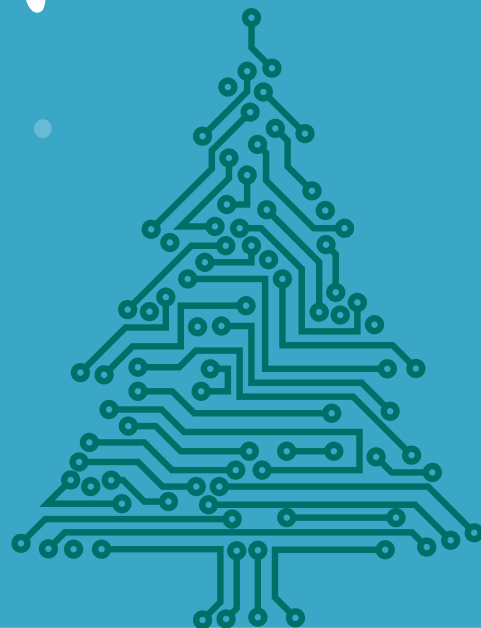
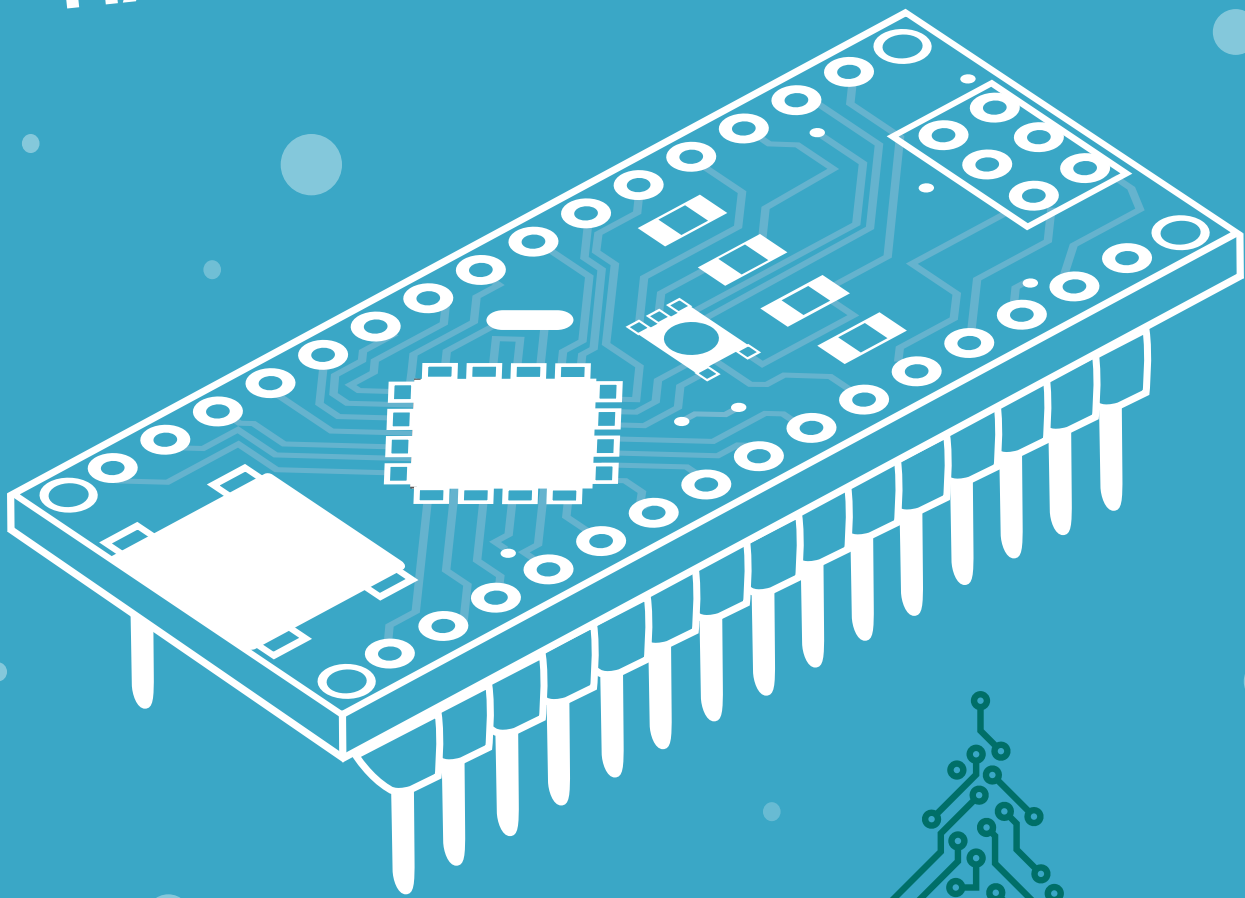


ADVENTSKALENDER FÜR

ARDUINO®

HANDBUCH



FRANZIS

BENÖTIGEN SIE HILFE ODER HABEN SIE FRAGEN ZU DIESEM ADVENTSKALENDER?

Unter **<https://www.franzis.de/faq-zu-den-adventskalendern>** finden Sie Antworten auf die häufigsten Fragen und Kontaktmöglichkeiten zu unserem Support-Team.

IMPRESSUM

© 2021 Franzis Verlag GmbH, Richard-Reitzner-Allee 2, D-85540 Haar, Germany
Änderungen, Innovationen und Druckfehler vorbehalten

www.franzis.de

2021/1

Autor: Christian Immler
Produktmanagement: Tobias Schärfl
Satz: PC-DTP-Satz und Informations GmbH
Copy Editor: Claudia Fliedner
Coverdesign: Thomas Preischl
GTIN: 4019631551030

Nicht für Kinder unter 14 Jahren geeignet.

Arduino® ist ein eingetragenes Markenzeichen der ARDUINO AG.

Alle in diesem Buch vorgestellten Schaltungen und Programme wurden mit der größtmöglichen Sorgfalt entwickelt, geprüft und getestet. Trotzdem können Fehler im Buch und in der Software nicht vollständig ausgeschlossen werden. Verlag und Autor haften in Fällen des Vorsatzes oder der groben Fahrlässigkeit nach den gesetzlichen Bestimmungen. Im Übrigen haften Verlag und Autor nur nach dem Produkthaftungsgesetz wegen der Verletzung des Lebens, des Körpers oder der Gesundheit oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht ein Fall der zwingenden Haftung nach dem Produkthaftungsgesetz gegeben ist.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

LIEBE KUNDEN



Dieses Produkt wurde in Übereinstimmung mit den geltenden europäischen Richtlinien hergestellt und trägt daher das CE-Zeichen. Der bestimmungsgemäße Gebrauch ist in der beiliegenden Anleitung beschrieben.

Bei jeder anderen Nutzung oder Veränderung des Produktes sind allein Sie für die Einhaltung der geltenden Regeln verantwortlich. Bauen Sie die Schaltungen deshalb genau so auf, wie es in der Anleitung beschrieben wird. Das Produkt darf nur zusammen mit dieser Anleitung weitergegeben werden.



Das Symbol der durchkreuzten Mülltonne bedeutet, dass dieses Produkt getrennt vom Hausmüll als Elektroschrott dem Recycling zugeführt werden muss. Wo Sie die nächstgelegene kostenlose Annahmestelle finden, sagt Ihnen Ihre kommunale Verwaltung.

WARNUNG! AUGENSCHUTZ UND LEDS:

Blicken Sie nicht aus geringer Entfernung direkt in eine LED, denn ein direkter Blick kann Netzhautschäden verursachen! Dies gilt besonders für helle LEDs im klaren Gehäuse sowie in besonderem Maße für Power-LEDs. Bei weißen, blauen, violetten und ultravioletten LEDs gibt die scheinbare Helligkeit einen falschen Eindruck von der tatsächlichen Gefahr für Ihre Augen. Besondere Vorsicht ist bei der Verwendung von Sammellinsen geboten. Betreiben Sie die LEDs so wie in der Anleitung vorgesehen, nicht aber mit größeren Strömen.

KURZSCHLÜSSE VERMEIDEN

Eine direkte Verbindung zwischen Minus- und Pluspol muss unbedingt vermieden werden, weil Drähte und Batterien heiß werden können und weil die Batterien sich dann schnell verbrauchen. Im Extremfall können Drähte glühend heiß werden, und die Batterie kann explodieren. Es besteht Brand- und Verletzungsgefahr. Verwenden Sie nach Möglichkeit nur normale Zink-Kohle-Batterien (6F20), die einen geringeren Kurzschlussstrom liefern und deshalb weniger gefährlich sind als Alkali-batterien (6RL61). Verwenden Sie keinesfalls Akkus.

Adventskalender für Arduino 2021

Das Programmieren von Mikrocontrollern war früher nur etwas für Ingenieure und Informatiker. Die Arduino-Plattform ermöglicht dank übersichtlicher Hardware und einfach zu verstehender Software auf einmal jedem den Einstieg in die Mikrocontrollertechnik.

Dieser Adventskalender bringt jeden Tag ein neues Programmierprojekt für das eigene kleine Smart Home, das bis Weihnachten in vollem Glanz erstrahlen wird.

Der Name Arduino

Der Arduino kommt aus Italien und wurde nach dem italienischen König Arduino benannt, der bis ins Jahr 1005 in Ivrea, dem Firmensitz des Arduino-Herstellers herrschte. Nach König Arduino ist dort heute die Lieblingsbar der Arduino-Entwickler Massimo Banzi und David Cuartielles benannt.

Vorsichtsmaßnahmen

Auf keinen Fall sollte man irgendwelche Arduino-Pins miteinander verbinden und abwarten, was passiert.

Nicht alle Arduino-Pins lassen sich frei programmieren. Einige sind für die Stromversorgung und andere Zwecke fest eingerichtet.

Einige Arduino-Pins sind direkt mit Anschlüssen des Mikrocontrollers verbunden, ein Kurzschluss kann den Arduino komplett zerstören – zumindest theoretisch. Die Arduino-kompatiblen Platinen sind erstaunlich stabil gegen Schaltungsfehler. Verbindet man über eine LED zwei Pins miteinander, muss immer ein Vorwiderstand dazwischengeschaltet werden, wenn in der LED nicht bereits einer eingebaut ist.

Für Logiksignale benötigen einige Arduino-kompatible Platinen 3,3 V, andere 5 V. Das Arduino-kompatible Nano-Board in diesem Adventskalender verwendet ein +5V-Signal als logisch *high* bzw. *wahr*.

Heute im Adventskalender

- Nano-Board (Arduino-kompatibel)

Haus backen

Auf der Rückseite des Adventskalenders finden Sie die Ausschneidevorlagen für das Lebkuchenhaus. Hier können Sie schon erkennen, wie das Haus aussehen wird. Heute bereiten wir erst einmal den Teig vor.

Für den Teig brauchen Sie:

800 g	Mehl
400 g	Honig
150 g	Butter
200 g	Zucker
20 g	Kakao
2	Eier
1 Tüte	Lebkuchengewürz
1 Prise	Salz
4 TL	Backpulver

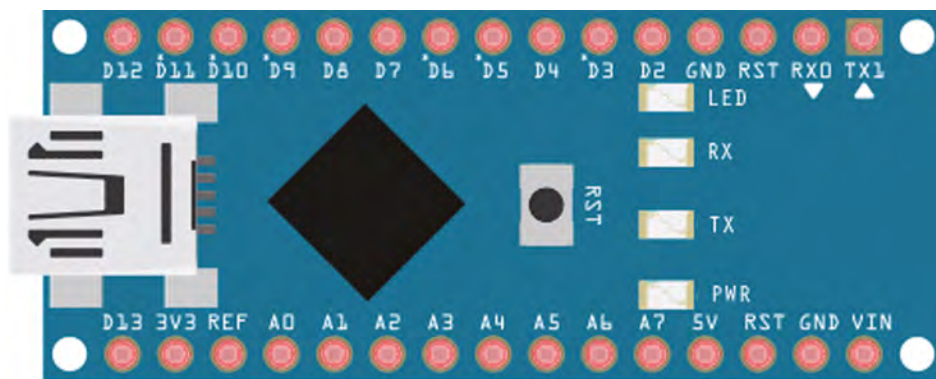
Schmelzen Sie die Butter in einem Topf auf kleiner Flamme. Sie darf nicht anbrennen. Rühren Sie dabei den Honig und den Zucker mit ein, sodass es eine gleichmäßige Masse ergibt.

Lassen Sie die Masse abkühlen und rühren Sie die Eier, das Lebkuchengewürz und das Salz hinein.

Mischen Sie das Mehl (lassen Sie ein bisschen übrig, um am nächsten Tag die Arbeitsfläche damit zu bestreuen), das Backpulver und den Kakao und geben Sie alles nach und nach unter Rühren zu der Masse. Dabei entsteht ein immer festerer Teig, den Sie zum Schluss mit der Hand kneten können. Kneten Sie so lange, bis alles gleichmäßig gemischt ist, und lassen Sie den Teig dann in einer mit einem Tuch abgedeckten Schüssel über Nacht ruhen.

Nano-Board – Arduino-kompatible Platine

Die Arduino-Plattform bietet mittlerweile eine große Vielfalt an Platinen für unterschiedliche Anwendungszwecke. Dieser Adventskalender enthält eine zum Arduino Nano Standard kompatible Platine, die direkt auf ein Steckbrett gesteckt werden kann, um weitere Elektronik anzuschließen.



Die Anschlüsse am Nano

Die mit D2...D12 beschrifteten Pins sind digitale Ein- oder Ausgänge, die die Logikwerte 0 oder 1 – oder anders geschrieben *Low* oder *High* annehmen können. Damit lassen sich zum Beispiel LEDs ein- und ausschalten.

Die mit A0...A7 beschrifteten Pins sind analoge Eingänge. Dort kann eine Spannung zwischen 0V und 5V angelegt werden. Der Nano wertet sie aus und kann darauf mit bestimmten Aktionen antworten.

Was braucht man noch?

In unserem Adventskalender sind bereits alle elektronischen Bauteile enthalten, die Sie zum Aufbau der Experimente benötigen.

PC

Zur Programmierung des Nano verwenden wir in den Experimenten einen PC mit Windows. Die Arduino-IDE wird außer für Windows (ab Windows 7) auch für Linux, macOS und sogar für den Raspberry Pi angeboten, die anderen Tools teilweise nur für Windows. Die neue Arduino IDE 2.0 läuft nur auf Windows 10 (64 Bit), Linux (64 Bit) und macOS ab Version 10.14.

USB-Kabel

Die Verbindung zwischen PC und Nano erfolgt über ein Micro-USB-Kabel. Sie brauchen sich nicht extra ein solches Kabel zu besorgen, die meisten nicht mehr ganz aktuellen Smartphones verwenden diesen Steckertyp.

Netzteil (optional)

So lange der Nano am PC angeschlossen ist, wird er auch über die USB-Schnittstelle mit Strom versorgt. Nachdem ein Programm auf dem Nano gespeichert wurde, kann er in vielen Fällen auch ohne PC laufen. Dazu können Sie entweder ein USB-Steckernetzteil oder eine Powerbank anschließen. Ein Spannungsregler auf dem Nano regelt die Stromversorgung dann automatisch.

USB-Anschluss

Schließen Sie den Nano nach Möglichkeit an einen USB-2.0-Anschluss des PCs an, da es an USB-3.0-Anschlüssen eher zu Verbindungsproblemen kommen kann und es bei USB-3.0-Anschlüssen, bedingt durch die höhere Stromstärke, öfter zur Überhitzung des Nano kommt. Zur besseren Unterscheidung sind USB-3.0-Anschlüsse meistens blau.

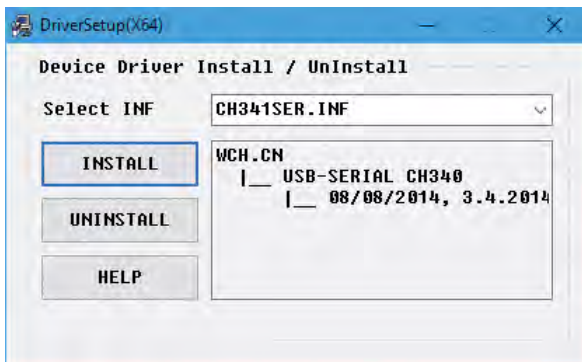
Nano ausschalten

Der Nano hat keinen Ausschalter, Sie brauchen einfach nur den Stecker ziehen, und er schaltet sich ab. Beim nächsten Einschalten startet automatisch der zuletzt gespeicherte Sketch. So werden auf der Arduino-Plattform die Programme genannt. Das gleiche passiert, wenn man den Reset-Taster drückt.

Softwareinstallation in Kürze

- 1 Laden Sie sich die Beispielprogramme und den Gerätetreiber bei www.franzis.de/adventskalender/adventskalender-fuer-arduino herunter.
- 2 Entpacken Sie das ZIP-Archiv in einen beliebigen Ordner auf der Festplatte.
- 3 Schließen Sie den Nano über das USB-Kabel an und starten Sie dann mit der Datei `CH341SER.EXE` aus dem Unterverzeichnis `Treiber` die Treiberinstallation. Zur Installation sind Administratorrechte erforderlich. Sie müssen eine Anfrage der Windows-Benutzerkontensteuerung bestätigen.
- 4 Klicken Sie im Installationsdialog auf **INSTALL** und warten Sie, bis die Bestätigung erscheint, dass der Treiber installiert wurde. Sollte ein Fehler angezeigt werden, beenden Sie Ihren Virens Scanner, wenn Sie einen externen verwenden. Der Windows Defender hat kein Problem mit diesem Treiber.

Arduino-Adventskalender

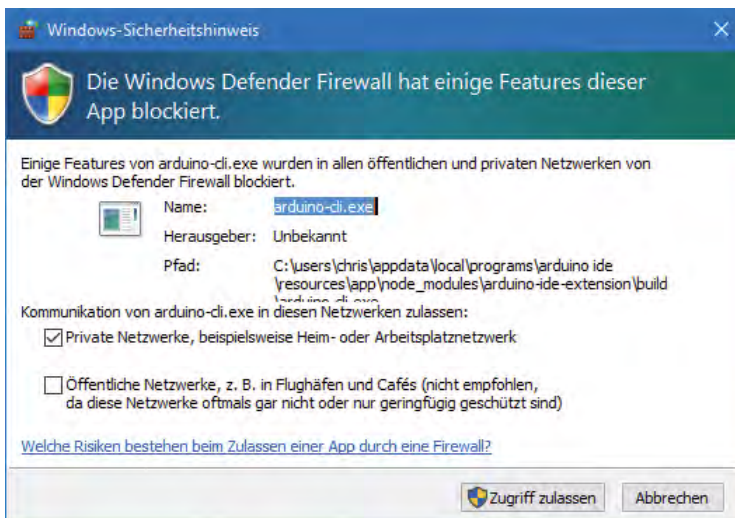


Installation des Gerätetreibers.

- 5 Für die Programmierung Arduino-kompatibler Platinen liefert Arduino eine Entwicklungsumgebung (IDE), in der man die Programme, die bei Arduino als Sketch bezeichnet werden, in einer C-ähnlichen Programmiersprache schreiben kann. Laden Sie sich die Arduino IDE bei www.arduino.cc/en/software herunter und installieren Sie sie. Sie finden die Installationsdatei auch im Download zum Adventskalender. Unter Windows 10 können Sie die Arduino IDE auch aus dem Windows Store herunterladen und installieren. Allerdings finden Sie dort meistens nicht die neueste Version.

Erster Start der Arduino IDE und Einrichten des Nano-Boards

- 1 Beim ersten Start der Arduino IDE erscheint eine Meldung der Windows-Firewall. Der Zugriff muss zugelassen werden, da sonst keine Verbindung zum angeschlossenen Nano-Board möglich ist. Externe Firewalls sollten bei Problemen komplett beendet werden, da die Arduino IDE in vielen Fällen keine Verbindung herstellen kann, selbst wenn sie in der Firewall als Ausnahme eingetragen ist.



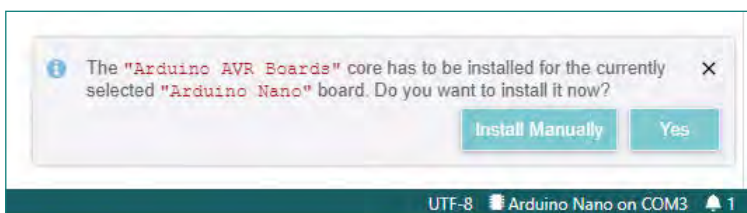
Arduino IDE in der Firewall freigeben

- 2 Klicken Sie oben auf das Listenfeld *no board selected*. Auf den meisten PCs werden dort ein oder zwei Zeilen *Unknown at COMx* angezeigt. Üblicherweise belegt der Nano-Treiber die COM-Schnittstelle mit der höchsten Nummer. Wählen Sie diese aus.



COM-Schnittstelle wählen

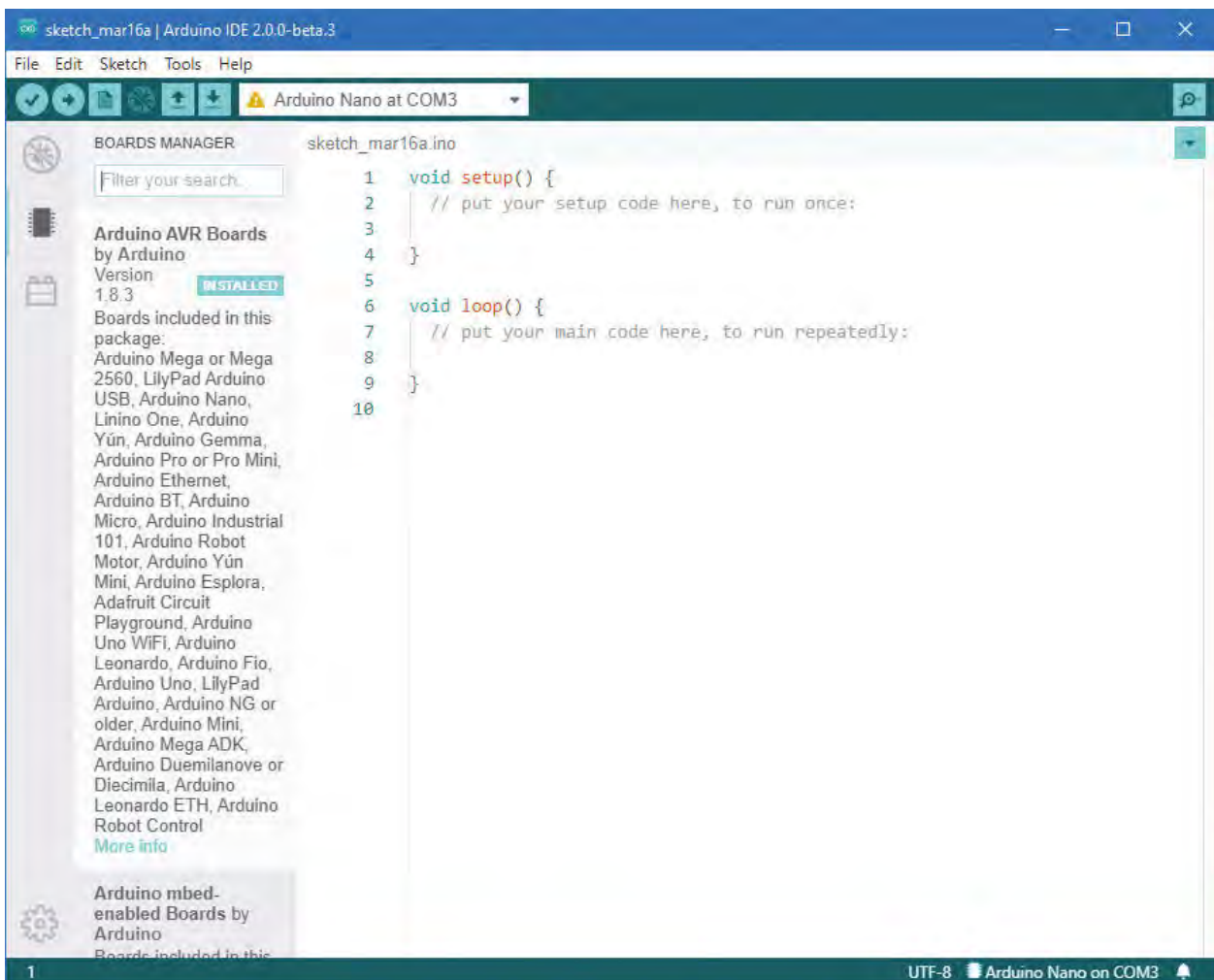
- 3 Wählen Sie im nächsten Dialogfeld in der Liste *Boards* den *Arduino Nano*. Wundern Sie sich nicht, dass im Bereich *Ports* die Meldung *No Ports discovered* auftaucht. Sie haben den Anschluss ja bereits gewählt.
- 4 Beim ersten Auswählen eines Arduino-kompatiblen Boards muss noch ein passendes Zusatzmodul installiert werden. Wenn die Meldung zur Installation des *Arduino AVR Boards core* erscheint, klicken Sie auf *Yes*.



Arduino AVR Boards core installieren

- 5 Während der Installation erscheint der *Boards Manager* in der Arduino IDE. Warten Sie bis das grüne Symbol *INSTALLED* erscheint.

Arduino-Adventskalender



Boards Manager in der Arduino IDE

- 6 Nachdem *Arduino AVR Boards core* installiert ist, können Sie den Boards Manager mit dem Symbol in der linken Symbolleiste wieder schließen.
- 7 Prüfen Sie noch, ob im Menü *Tools/Processor* die Option *ATmega 328P (Old Bootloader)* ausgewählt ist. Ist das nicht der Fall, wählen Sie diese Option aus.
- 8 Jetzt erscheint der Nano in der Auswahlliste am oberen Rand der Arduino IDE. Wenn Sie diese an einem anderen Tag starten und der Nano angeschlossen ist, können Sie ihn direkt dort auswählen.
- 9 Die Programme der einzelnen Tage finden Sie Arduino-typisch in einzelnen Ordnern im Download. Kopieren Sie diese Ordner in den Ordner *Dokumente\Arduino* in Ihrem Windows-Benutzerprofil oder in *OneDrive\Dokumente\Arduino*, wenn die automatische Synchronisation mit OneDrive eingerichtet ist. Dieser Ordner wird beim ersten Start der Arduino IDE angelegt. Ist er nicht vorhanden, finden Sie den genauen Pfad in der Arduino IDE über den Menüpunkt *File/Preferences* im Feld *Sketchbook location*.

LED blinkt

In einem ganz einfachen Programm soll die auf dem Nano eingebaute LED schnell blinken.

Bauteile: Nano-Board

```
//01blink
//Franzis Adventskalender für Arduino

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(50);
  digitalWrite(LED_BUILTIN, LOW);
  delay(50);
}
```

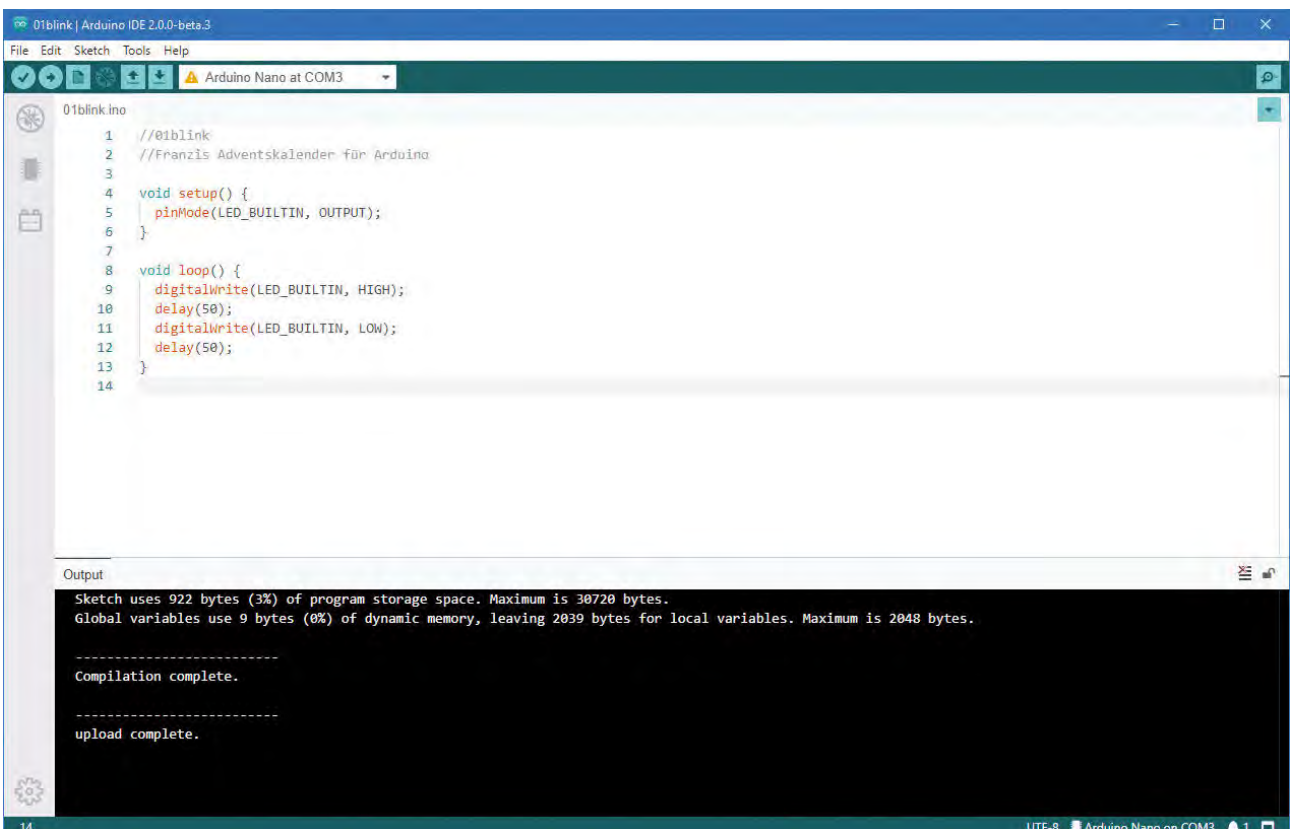
Die ersten beiden Zeilen, die mit // beginnen, sind nur Kommentare und haben für die Funktion des Programms keine Bedeutung. Sie stehen in den Programmen des Adventskalenders nur, um die abgedruckten Beispiele den heruntergeladenen Programmen einfacher zuordnen zu können. Wenn Sie die Programme selber schreiben, können Sie diese Kommentarzeilen weglassen.



Klicken Sie in der Arduino IDE auf das Symbol *New* und schreiben Sie das abgebildete Programm.



Sie finden das Programm `01blink` auch in der Arduino IDE, wenn Sie die Dateien aus dem Download wie beschrieben in den Arduino-Ordner kopiert haben. Klicken Sie dazu auf das Symbol *Open*.



Das Programm `01blink` in der Arduino IDE

Farbige Syntaxhervorhebung erleichtert es, bestimmte Programmelemente zu erkennen.

Programme auf den Nano übertragen



Jetzt muss das fertige Programm auf den Nano übertragen werden. Klicken Sie dazu auf das Symbol *Upload*. Im unteren Teilfenster der Arduino IDE werden ein paar Statusmeldungen angezeigt. Nachdem das Hochladen abgeschlossen ist, erscheint hier *upload complete*. Die LED auf dem Nano blinkt. Das Programm braucht also nicht extra gestartet zu werden.

So funktioniert das Programm

Alle Programme in der Arduino IDE bestehen aus mindestens zwei Funktionen. Die Funktion `void setup()` läuft ein einziges Mal beim Start und wird meistens zur Einrichtung der Pins verwendet. Selbst wenn das nicht nötig ist, bleibt sie einfach leer, muss aber im Programm vorhanden sein.

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

In diesem Programm wird die eingebaute LED, die über die Konstante `LED_BUILTIN` vordefiniert ist, aber auch über den Pin 13 angesprochen werden kann, als Ausgang definiert. Pins, an denen LEDs angeschlossen sind, sind immer Ausgänge.

Die Funktion `void loop()` wird immer wieder wiederholt, bis man die Stromversorgung trennt oder den Reset-Knopf drückt.

```
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(50);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(50);  
}
```

In der Digitalelektronik gibt es zwei Schaltzustände, `HIGH` und `LOW`. `HIGH` steht für die Betriebsspannung, beim Nano sind das +5 V, `LOW` bedeutet ein Massesignal mit 0 V.

Die Funktion `digitalWrite()` setzt einen digitalen Anschlusspin, in unserem Fall die eingebaute LED, auf `HIGH`-Signal und schaltet sie damit ein. Der erste Parameter dieser Funktion bezeichnet immer den Pin, der zweite das Signal, das dort ausgegeben werden soll.

Anschließend wartet das Programm 50 Millisekunden, bevor ein zweiter Aufruf der `digitalWrite()`-Funktion das Signal `LOW` auf der eingebauten LED ausgibt und diese damit ausschaltet.

Nach weiteren 50 Millisekunden starte die Endlosschleife neu, sodass die LED ständig blinkt.

Tag 2

Heute im Adventskalender

- Steckbrett
- LED rot mit Vorwiderstand

Haus zusammenbauen

Nachdem der Teig über Nacht geruht hat, rollen Sie ihn auf einer mit Mehl bestreuten Arbeitsfläche etwa 1 cm dick aus. Schneiden Sie dann die Vorlagen für die Teile des Hauses aus der Rückseite dieses Adventskalenders aus, legen Sie sie auf die Lebkuchenplatte und schneiden Sie die Teile aus. Beachten Sie dabei die Hinweise auf der Rückseite. Sie brauchen 1x Bodenplatte, 2x Dach, 1x Vorderseite, 1x Rückseite. Die Bodenplatte kann natürlich auch größer sein, wenn Sie um das Haus herum noch einen Garten gestalten möchten.

Aus dem übrigen Teig können Sie Bäume oder andere Figuren ausstechen.



Legen Sie die Teile vorsichtig auf ein Backblech mit Backpapier und backen Sie sie bei 175° etwa 10 bis 15 Minuten lang. Lassen Sie die Teile nach dem Backen über Nacht gut abkühlen.

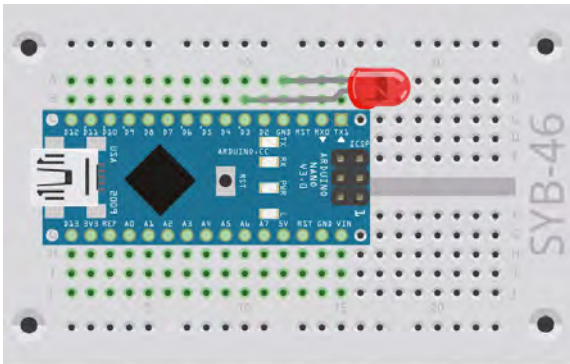


Arduino-Adventskalender

Zwei LEDs blinken im Wechsel

Die eingebaute LED soll mit einer extern angeschlossenen LED im Wechsel blinken.

Bauteile: Nano-Board, Steckbrett, LED rot mit Vorwiderstand



fritzing LED am Pin D3

Die Anode der LED (langer Draht) ist am Pin D3 angeschlossen, die Kathode (kurzer Draht) am Pin GND.

```
//02blink
//Franzis Adventskalender für Arduino

int led1 = 3;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(led1, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  digitalWrite(led1, LOW);
  delay(200);
  digitalWrite(LED_BUILTIN, LOW);
  digitalWrite(led1, HIGH);
  delay(200);
}
```

So funktioniert das Programm

Am Anfang des Programms wird die zusätzlich verwendete Pinnummer in einer Variablen vom Typ `Integer` gespeichert. Da die Pinnummer dann nur noch einmal im Programm auftaucht, braucht es nur leicht geändert zu werden, wenn die LED an einen anderen Pin angeschlossen werden soll.

```
int led1 = 3;
```


Die eingebaute LED ist über die Konstante `LED_BUILTIN` automatisch ansprechbar, ohne dass man sie eigens definieren muss. Die Funktion `void setup()` initialisiert beide LEDs als Ausgänge.

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
  pinMode(led1, OUTPUT);  
}
```

Die Funktion `void loop()` schaltet in jedem Durchlauf die eingebaute LED ein und die LED am Pin D3 aus und wartet dann 200 Millisekunden. Danach wird die LED am Pin D3 aus- und die eingebaute LED eingeschaltet. Nach einer weiteren Wartezeit von 200 Millisekunden startet die `void loop()`-Schleife erneut. Die Wartezeit wurde gegenüber dem Programm von gestern verlängert, damit das wechselnde Blinken deutlicher zu erkennen ist.

```
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  digitalWrite(led1, LOW);  
  delay(200);  
  digitalWrite(LED_BUILTIN, LOW);  
  digitalWrite(led1, HIGH);  
  delay(200);  
}
```

Heute im Adventskalender

● Schaltdraht

Schaltdraht

Heute ist Schaltdraht im Adventskalender enthalten. Damit stellen Sie kurze Verbindungsbrücken her, mit denen Kontaktreihen auf der Steckplatine verbunden werden. Schneiden Sie den Draht mit einem kleinen Seitenschneider je nach Experiment auf die passenden Längen ab. Um die Drähte besser in die Steckplatine stecken zu können, empfiehlt es sich, sie leicht schräg abzuschneiden, sodass eine Art Keil entsteht. Entfernen Sie an beiden Enden auf einer Länge von etwa einem halben Zentimeter die Isolierung.

Haus verzieren

Kleben Sie die Teile des Hauses mit Zuckerguss zusammen. Für den Zuckerguss brauchen Sie:

1 Eiweiß
100 g Puderzucker

Schlagen Sie das Eiweiß schaumig und geben Sie nach und nach den Puderzucker dazu, bis eine gleichmäßige klebrige Masse entsteht. Damit kleben Sie die Teile zusammen. Lassen Sie oben am Dachfirst einen Spalt frei, damit Sie LEDs oder Drähte durchstecken können.



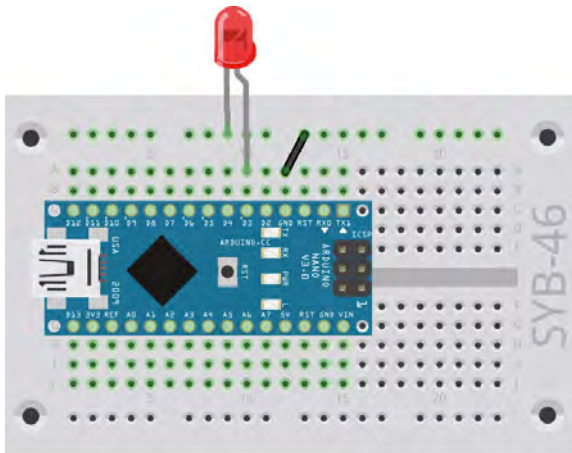


Arduino-Adventskalender

LED blinkt zufällig

Eine LED soll unregelmäßig blinken, wobei das Blinkintervall jedes Mal zufällig verändert wird.

Bauteile: Nano-Board, Steckbrett, LED rot mit Vorwiderstand, Drahtbrücke



LED am Pin D3 und an der Masseleiste

Die Schaltung von heute zeigt den typischen Schaltungsaufbau auf dem Steckbrett. Eine der horizontalen Kontaktleisten wird als Masseleitung verwendet, die über eine Drahtbrücke mit dem GND-Pin auf dem Nano verbunden ist. Achten Sie beim Aufbau der Schaltung darauf, dass die Kathode (kurzer Draht) der LED in der Masseleiste steckt, die Anode (langer Draht) ist in dieser Schaltung wieder mit dem Pin 3 verbunden.

```
//03blink
//Franzis Adventskalender für Arduino

int led1 = 3;

void setup() {
  pinMode(led1, OUTPUT);
}

void loop() {
  digitalWrite(led1, HIGH);
  delay(random(20, 200));
  digitalWrite(led1, LOW);
  delay(random(20, 200));
}
```

So funktioniert das Programm

Prinzipiell funktioniert das Programm ähnlich wie die Programme der letzten beiden Tage. Der wesentliche Unterschied ist, dass die Wartezeit zwischen zwei Schaltvorgängen jedes Mal zufällig festgelegt wird.

```
delay(random(20, 200));
```

In der Funktion `delay()` ist keine feste Zeit eingetragen, sondern eine weitere Funktion. Die Funktion `random()` ermittelt eine Zufallszahl. Der erste Parameter gibt die kleinste mögliche Zahl an, der zweite Parameter ist um 1 größer als die größte mögliche Zahl. In diesem Programm werden demnach Zahlen zwischen 20 und 199 erzeugt. Die Pause zwischen zwei Schaltvorgängen ist dann zufällig zwischen 20 und 199 Millisekunden lang.

Wie entstehen Zufallszahlen?

Gemeinhin denkt man, in einem Programm könne nichts zufällig geschehen – wie also kann ein Programm dann in der Lage sein, zufällige Zahlen zu generieren? Teilt man eine große Primzahl durch irgendeinen Wert, ergeben sich ab der x-ten Nachkommastelle Zahlen, die kaum noch vorhersehbar sind. Sie ändern sich auch ohne jede Regelmäßigkeit, wenn man den Divisor regelmäßig erhöht. Dieses Ergebnis ist zwar scheinbar zufällig, lässt sich aber durch ein identisches Programm oder den mehrfachen Aufruf des gleichen Programms jederzeit reproduzieren. Nimmt man aber eine aus einigen dieser Ziffern zusammengebaute Zahl und teilt sie wiederum durch eine Zahl, die sich aus der aktuellen Uhrzeitsekunde oder dem Inhalt einer beliebigen Speicherstelle des Computers ergibt, kommt ein Ergebnis heraus, das sich nicht reproduzieren lässt und daher als Zufallszahl bezeichnet wird.

FRANZIS

© 2021 FRANZIS Verlag GmbH, Richard-Reitzner-Allee 2, D-85540 Haar, Germany
2021/01



WEEE-Reg.-Nr.
DE21445697

GTIN 4019631551108



4 019631 551108