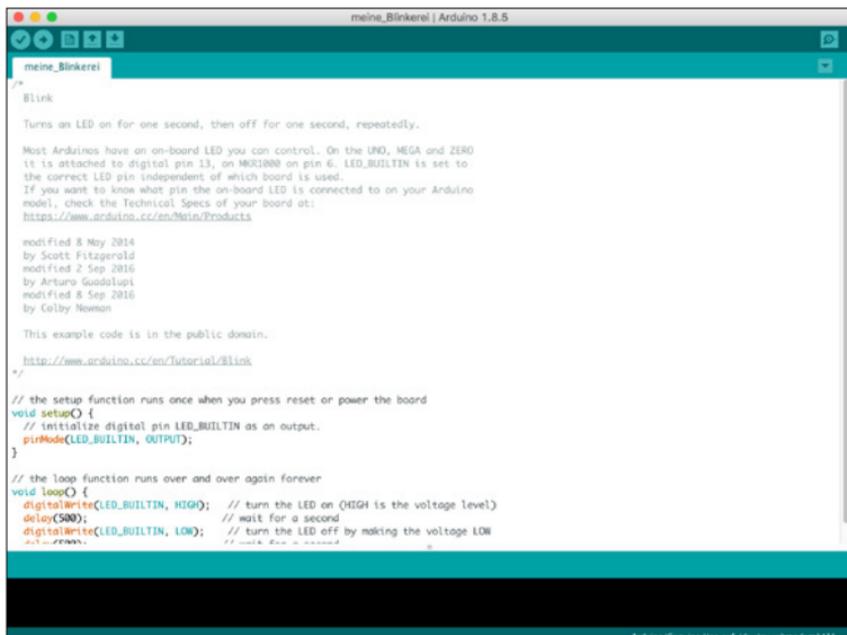


Abbildung 1.1: Die Arduino IDE ist sehr einfach aufgebaut. Es gibt keine unnötigen Funktionen und keinen grafischen Schnickschnack.

Ganz oben ist zunächst ein Kommentar eingefügt. Er verrät, was das Programm tut. Dieser Kommentar ist anhand des Zeichens // erkennbar. Mit zwei Schrägstrichen versehene Elemente werden nicht verarbeitet. So können Sie auch eigene Kommentare und Hinweise einfügen.

Der eigentliche Programmcode ist in eine geschweifte Klammer gesetzt. Mit den folgenden Befehlen wird die LED auf dem Arduino-Board angeschaltet. Ein Timer wartet eine Sekunde und ruft den nächsten Befehl auf. Die LED wird ausgeschaltet. Es folgt ein weiterer Timer. Wieder wird eine Sekunde gewartet. Der ganze Programmcode wird als Schleife (loop) wiederholt.

Dieser Beispielcode steuert also die auf dem Arduino-Board vorhandene LED an und bringt diese zum Blinken.

1.2 Die Installation der Arduino IDE auf einem Windows-10-Rechner

Im Internet finden Sie unter der Adresse <https://www.arduino.cc/en/Main/Software> den Web Editor. Er wird direkt im Browser gestartet und benötigt keine Installation.

Auf der gleichen Webseite finden Sie die Arduino IDE. Sie steht für die Plattformen Windows 8/10, Mac OSX und Linux zur Verfügung.

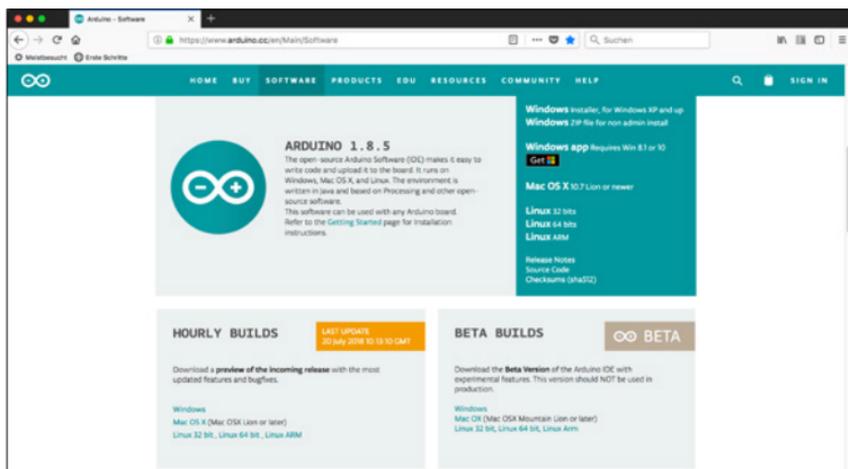


Abbildung 1.2: Im Internet finden Sie die IDE für Windows, Mac OSX und Linux.

Und so installieren Sie die Arduino IDE auf einem Windows-10-Rechner:

1. Öffnen Sie einen Webbrowser. Geben Sie die Adresse <https://www.arduino.cc/en/Main/Software> ein.
2. Klicken Sie rechts auf *Windows app*.
3. Auf der nächsten Seite werden Sie um eine Spende gebeten. Das ist natürlich freiwillig. Klicken Sie auf *Just Download*.
4. Ein weiteres Browserfenster wird geöffnet. Sie sehen eine Seite des Microsoft Stores vor sich. Hier wird Ihnen die Arduino IDE angeboten. Klicken Sie auf *Herunterladen*.

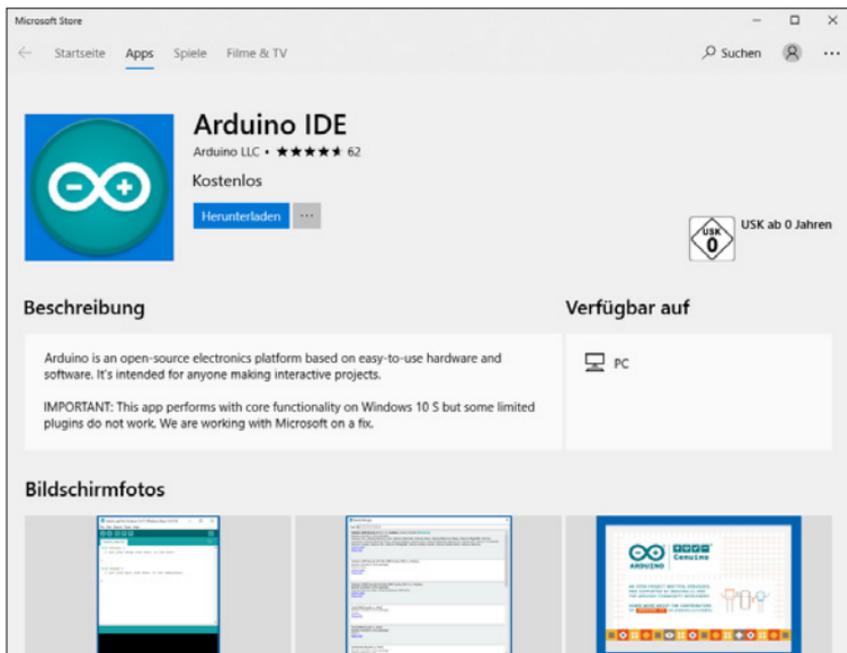


Abbildung 1.3: Die App herunterladen

- Die App wird nun auf Ihren Rechner geladen und installiert. Sie hat eine Größe von 445,6 MB. Warten Sie, bis der Download beendet ist. Schließen Sie den Browser.

Die App ist nun installiert und bereit zum Start. Sie finden sie im Windows-10-Kachelmenü unter *Kürzlich hinzugefügt* und unter *Arduino IDE*.

Natürlich können Sie auch gleich den Windows Store aufrufen und die Arduino IDE suchen. Wie Sie hier vorgehen, bleibt ganz Ihnen überlassen.

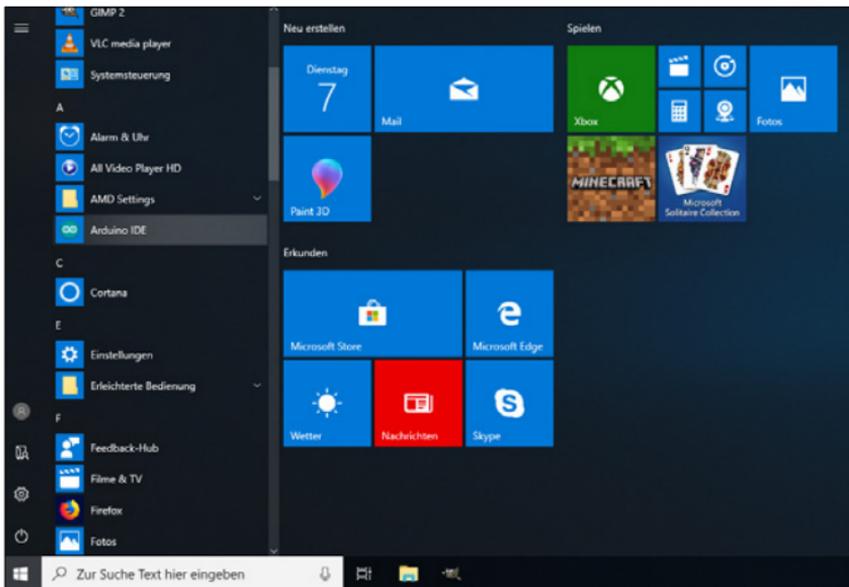


Abbildung 1.4: Die Arduino-App ist nun im Windows-10-Kachelmenü vorhanden.

1.3 Die Installation der Arduino IDE auf einem Mac-OSX-Rechner

1. Anwender von Mac OSX rufen zuerst einen Webbrowser auf und geben die Adresse
`https://www.arduino.cc/en/Main/Software`
ein. Wählen Sie die Mac-OSX-Version. Klicken Sie auf der nächsten Webseite auf *Just download*.
2. Der Browser öffnet nun ein Fenster und bietet Ihnen den Download der Mac-OSX-Installationsdatei an. Bestätigen Sie.
3. Öffnen Sie den Finder und begeben Sie sich in das Verzeichnis, in dem sich die heruntergeladene Datei befindet.
4. Mit einem Doppelklick wird die Zip-Datei entpackt.
5. Doppelklicken Sie auf die Arduino-Datei. So starten Sie das Programm. Eine Installation ist nicht notwendig.

6. Fügen Sie das Programm nach einem Start mit einem Doppelklick in das Dock ein. So müssen Sie es nicht erst im Finder unter *Programme* suchen, bevor Sie es starten können.

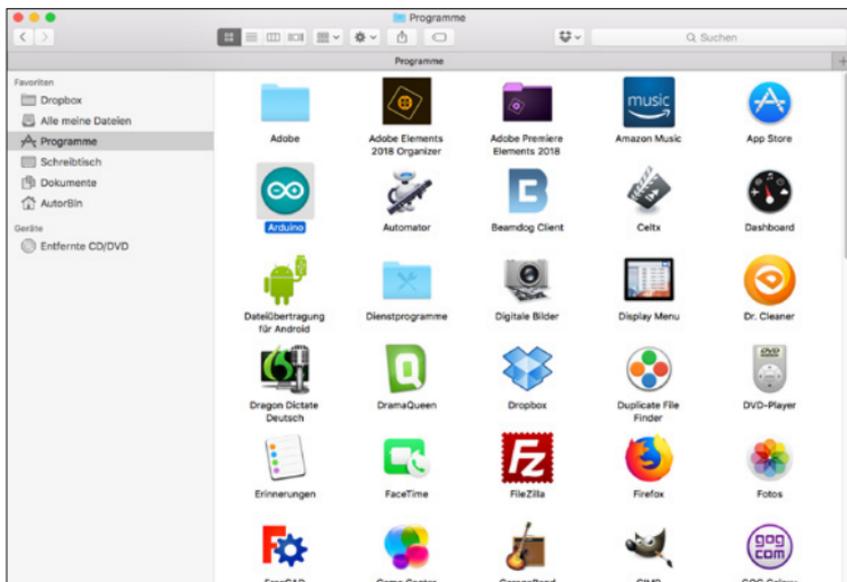


Abbildung 1.5: Die Arduino-App finden Sie auf einem Macintosh mit dem Finder (dem Macintosh-Dateimanager).

1.4 Die Installation der Arduino IDE auf einem Linux-Rechner

Die Arduino IDE können Linux Anwender über ihren Paketmanager installieren. So finden Anwender von Ubuntu die IDE im Ubuntu Software-Center und in der Synaptic-Paketverwaltung. Die notwendigen Pakete und Bibliotheken werden hier ebenfalls ausgewählt und installiert. Sie finden danach das Programm in Ihrem KDE- oder auch Gnome-Menü.

Natürlich finden Sie die Linux-Pakete auch unter <https://www.arduino.cc/en/Main/Software>.

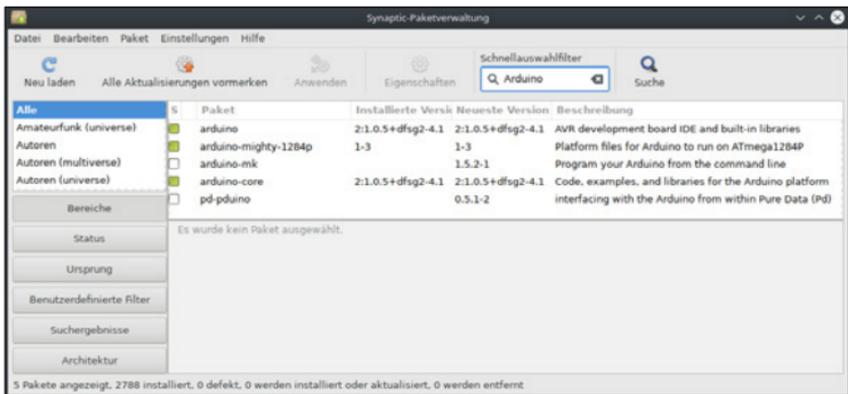


Abbildung 1.6: Die Arduino-Pakete finden Sie auch im Linux-Paketmanager (Synaptic).

1.5 Was sind Bibliotheken?

Bibliotheken enthalten vorprogrammierten Code. Er macht es einfacher, bestimmte Funktionen zur Verfügung zu stellen. Anstatt diese selbst mühselig programmieren zu müssen, verwenden Sie eine passende Bibliothek, binden diese in die IDE ein und greifen auf die in der Bibliothek vorhandenen Befehle und Funktionen zurück. Mit der Bibliothek *LiquidCrystal* können Sie beispielsweise ein LCD-Modul in einem Projekt verwenden. Es gibt eine große Anzahl an Bibliotheken für Arduino-Projekte. Einige stehen direkt über die Arduino IDE zur Verfügung. Sie müssen sie nur auswählen und installieren. Andere können Sie aus dem Internet laden. Beachten Sie bitte: Um eine Bibliothek verwenden zu können, müssen Sie diese zuerst auf Ihrem Rechner installieren.

Unter *Sketch / Bibliothek einbinden* finden Sie eine Liste der verfügbaren Bibliotheken. Mit einem Mausklick können Sie eine davon auswählen und installieren.

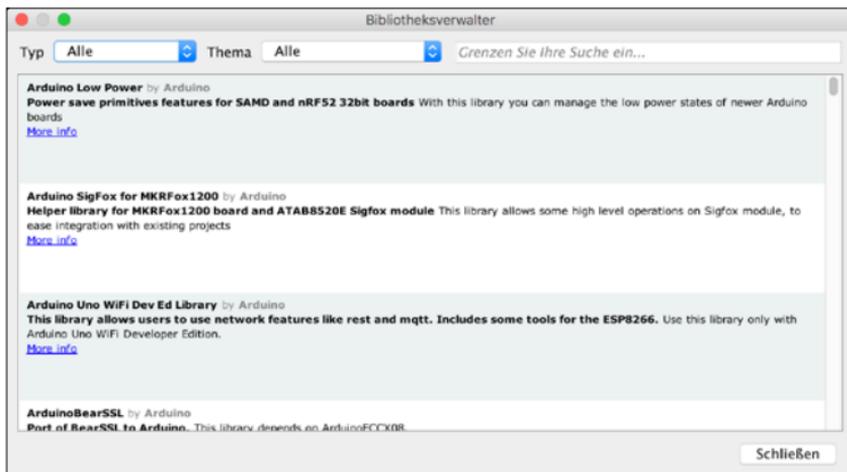


Abbildung 1.7: Erweitern Sie die IDE mit Bibliotheken.

Laden Sie eine Bibliothek aus dem Internet, wird diese in der Regel als Zip-Datei angeboten. Sie verwenden diese dann, indem Sie in der Arduino IDE *Sketch / Bibliothek einbinden / .ZIP-Bibliothek hinzufügen* wählen. Anschließend wählen Sie die Bibliothek aus und bestätigen.

Sie können danach die Bibliothek direkt in Ihren Sketch verwenden. Zu jeder Bibliothek gehören in der Regel auch Beispieldateien. Diese stehen jedoch erst nach einem Neustart der IDE zur Verfügung.

Das Programm selbst, das Sie erstellen und das eine bestimmte Aufgabe ausführen soll, heißt übrigens Sketch.

Die Arduino IDE benötigt eine installierte Java Runtime. Sie finden diese unter <https://java.com/de/download/>.

2. Ein erster Blick auf die Oberfläche der Arduino IDE

In diesem Kapitel möchte ich Ihnen die Arduino IDE vorstellen. Sie werden erfahren, wie Sie die IDE starten und welche Möglichkeiten diese in sich birgt. Sie werden den Aufbau der IDE kennenlernen, die Einstellungen und die Menüzeile.

2.1 Die Arduino IDE starten

Windows-Anwender finden die IDE im Windows-10-Kachelmenü. Klicken Sie auf *Arduino IDE*.

Über das Kontextmenü können Sie die Kachel *An „Start“ anheften*. So müssen Sie nicht erst durch das Menü blättern, sondern Sie klicken einfach auf die Arduino IDE Kachel.

Über *Mehr* können Sie die IDE *An Taskleiste anheften* und finden dann ein Symbol in der Windows-10-Startleiste.

Das ist auch ganz praktisch, genügt doch ein Mausklick auf das Bild in der Symbolleiste, um die IDE zu starten.

Natürlich können Sie auch ein Schnellstartsymbol auf dem Windows-10-Desktop anlegen und über dieses dann die IDE starten. Welchen Weg Sie gehen, um das Programm zu starten, bleibt Ihnen überlassen.

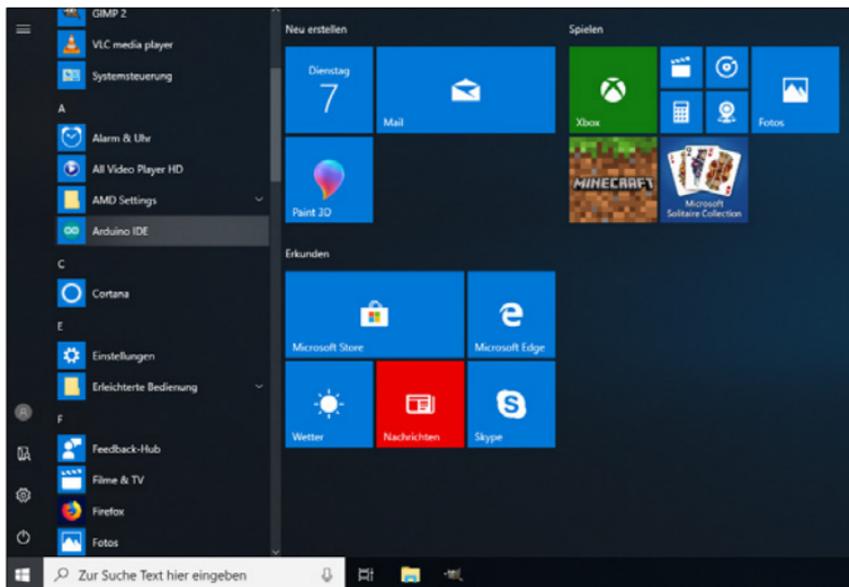


Abbildung 2.1: Im Windows-10-Kachelmenü finden Sie die Arduino IDE.

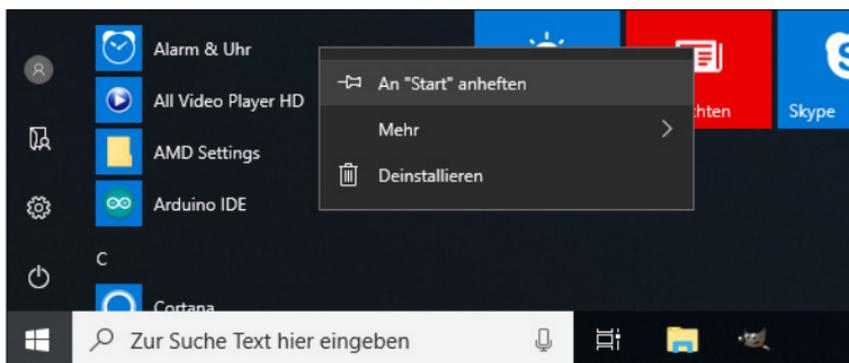


Abbildung 2.2: Über das Kontextmenü des Menüeintrags können Sie die Kachel „An Start anheften“.

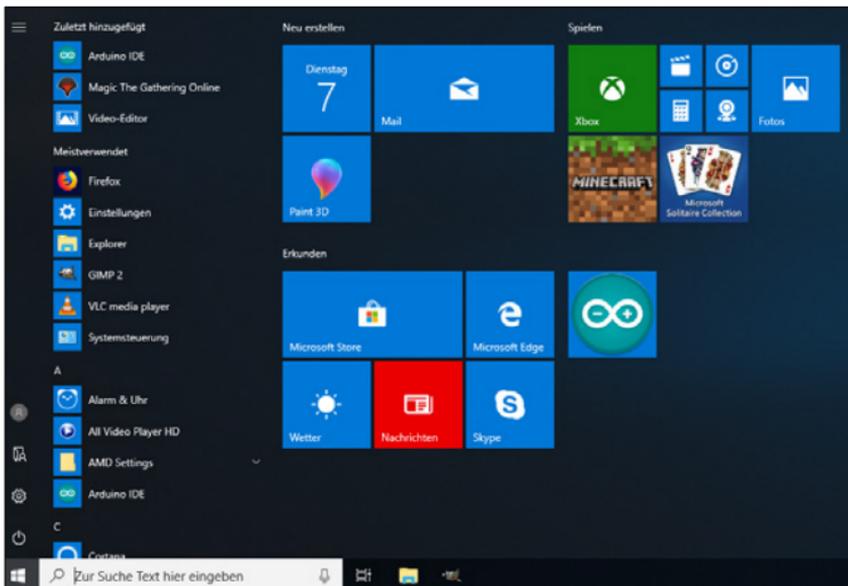


Abbildung 2.3: Tun Sie dies, finden Sie die Arduino-Kachel mit einer eigenen Kachel im Startmenü wieder.

Anwender von Mac OSX öffnen den Finder. Unter *Programme* finden Sie auch die Arduino IDE. Starten Sie das Programm mit einem Doppelklick.

Im Dock sehen Sie nun auch das Symbol der Arduino IDE. Markieren Sie es. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie *Optionen / Im Dock behalten*. So bleibt das Programmsymbol im Dock. Beim nächsten Mal starten Sie es von hier aus und müssen nicht erst den Finder öffnen.

Natürlich können Sie auch unter Mac OSX und Linux ein Schnellstartsymbol erstellen, auf dem Desktop ablegen und über dieses Ihr Programm starten.

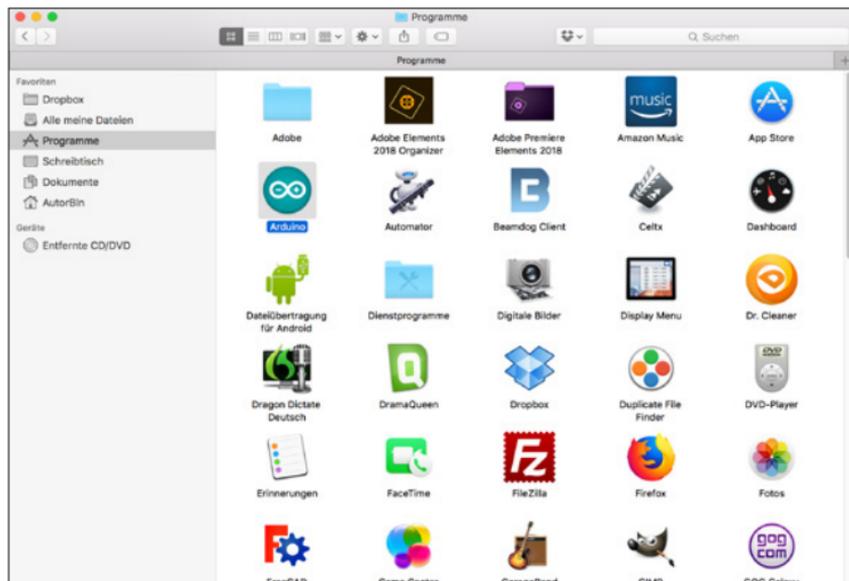


Abbildung 2.4: Die Arduino IDE finden Sie auf einem Macintosh-Rechner mit dem Finder. Sie befindet sich im Ordner Programme.

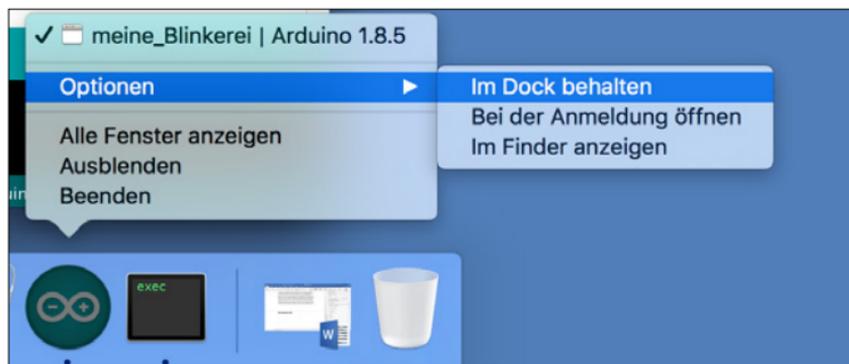


Abbildung 2.5: Behalten Sie das Symbol im Dock. Beim nächsten Programmstart müssen Sie nicht erst den Finder öffnen und das Programmsymbol suchen.

Linux-Anwender finden die IDE im KDE-Menü unter *Anwendungen / Entwicklung / Create physical computing projects*.

2.2 Ein erster Blick auf die Arduino IDE

IDE (*Integrated Development Environment*) steht übrigens für Integrierte Entwicklungsumgebung. So wird ein Programm bezeichnet, das bereits mehrere Tools und Funktionen für die Entwicklung von Software unter einer Oberfläche vereint.

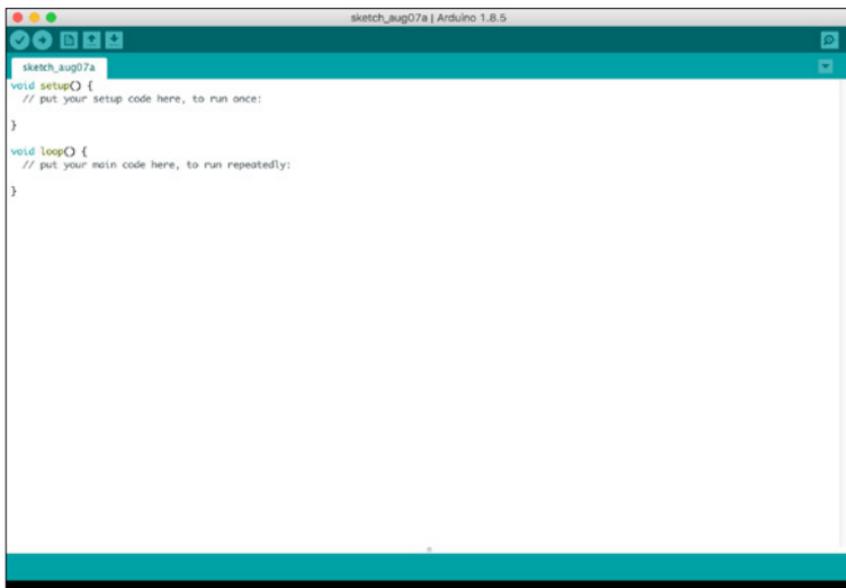


Abbildung 2.6: Die Arduino IDE nach dem Start

Den größten Bereich der IDE nimmt das Fenster ein, in dem Sie den Sketch schreiben. Zur Erinnerung: Der Sketch ist das Programm, das dem Board und seinen Bauteilen sagt, was es tun soll.

Das Beispielprogramm habe ich hier mit *Datei / Neu* entfernt. Nun ist die IDE bereit, einen neuen Code aufzunehmen. Dabei muss man sich an die Formatierung der Befehle, Variablen und Werte halten. Tut man dies nicht, versteht die Arduino IDE nicht, was sie tun soll. Man spricht bei dieser Schreibweise und Formatierung auch von einer „Syntax“.

Im Fenster der IDE sehen Sie schon den Beginn eines Codes. Der muss nur angepasst und ergänzt werden.

```
void setup() {  
    // put your setup code here, to run once:  
}  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

Der Kommentar sagt Ihnen, dass Sie hier den Hauptcode (main code) eintragen können. Dieser ist mit der geschweiften Klammer eingerahmt.

3. Ein kurzer Überblick über die wichtigsten Arduinoboards

In diesem Kapitel möchte ich Ihnen einen kleinen Überblick über die verschiedenen Arduinoboards geben. Sie erfahren zunächst, was ein Arduinoboard ist und wozu Sie es brauchen. Anschließend stelle ich Ihnen eine Auswahl interessanter Arduinoboards vor.

Die Liste der hier aufgeführten Arduinoboards ist keineswegs vollständig. Es kommen immer wieder neu entwickelte Boards mit interessanten Funktionen hinzu. Immer einmal werden Boards aktualisiert. Es gibt neue Versionen und manchmal auch erweiterte Boards mit neuen Bauteilen, neuen Funktionen und schnelleren Prozessoren. Um das Ganze noch etwas komplizierter zu machen, gibt es Nachbauten von anderen Firmen, Klone und eigene Entwicklungen. Ich gehe in diesem Buch auf die wichtigsten Arduinoboards ein, die Sie in den handelsüblichen Sets finden. Sie können sie auch einzeln erwerben. Die Original-Arduinoboards gibt es natürlich direkt bei Arduino. Jedoch tun es auch die nachgebauten Boards.

Übrigens: Sie müssen nicht für jedes Projekt ein neues Board kaufen. Die Bauelemente und Kabel werden aufgesteckt. So wird das ganze Projekt «zusammengebastelt». Die Elemente müssen nicht verlötet werden und können später für andere Projekte verwendet werden.

Nur wenn Sie ernsthaft an einem Projekt interessiert sind und es vielleicht auch erweitern wollen, empfehle ich Ihnen, die Bauteile und Kabelverbindungen zu verlöten. Gesteckte Verbindungen sind etwas friemelig und leicht zu lösen. Möchte man etwas länger nutzen, lohnt es sich, zum Lötkolben zu greifen.

Am Ende des Kapitels zeige ich Ihnen anhand des Arduino Uno, wie ein Arduinoboard aufgebaut ist und welche wichtigen Bauteile auf ihm vorhanden sind.

3.1 Was ist ein Arduinoboard und wozu brauche ich das?

Ein Arduinoboard ist ein Atmel-AVR-Mikrocontroller. Der AVR ist das Kernstück eines Arduinoboards. Neben diesem finden Sie auf dem Board viele andere Bau-

teile und Anschlussbuchsen. Dabei ist die Zusammenstellung je nach Board unterschiedlich.

Es gibt auch einige Ausnahmen, bei denen kein AVR-Mikrocontroller verwendet wird, sondern ein anderer. Beim Arduino Due wird zum Beispiel ein 32-Bit-Prozessor vom Typ ARM Cortex-M3 verwendet.

Alle Boards haben eines gemeinsam: Sie lassen sich leicht verwenden und sehr leicht mit der Arduino IDE programmieren. Letzteres geschieht über eine serielle Schnittstelle. Die Verbindung zu einem PC erfolgt hierbei über ein USB-Kabel.

Ein Bootloader ist bereits auf dem Arduinoboard vorhanden.

Jedes Arduinoboard verfügt über verschiedene Ein- und Ausgänge (Input- und Output-Pins). Diese kann man für verschiedenartige elektronische Schaltungen nutzen.

Darüber hinaus können einige Arduinoboards mit sogenannten Shields erweitert werden. Das sind Erweiterungsplatinen, die einfach auf das Board gesteckt werden.

Wozu braucht man ein Arduinoboard? Diese Frage ist gar nicht so leicht zu beantworten. Ich würde sagen, um ein cooles Projekt umzusetzen und etwas zu bauen, das man in Haus, Hof und Garten verwenden kann. Sie können von der Arduino-Community profitieren, Projekte nachbauen oder eigene Entwicklungen in das Internet setzen und mit anderen teilen. Sicher gibt es auch im Handel einen Automaten, der Ihre Blumen über eine Zeitschaltuhr gießen kann, wenn Sie im Urlaub sind. Aber es ist doch schon toll, wenn Sie so etwas selbst bauen. Darüber hinaus können Sie das genutzte Board und die Schaltung erweitern. Vielleicht möchten Sie über eine Webcam sehen, wie es Ihren Blumen geht? Vielleicht wollen Sie das Board über das Smartphone überwachen? Vielleicht möchten Sie eine eigene Wetterstation bauen, einen Lichtschalter, der auf Bewegung, Zuruf oder aufkommende Dunkelheit reagiert? Vielleicht wollen Sie ... Es gibt viele Möglichkeiten. Seien Sie kreativ und lassen Sie sich von den Projekten in diesem Buch und denen anderer Arduino-Fans inspirieren und mitreißen.