

CE

CONRAD

Alle Versuche im Überblick

Arduino-Adventskalender 2018	3	12. Tag	21
1. Tag	4	Heute im Adventskalender	21
Heute im Adventskalender	4	RGB-Farbmischung mit PWM	21
Nano-Board - Arduino-kompatible Platine	4	LEDs anschließen	21
Den Nano vorbereiten	4	So funktioniert das Programm	21
Softwareinstallation in Kürze	4	13. Tag	22
mBlock 3	4	Heute im Adventskalender	22
LED blinkt	5	Lauflicht in zwei Richtungen	22
So funktioniert das Programm	6	Das Programm	22
Programm auf den Nano übertragen	6	So funktioniert das Programm	22
2. Tag	7	14. Tag	23
Heute im Adventskalender	7	Heute im Adventskalender	23
Steckbrett	7	Potentiometer	23
LEDs	7	Lauflicht mit Potentiometer steuern	23
Wechselblinklicht	7	Das Programm	23
Das Programm	8	So funktioniert das Programm	23
So funktioniert das Programm	8	15. Tag	24
3. Tag	9	Heute im Adventskalender	24
Heute im Adventskalender	9	Sensorkontakt aus Knete	24
Schaltdraht	9	Das Programm	24
LEDs blinken mit einstellbarer Geschwindigkeit	9	So funktioniert das Programm	24
Das Programm	9	16. Tag	25
So funktioniert das Programm	9	Heute im Adventskalender	25
4. Tag	10	RGB-Farbmischung mit PWM	25
Heute im Adventskalender	10	Das Programm	25
LEDs blinken immer schneller	10	So funktioniert das Programm	25
Das Programm	10	17. Tag	26
So funktioniert das Programm	10	Heute im Adventskalender	26
5. Tag	11	RGB-Farbspektrum	26
Heute im Adventskalender	11	Das Programm	26
Widerstände und ihre Farbcodes	11	So funktioniert das Programm	26
Taster	11	18. Tag	27
LEDs mit Taster schalten	11	Heute im Adventskalender	27
Das Programm	12	LED-Würfel	27
So funktioniert das Programm	12	Das Programm	27
6. Tag	13	So funktioniert das Programm	27
Heute im Adventskalender	13	19. Tag	28
LEDs auf dem Engelsflügel umschalten	13	Heute im Adventskalender	28
Das Programm	13	LED-Würfel mit realistischem Würfelleffekt	28
So funktioniert das Programm	13	Das Programm	28
7. Tag	14	So funktioniert das Programm	28
Heute im Adventskalender	14	20. Tag	29
LEDs blinken zufällig	14	Heute im Adventskalender	29
Das Programm	14	Analoge Pegelanzeige mit LEDs	29
So funktioniert das Programm	14	Das Programm	29
Verbesserte Version des Programms	14	So funktioniert das Programm	29
8. Tag	16	21. Tag	30
Heute im Adventskalender	16	Heute im Adventskalender	30
LEDs dimmen	16	Lichteffekte mit Sensorkontakten steuern	30
Das Programm	16	Das Programm	30
So funktioniert das Programm	16	So funktioniert das Programm	30
9. Tag	17	22. Tag	32
Heute im Adventskalender	17	Heute im Adventskalender	32
Lauflicht auf dem Engelsflügel	17	Mehr LEDs als Arduino-Pins	32
Das Programm	17	Das Programm	32
So funktioniert das Programm	17	So funktioniert das Programm	32
10. Tag	18	23. Tag	33
Heute im Adventskalender	18	Heute im Adventskalender	33
Batteriekasten	18	Fototransistor	33
Lauflicht mit wechselnder Geschwindigkeit	18	Weihnachtliche Lichteffekte in der Dunkelheit	33
Das Programm	18	Das Programm	33
So funktioniert das Programm	18	So funktioniert das Programm	34
11. Tag	19	24. Tag	35
Heute im Adventskalender	19	Heute im Adventskalender	35
RGB-LEDs	19	Piezo-Summer	35
Verschiedene Farben auf einer RGB-LED	19	Weihnachtsbeleuchtung mit Musik	35
Das Programm	19	Das Programm	35
So funktioniert das Programm	20	So funktioniert das Programm	35

Arduino-Adventskalender 2018

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.

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-Platinen sind erstaunlich stabil gegen Schaltungsfehler. Verbindet man über eine LED zwei Pins miteinander, muss immer ein Vorwiderstand dazwischengeschaltet werden.

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

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.

1. Tag

1. Tag

Heute im Adventskalender

- Nano-Board (Arduino-kompatible Platine)

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. Jeden Tag wird im Adventskalender ein Hardwareexperiment mit zugehörigem Programm vorgestellt.

Die Experimente in diesem Adventskalender werden mit mBlock programmiert. Diese Programmiersprache basiert auf Scratch, einer der am leichtesten erlernbaren Programmiersprachen überhaupt. Die verwendeten Programme gibt es hier zum Download: bit.ly/c-adventskalender-arduino-18. Entpacken Sie das Zip-Archiv in ein Verzeichnis auf Ihrer Festplatte.

Den Nano vorbereiten

Um den Nano in Betrieb zu nehmen, braucht man:

- PC mit Windows
- MicroUSB-Kabel
- Treiber

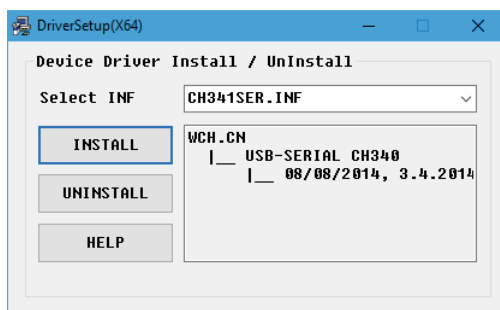
Die Verbindung zwischen PC und Nano erfolgt über ein MicroUSB-Kabel. Sie brauchen sich nicht extra ein solches Kabel zu besorgen, die meisten Smartphones verwenden diesen Steckertyp. Das Kabel wird gleichzeitig zur Stromversorgung und zur Datenübertragung verwendet.

Schließen Sie das Kabel nach Möglichkeit an einem USB-2.0-Anschluss Ihres PCs an, da es an USB-3.0-Anschlüssen eher zu Verbindungsproblemen kommen kann. Zur besseren Unterscheidung sind USB-3.0-Anschlüsse meistens blau.

Softwareinstallation in Kürze

Hier die Treiberinstallation in vier Schritten:

- 1 Entpacken Sie das ZIP-Archiv mit der heruntergeladenen Software in einen beliebigen Ordner unterhalb Ihres Windows-Benutzerordners.
- 2 Entpacken Sie das ZIP-Archiv `67006-9-nano-board-Treiber.zip` in einen eigenen Ordner.
- 3 Schließen Sie den Nano über das USB-Kabel an und starten Sie dann die Treiberinstallation mit der Datei `CH341SER.EXE` aus dem Unterordner `Windows` des Treiber-Ordners. Zur Installation müssen Sie eine Anfrage der Windows-Benutzerkontensteuerung bestätigen.
- 4 Klicken Sie im Installationsdialog auf **Install** und warten Sie, bis eine Bestätigung erscheint, dass der Treiber installiert wurde.



Installation des Gerätetreibers für das Nano-Board.



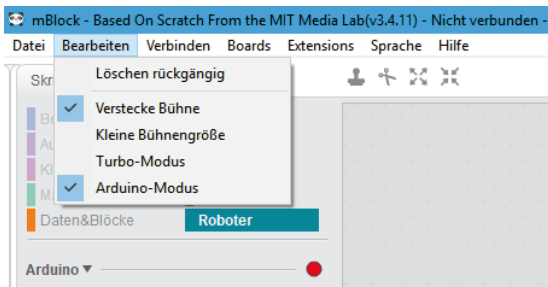
Firewall-Zugriff zulassen.

mBlock 3

Für die Projekte im Adventskalender verwenden wir die einfach zu erlernende Programmiersprache **mBlock 3**. Laden Sie sich diese bei www.mblock.cc/software/mblock/mblock3 herunter und installieren Sie sie.

Die Verbindung mit dem Arduino wird standardmäßig von der Windows Defender Firewall blockiert. mBlock benötigt eine Berechtigung, die Sie mit einem Klick auf **Zugriff zulassen** in dem Dialogfeld, das automatisch erscheint, gewähren müssen.

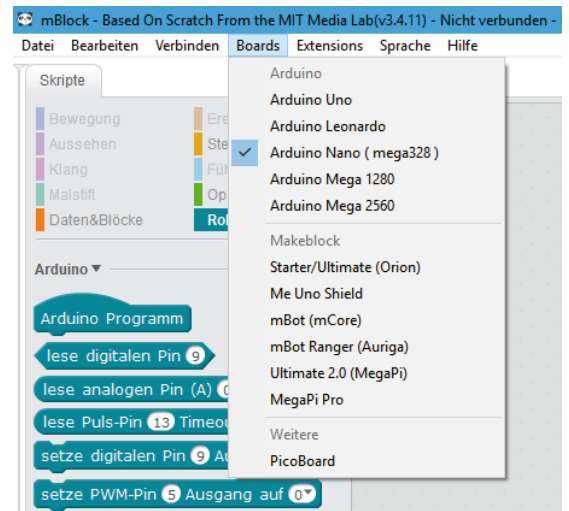
Sollte mBlock 3 nicht automatisch auf Deutsch starten, wählen Sie im Menü **Language** von mBlock die Sprache **Deutsch**. Wählen Sie anschließend im Menü **Bearbeiten / Arduino-Modus**.



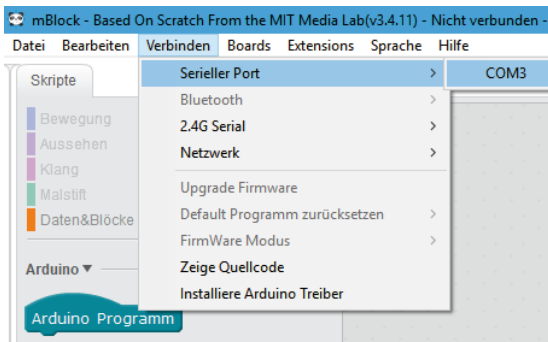
Arduino-Modus einschalten.

Im Menü **Boards** werden alle unterstützten Platinen aufgelistet. Wählen Sie hier **Arduino Nano**.

Um ein Programm auf das Nano-Board übertragen zu können, muss eine Verbindung zwischen PC und Nano hergestellt werden. Wählen Sie dazu im Menü **Verbinden / Serieller Port** den passenden Port. Solange hier nur eine COM-Schnittstelle angezeigt wird, wählen Sie diese aus. Werden zwei Schnittstellen angezeigt, ist in den meisten Fällen die mit der höheren Nummer die richtige.



Nano-Board auswählen.



Schnittstelle für die Verbindung auswählen.

Die notwendige Firmware wird bei der Übertragung von Programmen automatisch auf dem Nano-Board installiert.

LED blinkt

In mBlock 3 braucht man beim Programmieren keinen Programmcode zu tippen. Die Blöcke werden einfach nur per Drag and Drop aneinandergeschoben. Die Blockpalette im linken Teil des Fensters enthält nach Themen geordnet die verfügbaren Blöcke.

Die Programme zum Adventskalender

Die Programme zum Adventskalender finden Sie im Download, nach Tagen nummeriert. Sie können sie aber auch einfach jeden Tag anhand der Abbildungen selbst zusammenbauen. Wählen Sie nach dem Entpacken des Zip-Archivs im Menü von mBlock3 **Datei / Projekt laden**, um ein Programm zu öffnen.

Das erste Programm `01mBlock` verwendet die wichtigsten Blöcke:

Der Block **Arduino Programm** von der Palette **Roboter** bildet den Start für Programme, die nicht auf dem PC, sondern auf einer Arduino-kompatiblen Platine laufen.

Wiederhole 10-mal von der Palette **Steuerung** ist eine Schleife, die 10-mal wiederholt wird. im Zahlenfeld können Sie, wie in einigen anderen Blöcken auch, einen beliebigen (sinnvollen) Wert eintragen.

Setze digitalen Pin... auf... von der Palette **Roboter** setzt einen der digitalen Pins des Nano-Boards auf einen Logikwert **HIGH** oder **LOW**. Die Werte können mit einem Klick auf den kleinen Pfeil rechts im Feld ausgewählt werden.



Das Programm `01mBlock` lässt die LED auf dem Nano 10-mal kurz blinken.

Die LED am Pin 13

Für Statusanzeigen ohne Zusatzhardware hat der Nano eine eigene eingebaute LED, die über den Pin 13 steuerbar ist.

warte... Sek von der Palette **Steuerung** lässt das Programm eine bestimmte Zeit lang bis zum nächsten Schritt warten.

Den Nano ausschalten

Der Nano hat keinen Ausschalter, Sie brauchen einfach nur die Stromversorgung zu trennen, und er schaltet sich ab. Beim nächsten Einschalten startet automatisch das zuletzt gespeicherte Programm. Das Gleiche passiert, wenn man den Reset-Taster drückt.

So funktioniert das Programm

Die Schleife sorgt dafür, dass die darin enthaltenen Blöcke 10-mal hintereinander ausgeführt werden.

Nachdem die LED an Pin 13 eingeschaltet ist, wird 0,1 Sekunden gewartet, so lange leuchtet die LED. Danach wird die LED an Pin 13 wieder ausgeschaltet. Jetzt wartet das Programm wieder 0,1 Sekunden. Danach wiederholt sich der Zyklus von vorne.

Dezimalpunkt statt Komma

mBlock 3 verwendet, wie viele chinesische oder amerikanische Programme, den Punkt als Dezimaltrennzeichen, nicht das in Deutschland übliche Komma.

```

1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7
8 void setup() {
9   pinMode(13,OUTPUT);
10  for(int __i__=0; __i__<10; ++ __i__ )
11  {
12    digitalWrite(13,1);
13    _delay(0.1);
14    digitalWrite(13,0);
15    _delay(0.1);
16  }
17 }
18
19 void loop() {
20   _loop();
21 }
22
23 void _delay(float seconds){
24   long endTime = millis() + seconds * 1000;
25   while(millis() < endTime) _loop();
26 }
27
28 void _loop() {
29 }

```

avrdude: input file C:\Users\chris\AppData\Local\Temp\build1243307170804926706.tmp\project_0lnano4...
avrdude: reading on-chip flash data:
Reading | ##### | 100% 0.71s
avrdude: verifying ...
avrdude: 3116 bytes of flash verified
avrdude done. Thank you.

send encode mode
 Binär-Modus Zeichen-Modus

recv encode mode
 Binär-Modus Zeichen-Modus

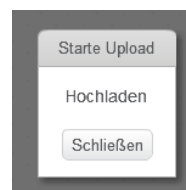
Senden

Programm auf das Nano-Board übertragen.

Programm auf den Nano übertragen

Der rechte Teil des mBlock-Bildschirms zeigt den automatisch generierten Arduino-Programmcode. Klicken Sie hier oben links auf **Upload zum Arduino**, um das Programm automatisch zu kompilieren und auf das Nano-Board zu übertragen.

Klicken Sie in der Meldung **Starte Upload** auf **Schließen**, um die Übertragung mitzufolgen zu können.



Meldung beim Start der Übertragung.

Im unteren Teil des Programmfensters sehen Sie den Fortschritt der Übertragung. Wenn der Text **Thank you** erscheint, ist die Übertragung abgeschlossen.

Das Programm startet nach der Übertragung auf den Nano automatisch. Außerdem können Sie es durch einen Druck auf die Taste auf dem Nano jederzeit wieder starten.

2. Tag

Heute im Adventskalender

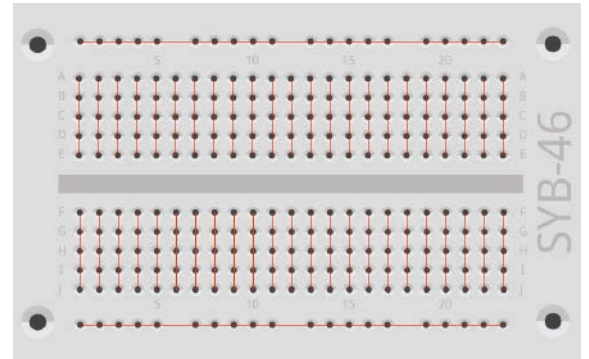
- 1 x Steckbrett (SYB 46)
- 1 x LED orange mit Vorwiderstand

2. Tag

Steckbrett

Für den schnellen Aufbau elektronischer Schaltungen, ohne löten zu müssen, ist im Adventskalender ein Steckbrett. Hier können elektronische Bauteile direkt in ein Lochraster gesteckt werden.

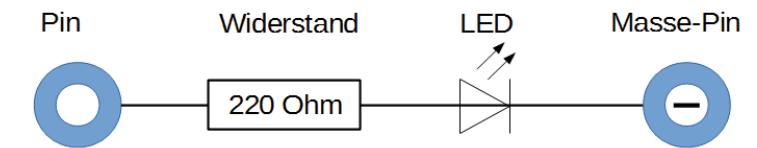
Bei diesem Steckbrett sind die äußeren Längsreihen mit Kontakten (X und Y) alle miteinander verbunden. Diese Kontaktreihen werden oft als Plus- und Minuspol zur Stromversorgung der Schaltungen genutzt. In den anderen Kontaktreihen sind jeweils fünf Kontakte (A bis E und F bis J) quer miteinander verbunden, wobei in der Mitte der Platine eine Lücke ist. So können in der Mitte größere Bauelemente eingesteckt und nach außen hin verdrahtet werden.



Die Verbindungen auf dem Steckbrett.

LEDs

LEDs (deutsch: Leuchtdioden) leuchten, wenn Strom in Durchflussrichtung durch sie fließt. LEDs werden in Schaltungen mit einem pfeilförmigen Dreieckssymbol dargestellt, das die Flussrichtung vom Pluspol zum Minuspol oder zur Masseleitung angibt. Eine LED lässt in der Durchflussrichtung nahezu beliebig viel Strom durch, sie hat nur einen sehr geringen Widerstand. Um den Durchflussstrom zu begrenzen und damit ein Durchbrennen der LED zu verhindern, muss üblicherweise zwischen dem verwendeten Anschlusspin und der Anode der LED oder zwischen Kathode und Masse-Pin ein 220-Ohm-Vorwiderstand eingebaut werden. Dieser Vorwiderstand schützt auch den Ausgang des Nano vor zu hohen Stromstärken. Die LEDs im Adventskalender haben den Vorwiderstand bereits eingebaut und können daher direkt an die Pins angeschlossen werden.



Schaltplan einer LED mit Vorwiderstand.

LED in welcher Richtung anschließen?

Die beiden Anschlussdrähte einer LED sind unterschiedlich lang. Der längere ist der Pluspol, die Anode, der kürzere der Minuspol, die Kathode. Einfach zu merken: Das Pluszeichen hat einen Strich mehr als das Minuszeichen und macht damit den Draht optisch etwas länger. Außerdem sind die meisten LEDs auf der Minuseite abgeflacht, vergleichbar mit einem Minuszeichen. Leicht zu merken: Kathode = kurz = Kante.

Wechselblinklicht

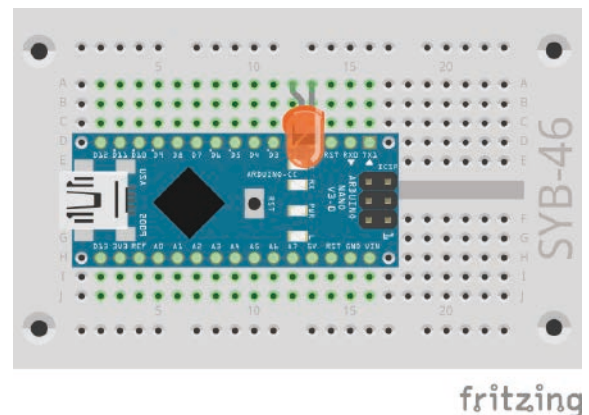
Ein einfaches Programm `02mb1ock` lässt zwei LEDs abwechselnd blinken, die eingebaute auf dem Nano-Board sowie die externe auf dem Steckbrett. Auf einem neuen Steckbrett lässt sich das Nano-Board oft nur schwer einstecken. Wenden Sie etwas Druck an, aber nutzen Sie KEIN Werkzeug, wie etwa einen Hammer oder Ähnliches.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 1 x LED orange mit Vorwiderstand

Achten Sie beim Aufbau der Schaltung darauf, dass die Kathode (kurzer Draht) der LED mit dem GND-Pin verbunden ist, die Anode (langer Draht) mit dem D2-Pin.

Die Pins auf dem Nano

Alle Pins mit **D...** sind digitale Ein- oder Ausgänge, die die Werte **Wahr** oder **Falsch** (Ein oder Aus) annehmen können. Die Pins mit **A...** sind analoge Eingänge. **GND**-Pins sind Masseleitungen. Arduino-kompatible Platinen arbeiten mit unterschiedlichen Spannungen und haben dazu standardmäßig zwei verschiedene Plus-Pins. Am Pin **3.3** liegen +3,3 V Spannung an. Am Pin **5V** liegen +5 V Spannung an. Der Nano im Adventskalender benötigt für ein logisches **Wahr**-Signal +5 V, manche anderen Platinen nur +3,3 V.



LED-Wechselblinklicht am Nano.

Das Programm

Das Programm `02mb1ock` lässt die eingebaute LED auf dem Nano und die extern angeschlossene LED abwechselnd blinken.



Das Programm `02mb1ock` lässt zwei LEDs abwechselnd blinken.

So funktioniert das Programm

Eine **wiederhole fortlaufend**-Schleife sorgt dafür, dass die beiden LEDs abwechselnd endlos blinken.

Nachdem die eingebaute LED an Pin 13 eingeschaltet ist, wird die LED an Pin 2 ausgeschaltet. Jetzt wartet das Programm 0,1 Sekunden.

Danach wird auf die gleiche Weise die LED an Pin 2 ein- und die an Pin 13 ausgeschaltet. Nach weiteren 0,1 Sekunden wiederholt sich der Zyklus.

3. Tag

Heute im Adventskalender

- Schaltdraht (isoliert)

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, die Drähte leicht schräg abzuschneiden, sodass eine Art Keil entsteht. Entfernen Sie an beiden Enden auf einer Länge von etwa einem halben Zentimeter die Isolierung.

LEDs blinken mit einstellbarer Geschwindigkeit

Das Experiment des 3. Tages lässt wieder zwei LEDs abwechselnd blinken. Allerdings können Sie die Geschwindigkeit mithilfe einer Variablen im Programm einstellen.

Bauteile 1 x Nano-Board, 1 x Steckbrett, 1 x LED orange mit Vorwiderstand, 1 x Drahtbrücke

Die Schaltung von heute zeigt den typischen Schaltungsaufbau auf dem Steckbrett. Eine der horizontalen Kontaktleisten wird als Masseleitung verwendet, die mit dem GND-Pin auf dem Nano über eine Drahtbrücke 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 mit dem Pin 10 verbunden.

Das Programm

Das Programm `03mb1ock` funktioniert ähnlich wie das von gestern und lässt wieder die eingebaute LED auf dem Nano und die diesmal am Pin 10 angeschlossene externe LED abwechselnd blinken. Die Blinkfrequenz lässt sich über eine Variable festlegen.

So funktioniert das Programm

Die **wiederhole fortlaufend**-Schleife lässt die beiden LEDs wieder abwechselnd endlos blinken. Anstatt eine vom Programm fest vorgegebene Zeit zwischen dem Umschalten wird eine Variable verwendet.

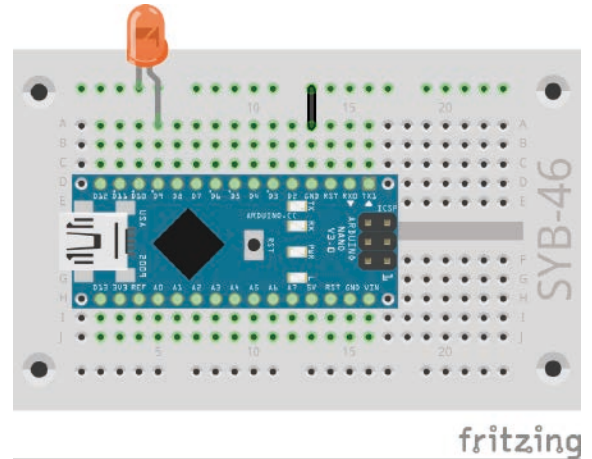
Variablen in mBlock

Variablen sind kleine Speicherplätze, in denen man sich während eines Programms eine Zahl oder irgendetwas anderes merken kann. Wenn das Programm beendet wird, werden diese Variablenspeicher automatisch wieder geleert. Variablen müssen in mBlock erst einmal auf der Blockpalette **Daten & Blöcke** mit dem Button **Neue Variable** angelegt werden, bevor man sie benutzen kann. Anschließend können Sie das Symbol der neu angelegten Variablen aus der Blockpalette in ein dafür vorgesehenes Feld eines Blocks im Programm ziehen. Auf der Blockpalette stehen zusätzlich verschiedene Blöcke zum Auslesen und Verändern der Variable zur Verfügung.

Nach dem Anlegen erscheint eine Variable als orangefarbenes Symbol auf der Blockpalette.

Im Programm wird der eingestellte Wert der Variable **zeit** für beide Wartezeiten verwendet. Geben Sie im Block **setze zeit auf 0.2** am Anfang des Programms einen anderen Wert ein, blinken die LEDs entsprechend schneller oder langsamer.

3. Tag



LED-Wechselblinklicht am Nano.

Das Programm `03mb1ock` lässt zwei LEDs abwechselnd mit einstellbarer Geschwindigkeit blinken.

Blockpalette Daten & Blöcke mit einer Variable.

4. Tag

4. Tag

Heute im Adventskalender

• 1 x LED gelb mit Vorwiderstand

LEDs blinken immer schneller

Das Experiment des 4. Tages lässt zwei LEDs auf dem Steckbrett abwechselnd blinken. Dabei nimmt die Blinkgeschwindigkeit im Laufe der Zeit immer weiter zu.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 1 x LED orange mit Vorwiderstand, 1 x LED gelb mit Vorwiderstand, 1 x Drahtbrücke

Das Programm

Das Programm `04mblock` funktioniert ähnlich wie das von gestern. Wieder werden in einer Endlosschleife abwechselnd zwei verschiedene digitale Pins ein- und ausgeschaltet. Die Variable **zeit**, die die Blinkdauer festlegt, wird in jedem Schleifendurchlauf auf 80 % des vorherigen Wertes reduziert.

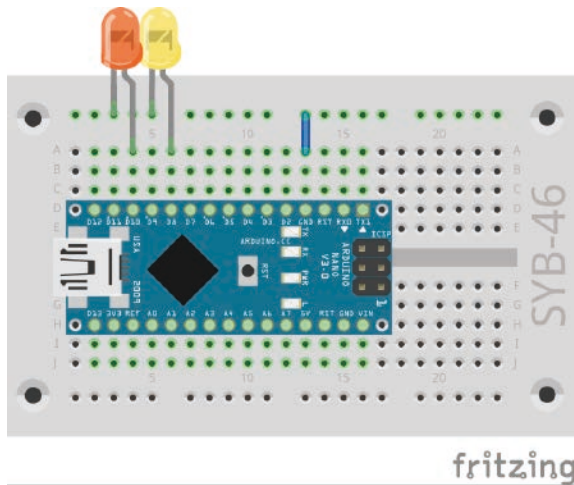


Das Programm `04mblock` lässt zwei LEDs blinken.

So funktioniert das Programm

Am Anfang des Programms wird die Variable **zeit** auf 2 Sekunden gesetzt. In jedem der 20 Schleifendurchläufe wird als Erstes die Variable **zeit** mit 0.8 multipliziert und wieder als neuer Wert für **zeit** gespeichert. Auf der Blockpalette **Operatoren** finden sich Blöcke für die vier Grundrechenarten und einige andere mathematische Operatoren.

Die weiteren Programmblöcke entsprechen der vorherigen Programmversion. Sie lassen die beiden LEDs abwechselnd blinken, wobei bei jedem Durchlauf der aktuelle Wert der Variablen **zeit** als Leuchtdauer verwendet wird.



Zwei LEDs auf dem Steckbrett blinken.

5. Tag

Heute im Adventskalender

• 1 x Taster

• 1 x Widerstand 10 kOhm (braun - schwarz - orange)

5. Tag

Widerstände und ihre Farbcodes

Widerstände werden zur Strombegrenzung an empfindlichen elektronischen Bauteilen sowie als Vorwiderstände für LEDs verwendet. Die Maßeinheit für Widerstände ist Ohm. 1.000 Ohm entsprechen einem Kiloohm, abgekürzt kOhm. 1.000 kOhm entsprechen einem Megaohm, abgekürzt MOhm. Oft wird für die Einheit Ohm auch das Omega-Zeichen Ω verwendet.

Die farbigen Ringe auf den Widerständen geben den Widerstandswert an. Mit etwas Übung sind sie deutlich leichter zu erkennen als winzig kleine Zahlen, die man nur noch auf ganz alten Widerständen findet.

Die meisten Widerstände haben vier solcher Farbringe. Die ersten beiden Farbringe stehen für die Ziffern, der dritte bezeichnet einen Multiplikator und der vierte die Toleranz. Dieser Toleranzring ist meistens gold- oder silberfarben - Farben, die auf den ersten Ringen nicht vorkommen. Dadurch ist die Leserichtung immer eindeutig. Der Toleranzwert selbst spielt in der Digitalelektronik kaum eine Rolle. Die Tabelle zeigt die Bedeutung der farbigen Ringe auf Widerständen.

In welcher Richtung ein Widerstand eingebaut wird, ist egal. Bei LEDs dagegen spielt die Einbaurichtung eine wichtige Rolle.

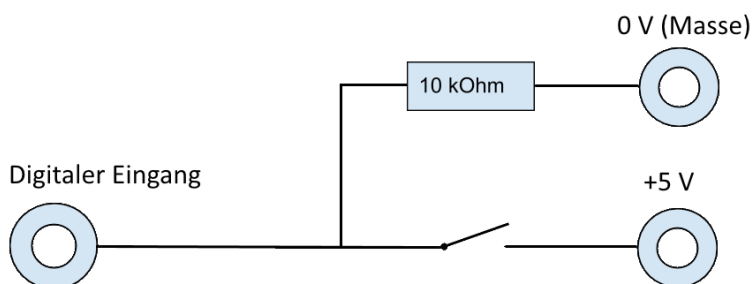
Farbe	Widerstandswert in Ohm			
	1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multiplikator)	4. Ring (Toleranz)
Silber			$10^{-2} = 0,01$	$\pm 10\%$
Gold			$10^{-1} = 0,1$	$\pm 5\%$
Schwarz		0	$10^0 = 1$	
Braun	1	1	$10^1 = 10$	$\pm 1\%$
Rot	2	2	$10^2 = 100$	$\pm 2\%$
Orange	3	3	$10^3 = 1.000$	
Gelb	4	4	$10^4 = 10.000$	
Grün	5	5	$10^5 = 100.000$	$\pm 0,5\%$
Blau	6	6	$10^6 = 1.000.000$	$\pm 0,25\%$
Violett	7	7	$10^7 = 10.000.000$	$\pm 0,1\%$
Grau	8	8	$10^8 = 100.000.000$	$\pm 0,05\%$
Weiß	9	9	$10^9 = 1.000.000.000$	

Taster

Digitale Pins können nicht nur Daten ausgeben, zum Beispiel über LEDs, sondern auch zur Dateneingabe verwendet werden. Zur Eingabe verwenden wir im heutigen Projekt einen Taster, der direkt auf die Steckplatine gesteckt wird. Der Taster hat vier Anschlusspins, wobei je zwei gegenüberliegende (großer Abstand) miteinander verbunden sind. Solange die Taste gedrückt ist, sind alle vier Anschlüsse miteinander verbunden. Im Gegensatz zu einem Schalter rastet ein Taster nicht ein. Die Verbindung wird beim Loslassen sofort wieder getrennt.

Liegt auf einem digitalen Eingang ein +5-V-Signal an, wird es als logisch **wahr** ausgewertet.

Bei offenem Taster hätte der Eingang keinen eindeutig definierten Zustand. Wenn ein Programm diesen Pin abfragt, kann es zu zufälligen Ergebnissen kommen. Um das zu verhindern, schließt man einen vergleichsweise sehr hohen Widerstand - üblicherweise 10 kOhm - gegen Masse. Dieser sogenannte Pull-down-Widerstand zieht den Status des Eingangspins bei geöffnetem Taster wieder nach unten auf 0 V. Da der Widerstand sehr hoch ist, besteht, solange der Taster gedrückt ist, auch keine Kurzschlussgefahr. Im gedrückten Zustand des Tasters sind +5 V und die Masseleitung direkt über diesen Widerstand verbunden.

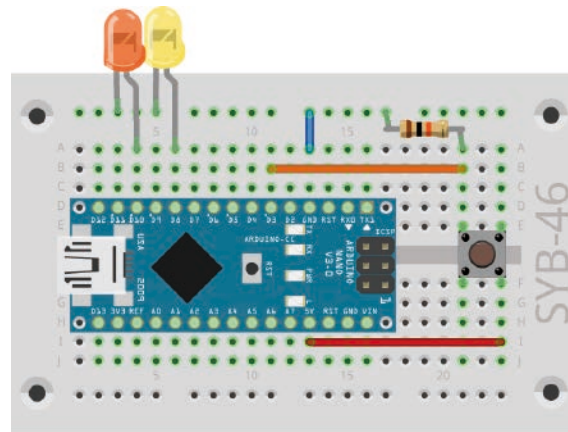


Schaltschema eines Tasters mit Pull-down-Widerstand.

LEDs mit Taster schalten

Das Experiment des 5. Tages schaltet LEDs beim Drücken eines Tasters um.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 1 x LED orange mit Vorwiderstand, 1 x LED gelb mit Vorwiderstand, 1 x Taster, 1 x 10 kOhm Widerstand (braun - schwarz - orange), 3 x Drahtbrücke (unterschiedliche Längen)



fritzing

Zwei LEDs mit einem Taster umschalten.

Das Programm

Das Programm `05mb1ock` prüft regelmäßig, ob der Taster gedrückt ist. Je nachdem wird eine LED ein- und die andere ausgeschaltet.



Das Programm `05mb1ock` schaltet die LEDs um, wenn der Taster gedrückt ist.

So funktioniert das Programm

In jedem Durchlauf der Endlosschleife wird in einem **falls ... dann ... sonst**-Block der digitale Pin 3 abgefragt, an dem der Taster angeschlossen ist. Ist der Taster gedrückt, ergibt die Abfrage den Logikwert **wahr**. In diesem Fall wird die LED am Pin 8 ein- und die LED am Pin 10 ausgeschaltet. Ist der Taster nicht gedrückt, wird umgekehrt die LED am Pin 8 aus- und die LED am Pin 10 eingeschaltet.

6. Tag

Heute im Adventskalender

• 4 x Anschlusskabel

Heute sind Anschlusskabel im Adventskalender, mit denen die LEDs auf dem Engelsflügel mit dem Steckbrett verbunden werden. Schneiden Sie den Engelsflügel von der Rückseite des Adventskalenders aus und stecken Sie zwei LEDs in die dafür vorgesehenen Löcher. Die Positionen mit zwei Löchern sind für einfarbige LEDs, die Positionen mit vier Löchern werden später erst benötigt.

Stecken Sie die Buchsen der Anschlusskabel auf die Drähte der LEDs. Stecken Sie dann die Steckerseiten der Anschlusskabel gemäß der Abbildung auf das Steckbrett.

LEDs auf dem Engelsflügel umschalten

Das Experiment des 6. Tages schaltet über einen Taster zwei LEDs um. Im Gegensatz zum vorherigen Programm behalten die LEDs ihren Status, nachdem der Taster losgelassen wurde. Erst der nächste Tastendruck schaltet die LEDs wieder um.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 1 x LED orange mit Vorwiderstand, 1 x LED gelb mit Vorwiderstand, 1 x Taster, 1 x 10-kOhm-Widerstand (braun - schwarz - orange), 3 x Drahtbrücke (unterschiedliche Längen), 4 x Anschlusskabel

Das Programm

Das Programm `06mblock` wartet, bis der Taster gedrückt wird, und schaltet dann die LEDs um.



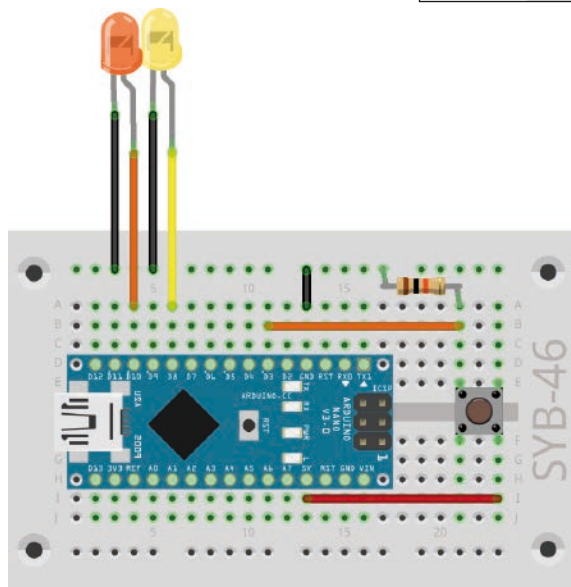
Das Programm `06mblock` schaltet zwei LEDs mit einem Taster um.

So funktioniert das Programm

Am Anfang jedes Durchlaufs der Endlosschleife wird die LED am Pin 8 ein- und die am Pin 10 ausgeschaltet. Anschließend wartet das Programm, bis der Block **lese digitalen Pin 3** den Wert **wahr** ergibt, der Taster also gedrückt wurde.

Jetzt wird die LED am Pin 8 aus- und die am Pin 10 eingeschaltet und danach wieder gewartet, bis der Taster gedrückt wird. Erst dann startet die Endlosschleife neu und schaltet die LEDs wieder auf die Ausgangsstellung.

6. Tag



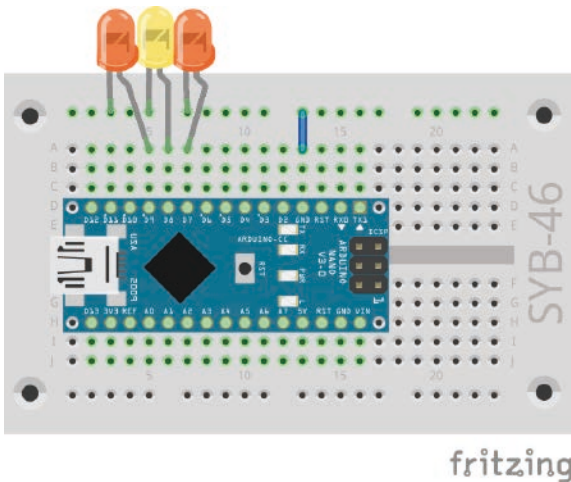
Ein Taster schaltet zwei LEDs um.

7. Tag

7. Tag

Heute im Adventskalender

- 1 x LED orange mit Vorwiderstand



Drei LEDs blinken zufällig.

LEDs blinken zufällig

Das Experiment des 7. Tages lässt drei LEDs in zufälliger Reihenfolge blinken. Die drei LEDs stecken sehr dicht nebeneinander auf dem Steckbrett, da das Programm aufeinander folgende Pinnummern benötigt.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 1 x LED gelb mit Vorwiderstand, 1 x Drahtbrücke

Das Programm

Das Programm `07mblock01` lässt drei LEDs mit aufeinander folgenden Pin-Nummern zufällig abwechselnd blinken.



Das Programm `07mblock01` lässt drei LEDs zufällig blinken.

Wie entstehen Zufallszahlen?

Gemeinhin denkt man, in einem Programm könne nichts zufällig geschehen – wie also kann ein Programm 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 zusammengesetzte 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.

So funktioniert das Programm

Im ersten Schritt jedes Durchlaufs der Endlosschleife wird die Variable `n` auf eine Zufallszahl zwischen 7 und 9 gesetzt. Diese gibt die Pinnummer der einzuschaltenden LED an. Deshalb benötigt die Schaltung drei aufeinander folgende Pinnummern.

Die Geschwindigkeit des Farbwechsels wird über die Variable `zeit` gesteuert, die am Anfang des Programms auf 0.2 Sekunden eingestellt wird und dann für jeden Schaltvorgang gilt.

Die zufällig gewählte LED wird für die eingestellte Zeit eingeschaltet und danach genauso lange ausgeschaltet. Im nächsten Schleifendurchlauf wird wieder eine neue LED zufällig gewählt. Dabei kann es durchaus passieren, dass mehrmals hintereinander die gleiche LED aufleuchtet.

Verbesserte Version des Programms

Bei diesem Programm fällt auf, dass die LEDs deutlich schwächer leuchten als bei den bisherigen Programmen.

Ein Blick in den Arduino-Programmcode im rechten Teilfenster von mBlock zeigt die Ursache dieses Problems.

Jeder Pin, der als digitaler Ausgang genutzt werden soll, muss vor seiner ersten Verwendung initialisiert werden. mBlock legt dazu normalerweise im Bereich `void setup() {...}` automatisch die notwendigen Zeilen an.

Da die verwendeten Pin-Nummern nicht einzeln genannt werden, kann mBlock in diesem Fall die Initialisierung nicht vornehmen.

Fügen Sie am Anfang des Programms drei Blöcke **setze digitalen Pin ... Ausgang auf LOW** ein. Dann wird jeder der drei verwendeten Pins einzeln ausgeschaltet und damit auch initialisiert.



Das Programm 07mb1ock02 lässt drei LEDs zufällig, aber heller blinken.

Diese drei Blöcke sorgen dafür, dass die notwendigen Zeilen zur Initialisierung der Ausgänge in den Arduino-Programmcode geschrieben werden.

```

1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7 double zeit;
8 double n;
9
10 void setup() {
11     pinMode(7,OUTPUT);
12     pinMode(8,OUTPUT);
13     pinMode(9,OUTPUT);
14     pinMode(n,OUTPUT);
15     digitalWrite(7,0);
16     digitalWrite(8,0);
17     digitalWrite(9,0);
18     zeit = 0.2;
19 }

```

Im Programm 07mb1ock02 werden die Ausgänge für die LEDs initialisiert.

In dieser Programmversion leuchten die LEDs deutlich heller.

Zurück
Upload zum Arduino

```

1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7 double zeit;
8 double n;
9
10 void setup() {
11     zeit = 0.2;
12     pinMode(n,OUTPUT);
13 }

```

Die Ausgänge für die LEDs werden nicht initialisiert.

8. Tag

8. Tag

Heute im Adventskalender

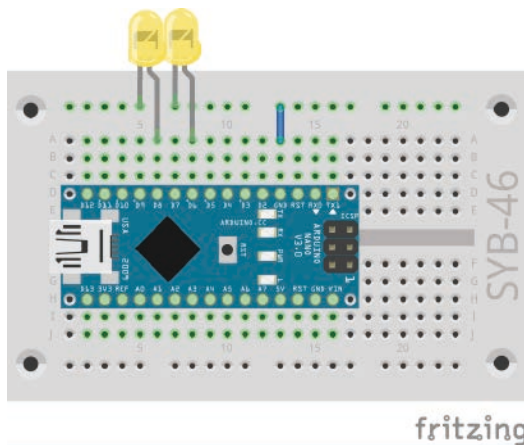
- 1 x LED gelb mit Vorwiderstand

LEDs dimmen

LEDs sind typische Bauteile zur Ausgabe von Signalen in der Digitalelektronik. Sie können zwei verschiedene Zustände annehmen, ein und aus, 0 und 1 oder **LOW** und **HIGH**. Das Gleiche gilt für die als Ausgänge definierten digitalen Pins. Demnach wäre es theoretisch nicht möglich, eine LED zu dimmen.



Links: Tastverhältnis 50 % - rechts: Tastverhältnis 20 %.



Eine LED wird gedimmt, die zweite leuchtet zum Vergleich mit voller Helligkeit.



Das Programm 08mb1ock dimmt eine LED am PWM-Ausgang.

Mit einem Trick ist es dennoch möglich, die Helligkeit einer LED an einem digitalen Pin zu regeln. Lässt man eine LED schnell genug blinken, nimmt das menschliche Auge das nicht mehr als Blinken wahr. Die als Pulsweitenmodulation (PWM) bezeichnete Technik erzeugt ein pulsierendes Signal, das sich in sehr kurzen Abständen ein- und ausschaltet. Die Spannung des Signals bleibt immer gleich, nur das Verhältnis zwischen Level **LOW** (0 V) und Level **HIGH** (+3,3 V) wird verändert. Das

Tastverhältnis gibt das Verhältnis der Länge des eingeschalteten Zustands zur Gesamtdauer eines Schaltzyklus an.

Je kleiner das Tastverhältnis, desto kürzer ist die Leuchtzeit der LED innerhalb eines Schaltzyklus. Dadurch wirkt die LED dunkler als eine permanent eingeschaltete LED.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED gelb mit Vorwiderstand, 1 x Drahtbrücke

Pins für PWM-Signale

Die Pins 3, 5, 6, 9, 10, 11 sind auf den Schaltbildern mit einem '~-Symbol gekennzeichnet. Diese Pins können für PWM verwendet werden.

Das Programm

Das Programm 08mb1ock dimmt die LED am Pin 6 zyklisch heller und dunkler. Die gleichfarbige LED am Pin 8 leuchtet zum Vergleich in voller Helligkeit.

So funktioniert das Programm

Am Anfang wird der Pin 8 als digitaler Pin auf **HIGH** gesetzt und damit eingeschaltet. Danach werden drei Variablen definiert: **zeit** legt die Geschwindigkeit beim Dimmen fest, **hell** bezeichnet den PWM-Wert für die Helligkeit der LED und **schritt** gibt die Schrittweite beim Dimmen an.

Jetzt beginnt eine Endlosschleife. Als Erstes wird bei jedem Schleifendurchlauf der aktuelle Wert der Variable **hell** als PWM-Wert auf den Pin 6 ausgegeben. Anschließend wird der Wert der Variable **hell** um den Wert **schritt** erhöht.

Im nächsten Schritt wird überprüft, ob der Wert von **hell** die Grenzen 0 oder 250 erreicht hat. In diesem Fall wird ein **oder**-Block eingesetzt, der wiederum Platz für zwei weitere Abfragen enthält. Ist von diesen beiden mindestens eine wahr, gibt der **oder**-Block den Wert **wahr** zurück und der Inhalt des **falls**-Blocks wird ausgeführt.

Zwei Gleichheitsabfragen prüfen, ob der Wert der Variablen **hell** den Wert 0 oder 250 erreicht hat. Trifft dies zu, wird die Variable **schritt** auf einen neuen Wert gesetzt. Da mBlock keine Möglichkeit bietet, das Vorzeichen einer Variable umzukehren, verwenden wir den Operator '-' und subtrahieren den Wert der Variable von 0, was das gleiche Ergebnis liefert.

Zum Schluss wartet das Programm in jedem Schleifendurchlauf die in der Variable **zeit** gespeicherten 0.2 Sekunden. Anschließend startet die Endlosschleife neu und liefert der LED einen neuen PWM-Wert.

9. Tag

Heute im Adventskalender

• 4 x Anschlusskabel

Lauflicht auf dem Engelsflügel

Das Experiment des 9. Tages lässt vier LEDs zyklisch als Lauflicht aufleuchten. Stecken Sie die LEDs in die dafür vorgesehenen Löcher auf dem Engelsflügel und verbinden Sie sie mit den Anschlusskabeln gemäß der Abbildung mit dem Steckbrett.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 1 x Drahtbrücke, 8 x Anschlusskabel

Das Programm

Das Programm `09mb1ock` lässt die vier LEDs zyklisch als Lauflicht aufleuchten. Die Pin-Nummern haben jeweils einen Abstand von 2 voneinander. So können sie einfach in einer Schleife hochgezählt werden.



Das Programm `09mb1ock` steuert ein Lauflicht aus vier LEDs auf dem Engelsflügel.

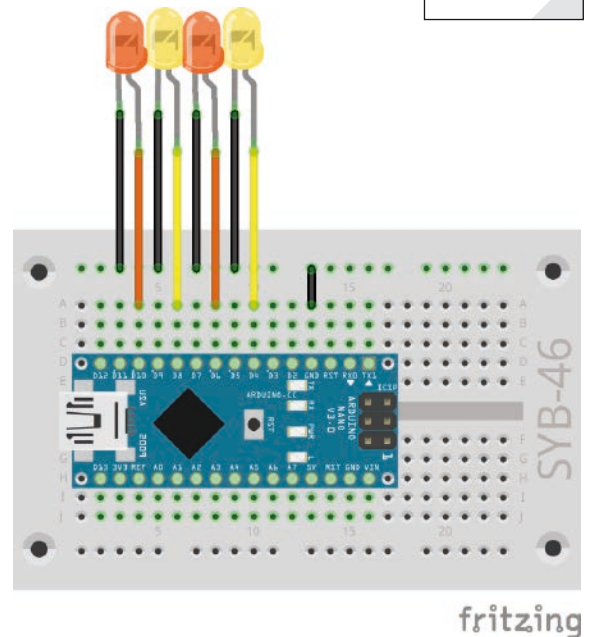
So funktioniert das Programm

Das Programm legt am Anfang in der Variablen `zeit` die Blinkgeschwindigkeit fest. Hier können Sie natürlich auch einen anderen Wert eintragen. Danach werden die vier für die LEDs verwendeten Pins einmal ausgeschaltet, um sie zu initialisieren, da sie später nur noch über eine Variable angesprochen werden.

Danach startet die Endlosschleife, die in jedem Durchlauf die Variable `n` auf 4 setzt, die Nummer der ersten LED. Jeder Durchlauf der Endlosschleife bewirkt einen Lauflichtzyklus aller vier LEDs.

Eine innere Schleife lässt nacheinander die vier LEDs kurz aufleuchten. Nach jeder LED wird die Variable `n` um 2 erhöht und damit die nächste LED angesteuert. Nachdem alle LEDs einmal geleuchtet haben, startet der nächste Durchlauf der Endlosschleife mit der LED am Pin 4.

9. Tag



Vier LEDs leuchten als Lauflicht.

10. Tag

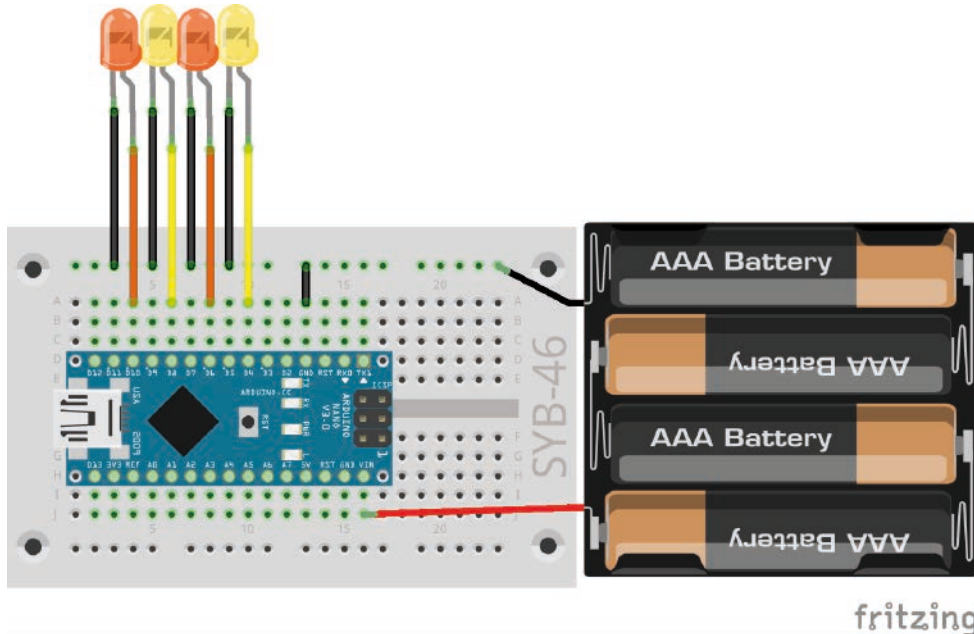
10. Tag

Heute im Adventskalender

• 1 x Batteriekasten

Batteriekasten

Das Nano-Board kann auch ohne PC genutzt werden und ein gespeichertes Programm abarbeiten. Dazu benötigt es eine externe Stromversorgung. Das kann ein USB-Handyladegerät, eine Powerbank oder auch eine Batterie sein. Heute ist im Adventskalender ein Batteriekasten enthalten, der mit vier AAA-Batterien eine Spannung von 6 V liefert, mit Akkus 4,8 V, was zur Stromversorgung des Nano-Boards auch ausreicht. Die Batterien sind nicht enthalten.



Vier LEDs leuchten als Lauflicht.

Das Steckbrett können Sie mit der selbstklebenden Rückseite auf die Rückseite des Engelsflügels kleben und mit dem Batteriekasten die ganze Schaltung unabhängig vom PC herumtragen.

Schließen Sie den Batteriekasten jetzt noch nicht an, da Sie den Nano noch über den PC mit Strom versorgen, bis das neue Programm übertragen ist. Der Batteriekasten wird nach dem Trennen des USB-Kabels mit den Pins **VIN** und **GND** des Nano-Boards verbunden.

Lauflicht mit wechselnder Geschwindigkeit

Das Experiment des 10. Tages lässt die vier LEDs wieder als Lauflicht leuchten, dabei verändert sich die Geschwindigkeit zyklisch. Nach der Übertragung des Programms trennen Sie das USB-Kabel vom PC und stecken den Batteriekasten wie in der Abbildung an. Das Programm läuft dann eigenständig auf dem Nano-Board.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 1 x Drahtbrücke, 8 x Anschlusskabel, 1 x Batteriekasten

Das Programm

Das Programm `10mb1ock` lässt die vier LEDs wieder zyklisch als Lauflicht aufleuchten. Nach jedem Zyklus wird die Geschwindigkeit verändert. Die Leuchtdauer jeder einzelnen LED schwankt langsam zwischen 0.5 Sekunden und 0.05 Sekunden.

So funktioniert das Programm

Das Programm `10mb1ock` funktioniert ähnlich wie das Programm des 9. Tages. Nach jedem Durchlauf des Lauflichts wird die Variable **zeit** mit dem Faktor **f** multipliziert. Dieser Faktor hat am Anfang den Wert 0.8. Jeder Durchlauf dauert demnach nur noch 80 % des vorhergehenden Durchlaufs. Das Lauflicht wird immer schneller.

Ist nach mehreren Wiederholungen die Leuchtzeit der einzelnen LEDs kürzer als 0.05 Sekunden geworden, wird der Faktor **f** auf 1.2 gesetzt. Dadurch dauert jeder zukünftige Durchlauf 20 % länger als der vorhergehende. So wird das Lauflicht langsam abgebremst, bis die Leuchtzeit der einzelnen LEDs wieder länger als 0.5 Sekunden geworden ist. Dann wird **f** wieder auf 0.8 gesetzt.

```

Arduino Programm
setze zeit auf 0.5
setze f auf 0.8
setze digitalen Pin 4 Ausgang auf LOW
setze digitalen Pin 6 Ausgang auf LOW
setze digitalen Pin 8 Ausgang auf LOW
setze digitalen Pin 10 Ausgang auf LOW
wiederhole fortlaufend
  setze n auf 4
  wiederhole 4 mal
    setze digitalen Pin n Ausgang auf HIGH
    warte zeit Sek.
    setze digitalen Pin n Ausgang auf LOW
    ändere n um 2
  setze zeit auf zeit * f
  falls zeit < 0.05 dann
    setze f auf 1.2
  falls zeit > 0.5 dann
    setze f auf 0.8
  
```

Das Programm `10mb1ock` lässt die LEDs als Lauflicht mit veränderlicher Geschwindigkeit aufleuchten.

11. Tag

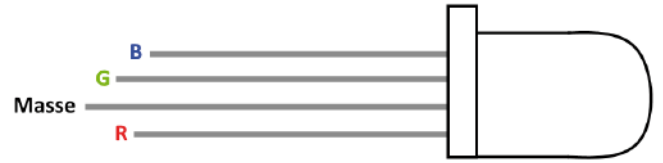


Heute im Adventskalender

• 1 x RGB-LED mit Vorwiderständen

RGB-LEDs

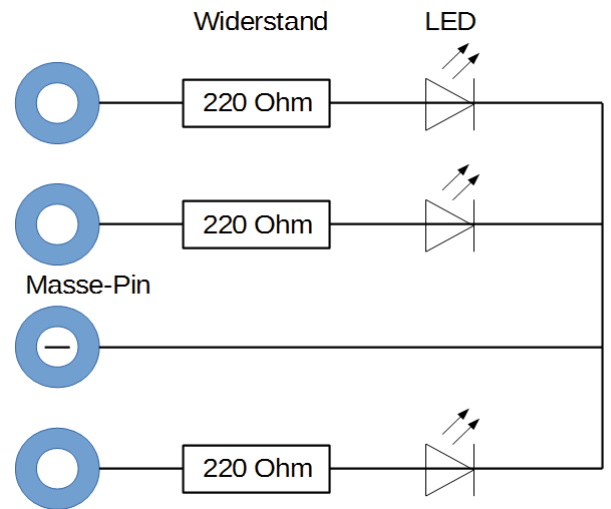
Eine normale LED leuchtet immer nur in einer Farbe. Die im Adventskalender verwendeten RGB-LEDs können wahlweise in mehreren Farben leuchten. Hier sind im Prinzip drei LEDs in verschiedenen Farben in einem transparenten Gehäuse eingebaut. Jede dieser drei LEDs hat eine eigene Anode, über die sie mit einem GPIO-Pin verbunden wird. Die Kathode, die mit der Masseleitung verbunden wird, ist nur einmal vorhanden. Deshalb hat eine RGB-LED vier Anschlussdrähte. Auf dem Engelsflügel sind Positionen mit vier Löchern für RGB-LEDs vorgesehen.



Anschlusspins einer RGB-LED.

Die Anschlussdrähte der RGB-LEDs sind unterschiedlich lang, um sie eindeutig kenntlich zu machen. Anders als bei normalen LEDs ist die Kathode bei ihnen der längste Draht.

RGB-LEDs funktionieren wie drei einzelne LEDs und brauchen deshalb auch drei 220-Ohm-Vorwiderstände (rot - rot - braun). In den RGB-LEDs in diesem Adventskalender sind sie ebenfalls bereits eingebaut.

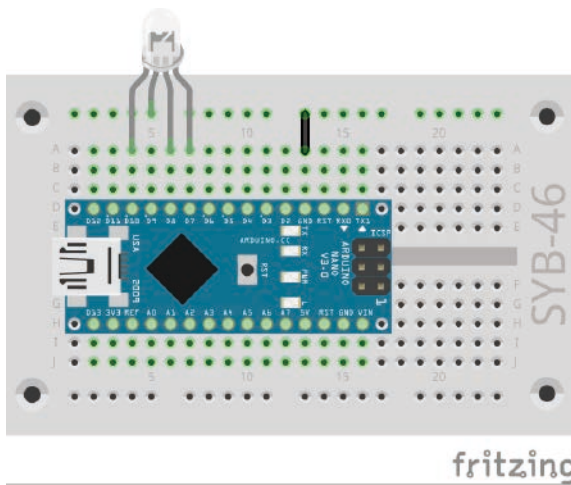


Schaltplan für eine RGB-LED mit drei Vorwiderständen.

Verschiedene Farben auf einer RGB-LED

Das Experiment des 11. Tages zeigt unterschiedliche Grund- und Mischfarben auf einer RGB-LED.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 1 x RGB-LED mit Vorwiderständen, 1 x Drahtbrücke



Eine RGB-LED zeigt wechselnde Farben.

Das Programm

Das Programm 11mblock funktioniert ähnlich wie die Programme der letzten Tage. Auch hier werden in einer Endlosschleife nacheinander verschiedene digitale Pins ein- und ausgeschaltet. In diesem Fall handelt es sich um die drei Farbkomponenten der RGB-LED.

Additive Farbmischung
 RGB-LEDs verwenden die sogenannte additive Farbmischung. Dabei werden die drei Lichtfarben Rot, Grün und Blau addiert und ergeben am Ende reines Weiß. Im Gegensatz dazu verwendet ein Farbdruker die subtraktive Farbmischung. Jede Farbe wirkt auf einem weißen Blatt wie ein Filter, der einen Teil des weiß reflektierten Lichtes wegnimmt (= subtrahiert). Drückt man alle drei Druckerfarben übereinander, ergibt es Schwarz, das gar kein Licht mehr reflektiert.



Verschiedene Farben auf einer RGB-LED.



Additive Farbmischung.

Der Download zu diesem Adventskalender enthält eine farbige Version des Handbuchs als PDF-Datei. Dort ist der Effekt der additiven Farbmischung besser zu erkennen.

So funktioniert das Programm

Im Programm leuchten durch abwechselndes Ein- und Ausschalten immer mal eine, mal zwei Farbkomponenten. Dadurch wechselt die RGB-LED zwischen sechs verschiedenen Farben hin und her. Die Variablen **r**, **g**, **b** bezeichnen die drei Anschlusspins der RGB-LED. So ist im Programm leicht zu erkennen, welche Farbkomponenten ein- oder ausgeschaltet werden.

Die Geschwindigkeit des Farbwechsels wird auch hier über die Variable **zeit** gesteuert, die am Anfang des Programms auf 0.2 Sekunden gesetzt wird und dann für jeden Farbwechsel gilt.

12. Tag

Heute im Adventskalender

• 4 x Anschlusskabel

RGB-Farbmischung mit PWM

Das Experiment des 12. Tages zeigt zusätzlich zu den Farben auf der RGB-LED auch die einzelnen Farbkomponenten mit drei LEDs an. Die Geschwindigkeit des Farbwechsels kann in zwei Stufen gewählt werden.

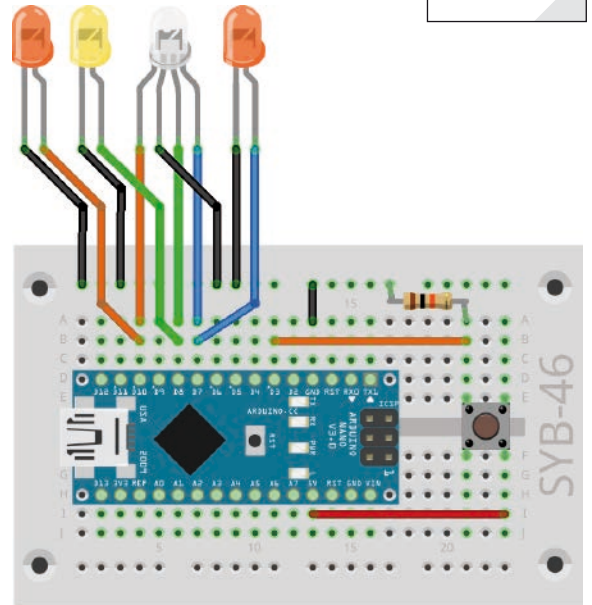
Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 1 x LED gelb mit Vorwiderstand, 1 x RGB-LED mit Vorwiderstand, 1 x Taster, 1 x 10 kOhm Widerstand (braun - schwarz - orange), 3 x Drahtbrücke (unterschiedliche Längen), 10 x Anschlusskabel

LEDs anschließen

An jedem der drei verwendeten Ausgangspins sind je eine Farbe der RGB-LED und eine einzelne LED angeschlossen, die dann immer gleichzeitig leuchten. Stecken Sie die LEDs und die RGB-LED in die dafür vorgesehenen Löcher auf dem Engelsflügel und verbinden Sie die Anschlussdrähte über die Kabel mit dem Steckbrett, wie abgebildet.

So funktioniert das Programm

Das Programm `12mblock` funktioniert mit einem Unterschied wie das Programm des 11. Tages.



fritzing

RGB-LED und drei einzelne LEDs mit einem Taster steuern.



Das Programm `12mblock` schaltet eine RGB-LED und einzelne LEDs.

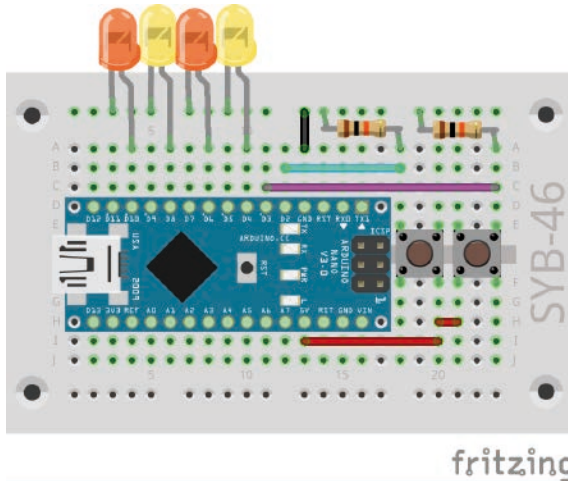
Am Anfang jedes Schleifendurchlaufs wird der Taster am Pin 3 abgefragt. Solange er nicht gedrückt ist, erhält die Variable **zeit** den Wert 0.1. Der Farbwechsel läuft schnell. Durch einen Druck auf den Taster lässt er sich bremsen. Solange der Taster gedrückt ist, erhält die Variable **zeit** den Wert 0.5. Die einzelnen Farben leuchten deutlich länger.

13. Tag

13. Tag

Heute im Adventskalender

- 1 x Taster
- 1 x Widerstand 10 kOhm (braun - schwarz - orange)



Lauflicht mit vier LEDs und zwei Tastern mit Pull-down-Widerständen.

Lauflicht in zwei Richtungen

Lauflichter sind immer wieder beliebte Effekte, nicht nur für Werbung und Party-räume. Das Experiment des 13. Tages lässt vier LEDs beim Druck auf einen Taster als Lauflicht leuchten. Der andere Taster lässt das Lauflicht in umgekehrter Richtung laufen.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 2 x Taster, 2 x 10-kOhm-Widerstand (braun - schwarz - orange), 5 x Drahtbrücke (unterschiedliche Längen)

Das Programm

Das Programm `13mblock` funktioniert ähnlich wie das Programm des 9. Tages, fragt nur zusätzlich die beiden Taster ab, um die Richtung des Lauflichts mithilfe einer zusätzlichen Variable auszuwählen.



Lauflichtrichtung mit zwei Tastern umschalten.

So funktioniert das Programm

Die Variable `i` gibt den Wert an, um den die Nummer des Pins für die LED, die leuchten soll, in jedem Schleifendurchlauf verändert wird. Der Wert 2 lässt das Lauflicht von rechts nach links laufen: 4, 6, 8, 10, der Wert -2 lässt es von links nach rechts laufen: 10, 8, 6, 4.

Die Variable `n` enthält die Nummer des Pins, der gerade leuchtet oder als Nächstes leuchten soll.

In jedem Durchlauf der Schleife leuchtet die LED am Pin `n` für die in der Variable `zeit` festgelegte Zeit. Danach werden die beiden Taster abgefragt. Ist der linke Taster am Pin 2 gedrückt, wird die Variable `i` auf 2 gesetzt, damit das Lauflicht nach links läuft. Ist der rechte Taster am Pin 3 gedrückt, wird die Variable `i` auf -2 gesetzt, damit das Lauflicht nach rechts läuft.

Die Pin-Nummer `n` wird auf den neuen Wert gesetzt. Danach wird noch geprüft, ob `n` den Bereich der möglichen Werte für die LED-Pins überschritten hat. Hat `n` durch die Veränderung einen Wert größer als 10, beginnt das Lauflicht rechts wieder bei Pin 4. Ist der neue Wert von `n` dagegen kleiner als 4, beginnt das Lauflicht wieder links bei Pin 10.

14. Tag

Heute im Adventskalender

• 15-kOhm-Potentiometer

Potentiometer

Das Potentiometer aus dem Adventskalender von heute ist ein einstellbarer Widerstand, der Werte zwischen 0 Ohm und 15 kOhm annehmen kann, indem man den Knopf dreht. Mit dem Potentiometer lässt sich ein Spannungsteiler bauen, der eine beliebige Spannung zwischen 0 V und +5 V liefern kann. Diese analoge Spannung muss in einen digitalen Wert umgerechnet werden, der dann vom Programm weiterverarbeitet wird.

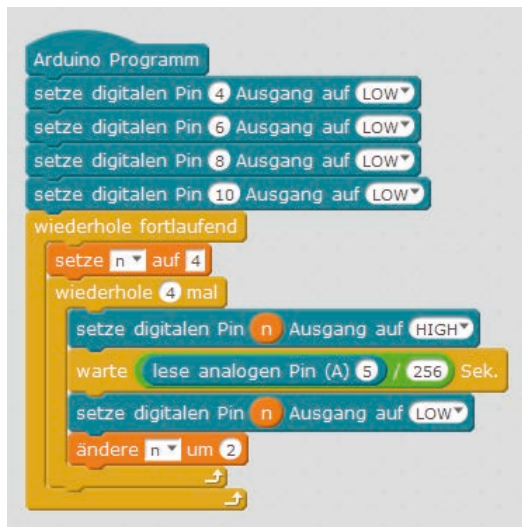
Laufflicht mit Potentiometer steuern

Das Nano-Board verfügt über acht analoge Eingangspins **A0...A8**, die eine anliegende Spannung in einen Zahlenwert umrechnen. Ein Potentiometer an einem analogen Eingangspins des Nano-Boards regelt die Geschwindigkeit des Laufflichts.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 1 x Potentiometer, 4 x Drahtbrücke (unterschiedliche Längen)

Das Programm

Das Programm `14mblock` steuert eine LED-Leiste anhand der Einstellung des Potentiometers.



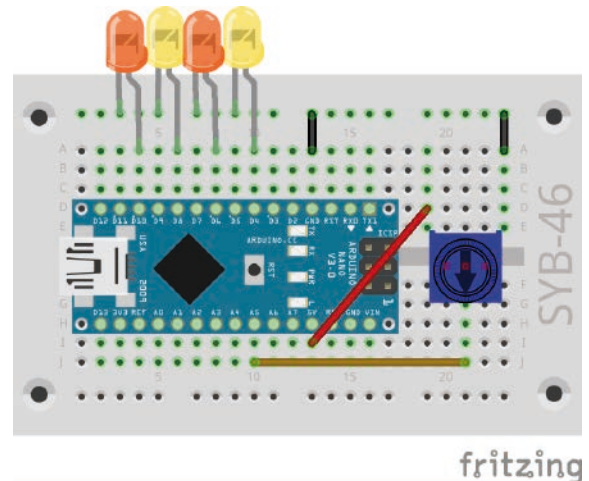
Das Programm `14mblock`.

So funktioniert das Programm

Das Laufflicht funktioniert wie die Laufflicht-Programme der letzten Tage, nur wird die Leuchtzeit der einzelnen LEDs nicht in einer Variable **zeit** festgelegt, sondern über das Potentiometer am analogen Eingang **A5** interaktiv eingestellt.

Der analoge Pin **A5** wird in jedem Schleifendurchlauf abgefragt und sein Wert durch 256 geteilt, um brauchbare Werte für das Laufflicht zu erhalten. Der so errechnete Wert legt die Zeit fest, wie lange jede einzelne LED leuchtet.

14. Tag



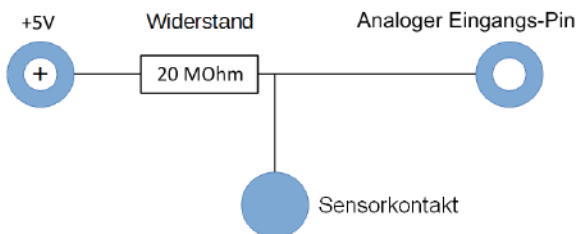
Laufflicht mit vier LEDs mit Potentiometer steuern.

15. Tag

15. Tag

Heute im Adventskalender

- 1 x Knete
- 1 x 20-MOhm-Widerstand (rot - schwarz - blau)



Sensorkontakt aus Knete

Ampeln, Türöffner, Lichtschalter und Automaten werden heute oft mit Sensorkontakten gesteuert, die man nur zu berühren braucht. Taster, die wirklich gedrückt werden müssen, werden immer seltener. Das Experiment des 15. Tages schaltet zwei LEDs über einen einfachen Sensorkontakt um.

Der als Eingang geschaltete Pin ist über einen extrem hochohmigen Widerstand (20 MOhm) mit +5 V verbunden, sodass ein schwaches, aber eindeutig als **HIGH** definiertes Signal anliegt. Ein Mensch, der nicht gerade frei in der Luft schwebt, ist immer geerdet und liefert über die elektrisch leitfähige Haut einen Low-Pegel. Berührt der Mensch einen Sensorkontakt, wird das schwache **HIGH**-Signal von dem deutlich stärkeren **LOW**-Pegel der Fingerkuppe überlagert und zieht den Pin auf **LOW**-Pegel.

Wie hoch allerdings der Widerstand zwischen Hand und Masse wirklich ist, hängt von vielen Dingen ab, unter anderem von Schuhen und Fußboden. Barfuss im nassen Gras ist die Verbindung zur Masse der Erde am besten, aber auch auf Steinfußboden funktioniert es meistens gut. Holzfußböden isolieren stärker, Kunststoffbodenbeläge sind oft sogar positiv aufgeladen. Damit die Schaltung immer funktioniert, ist ähnlich wie bei Sensortasten an Aufzügen und Türen bei jeder Schaltung zusätzlich ein Massekontakt vorgesehen. Berührt man diesen und den eigentlichen Sensor gleichzeitig, ist die Masseverbindung auf jeden Fall hergestellt.

Knete leitet den Strom etwa so gut wie menschliche Haut. Sie lässt sich leicht in jede beliebige Form bringen, und ein Knetekontakt fasst sich viel besser an als ein einfaches Stück Draht. Die Fläche, mit der die Hand den Kontakt berührt, ist deutlich größer. So kommt es nicht so leicht zu einem „Wackelkontakt“. Schneiden Sie ein etwa 10 cm langes Stück des Schaltdrahts ab, entfernen Sie an beiden Enden auf etwa 1 cm Länge die Isolierung und stecken Sie das eine Ende in ein Stück Knete. Stecken Sie das andere Ende wie auf der Abbildung in das Steckbrett.

Da mBlock die in den meisten Arduino-kompatiblen Platinen eingebauten Pull-down-Widerstände immer einschaltet, werden digitale Eingänge automatisch auf 0 V gezogen und haben auch ohne Berührung einen Low-Pegel. Arduino-kompatible Platinen verfügen aber zusätzlich über analoge Eingänge, die sich sehr gut für Sensorkontakte eignen. Analoge Eingänge liefern Werte zwischen 0 (**LOW**-Pegel) und 1023 (**HIGH**-Pegel). Je nach Platinentyp sind Werte um 200 gute Grenzwerte, um zwischen berührtem und nicht berührtem Sensorkontakt zu unterscheiden.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 1 x LED orange mit Vorwiderstand, 1 x LED gelb mit Vorwiderstand, 1 x 20-MOhm-Widerstand (rot - schwarz - blau), 2 x Knetekontakt, 1 x Drahtbrücke

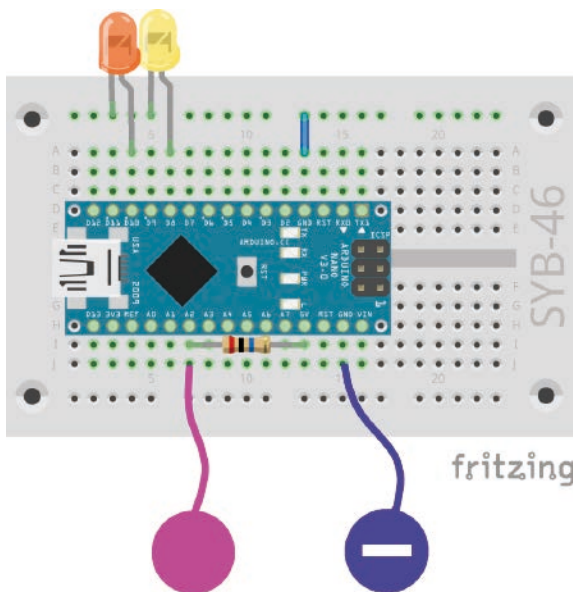
Das Programm

Das Programm `15mblock` schaltet zwei LEDs über einen Sensorkontakt aus Knete um.

So funktioniert das Programm

In jedem Durchlauf der Endlosschleife wird der analoge Pin A2 ausgelesen. Er hat, solange der Sensorkontakt nicht berührt wird, einen sehr hohen Wert, da er über den 20-MOhm-Widerstand mit +5 V verbunden ist.

Berührt man den Sensorkontakt und am besten den Massekontakt gleichzeitig, fällt der Wert des analogen Eingangs-Pins deutlich, bis fast auf 0. Das Programm prüft, ob der Grenzwert von 200 unterschritten wird. In diesem Fall wird die LED am Pin 10 ein- und die LED am Pin 8 ausgeschaltet. Solange der Eingangswert über 200 liegt, leuchtet umgekehrt die LED am Pin 8 und die am Pin 10 ist ausgeschaltet.



LEDs mit Sensorkontakt umschalten.



Das Programm `15mblock`.

16. Tag

Heute im Adventskalender

• 1 x RGB-LED mit Vorwiderständen

RGB-Farbmischung mit PWM

Über PWM-Signale lassen sich auf RGB-LEDs noch deutlich mehr Farben anzeigen. Im Experiment des 16. Tages regelt ein Potentiometer auf einer RGB-LED die rote Farbe, auf der anderen RGB-LED die grüne Farbe. Ein Knetkontakt schaltet die jeweils andere Farbe, Grün oder Rot, dazu. Für die beiden PWM-Signale werden PWM-Pins des Nano-Boards verwendet.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x RGB-LED mit Vorwiderstand, 1 x Potentiometer, 1 x 20-MOhm-Widerstand (rot - schwarz - blau), 2 x Knetkontakt, 4 x Drahtbrücke (unterschiedliche Längen)

Das Programm

Je nach Stellung des Potentiometers leuchtet die eine RGB-LED mehr oder weniger stark in Rot, die andere in Grün. Beim Berühren des Sensorkontakts werden die Farben Grün und Rot in voller Helligkeit dazugeschaltet, was bei schwach leuchtenden LEDs eine scheinbare Farbumkehr bewirkt. Bei heller Einstellung leuchten beide in einem hellen gelblichen Grün.



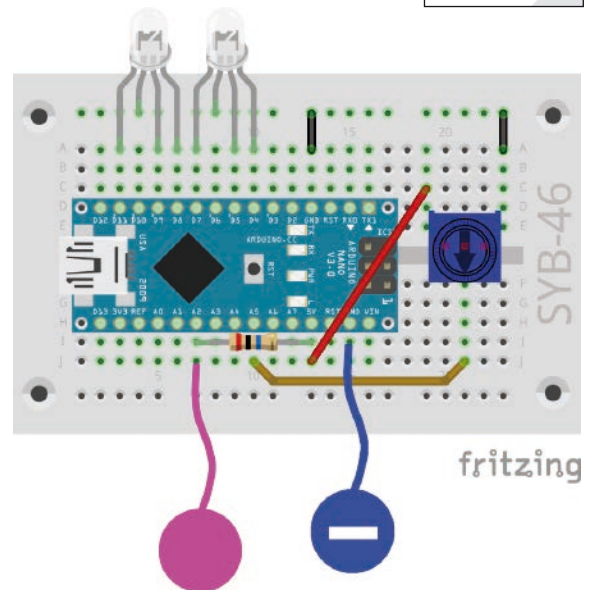
Das Programm 16mBlock steuert zwei RGB-LEDs über ein Potentiometer und einen Sensorkontakt.

So funktioniert das Programm

Die Anschlusspins der blauen Farbe werden für beide RGB-LEDs am Anfang ausgeschaltet, um keine zufälligen Farbverfälschungen zu zeigen. Alternativ können Sie diese Anschlusspins auch frei in der Luft hängen lassen oder mit Masse verbinden.

Die Endlosschleife liest den analogen Wert am Pin A5 und rechnet ihn auf die Skala möglicher PWM-Werte um. Die beiden PWM-Pins 5 und 11, an denen die grüne Farbe der rechten und die rote Farbe der linken RGB-LED angeschlossen sind, werden auf den errechneten Wert gesetzt. Danach fragt ein **falls...dann...sonst**-Block wie im Programm des 15. Tages den Sensorkontakt ab und schaltet die Pins 7 und 9, an denen die rote Farbe der rechten und die grüne Farbe der linken RGB-LED angeschlossen sind, bei Berührung ein.

16. Tag



RGB-LEDs über Potentiometer und Knetkontakt steuern.

17. Tag

17. Tag

Heute im Adventskalender

- 4 x Anschlusskabel

RGB-Farbspektrum

Das Programm des 17. Tages lässt eine RGB-LED zyklisch in allen Farben des Farbspektrums leuchten, wenn der Sensorkontakt berührt wird. Lässt man den Sensor los, wird die RGB-LED dunkel.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 1 x RGB-LED mit Vorwiderstand, 1 x 20-MOhm-Widerstand (rot - schwarz - blau), 2 x Knetkontakt, 1 x Drahtbrücke, 4 x Anschlusskabel

Die RGB-LED kann in die dafür vorgesehenen Löcher auf dem Engelsflügel gesteckt werden.

Das Programm

Die Hauptschleife des Programms `17mb1ock` errechnet nacheinander die RGB-Werte für alle Farbwerte des HSV-Farbspektrums. Nach jedem Wert wird geprüft, ob der Sensorkontakt berührt wird. Ist dies der Fall, leuchtet die RGB-LED in der entsprechenden Farbe.

HSV- und RGB-Farbsystem

Das RGB-Farbsystem, das bisher in allen Programmen verwendet wurde, beschreibt Farben als drei Komponenten, Rot, Grün und Blau, die miteinander gemischt werden. Es ist für einen Menschen relativ schwierig, sich eine Mischfarbe vorzustellen. Im Gegensatz dazu beschreibt das HSV-Farbsystem die Farben über die Werte H = Hue (Farbwert), S = Saturation (Sättigung) und V = Value (Helligkeitswert).

Durch einfache Veränderung des H-Wertes können alle Farben des Farbspektrums in voller Intensität beschrieben werden, wenn man die beiden anderen Werte auf Maximum einstellt.

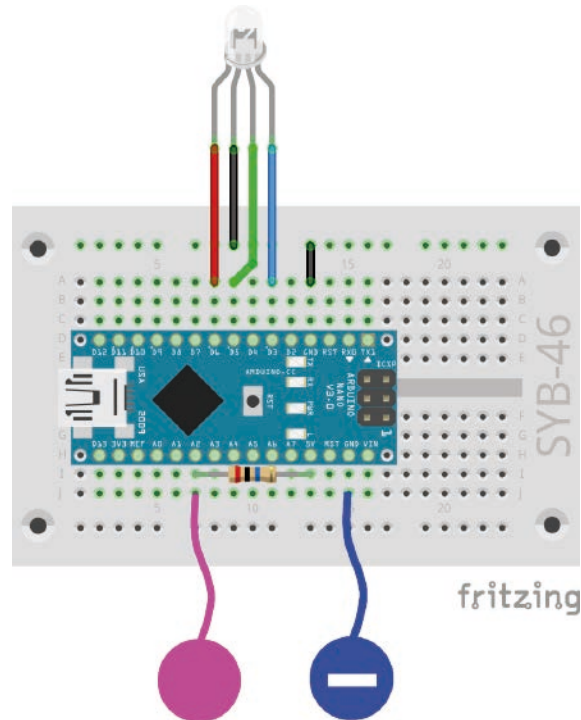
So funktioniert das Programm

Die drei Variablen `r1`, `g1`, `b1` enthalten die als PWM-Ausgänge definierten Anschlusspins für die RGB-LED. Zusätzlich werden drei Variablen `rot`, `gruen`, `blau` angelegt, in denen während der Berechnung die PWM-Werte für die drei Farbkomponenten der RGB-LED eingetragen werden.

Die Hauptschleife zählt den H-Wert entsprechend der Gradzahlen auf einem Farbkreis zwischen 0 und 360 hoch und errechnet daraus die drei Farbkomponenten R, G und B. Die Werte S = Saturation (Sättigung) und V = Value (Helligkeitswert) werden automatisch als Maximalwert gesetzt. Wie die Grafik zeigt, gelten für die sechs 60°-Bereiche jeweils eigene lineare Verlaufskurven.

Nachdem die RGB-Werte errechnet und in den Variablen `rot`, `gruen`, `blau` gespeichert sind, wird der Sensorkontakt abgefragt. Wenn er berührt wird, werden die PWM-Pins auf die jeweiligen Farbwerte gesetzt. Nach einer kurzen Wartezeit wird der H-Wert um 1 erhöht. Nach einem vollen Durchlauf über 360° beginnt der Zyklus von Neuem bei 0.

Im Download zum Adventskalender ist dieses Handbuch als farbiges PDF enthalten, in dem das Farbspektrum als solches zu erkennen ist.

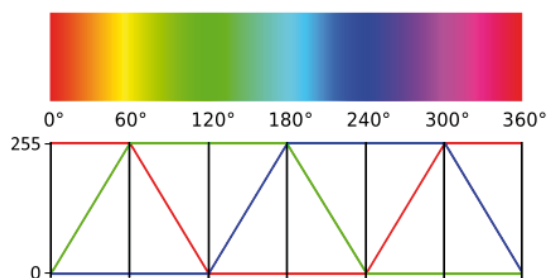


RGB-Farbverlauf über Knetkontakt steuern.

```

Arduino Programm
setze zeit auf 0.1
setze r1 auf 6
setze g1 auf 5
setze b1 auf 3
setze PWM-Pin r1 Ausgang auf 0
setze PWM-Pin g1 Ausgang auf 0
setze PWM-Pin b1 Ausgang auf 0
wiederhole fortlaufend
  setze h auf 0
  wiederhole 360 mal
    setze x auf Betrag h / 60 mod 2 - 1 * 255
    falls h < 360 dann
      setze rot auf 255
      setze gruen auf 0
      setze blau auf x
    falls h < 300 dann
      setze rot auf x
      setze gruen auf 0
      setze blau auf 255
    falls h < 240 dann
      setze rot auf 0
      setze gruen auf x
      setze blau auf 255
    falls h < 180 dann
      setze rot auf 0
      setze gruen auf 255
      setze blau auf x
    falls h < 120 dann
      setze rot auf x
      setze gruen auf 255
      setze blau auf 0
    falls h < 60 dann
      setze rot auf 255
      setze gruen auf x
      setze blau auf 0
    falls lese analogen Pin (A) 2 < 200 dann
      setze PWM-Pin r1 Ausgang auf rot
      setze PWM-Pin g1 Ausgang auf gruen
      setze PWM-Pin b1 Ausgang auf blau
    sonst
      setze PWM-Pin r1 Ausgang auf 0
      setze PWM-Pin g1 Ausgang auf 0
      setze PWM-Pin b1 Ausgang auf 0
  warte zeit Sek
  ändere h um 1
  
```

Das Programm `17mb1ock` rechnet HSV-Werte in RGB um.



Die Grafik zeigt, wie der H-Wert einer HSV-Farbe in RGB-Werte umgerechnet wird.

18. Tag

Heute im Adventskalender

• 1 x RGB-LED mit Vorwiderständen

LED-Würfel

Die typischen Spielwürfel, die ein bis sechs Augen zeigen, kennt jeder und hat jeder zu Hause. Wesentlich cooler ist ein elektronisch gesteuerter Würfel, der auf Tastendruck die Augen leuchten lässt – aber nicht einfach ein bis sechs LEDs in einer Reihe, sondern in der Anordnung eines Spielwürfels. Diese haben Augen in der typischen quadratischen Anordnung, wozu man sieben LEDs braucht. Auf dem Engelsflügel finden Sie passende Stecklöcher für vier einfarbige LEDs und drei RGB-LEDs in der Anordnung von Würfelaugen. Die drei RGB-LEDs werden in diesem Projekt nur als rote LEDs genutzt. Hier ist nur jeweils eine Farbe angeschlossen.

Für die Ansteuerung der LEDs werden nur vier statt sieben digitale Pins benötigt, da ein Würfel zur Darstellung gerader Zahlen die Augen paarweise nutzt.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 3 x RGB-LED mit Vorwiderstand, 1 x 20-MOhm-Widerstand (rot – schwarz – blau), 2 x Knetkontakt, 1 x Drahtbrücke, 14 x Anschlusskabel

Das Programm

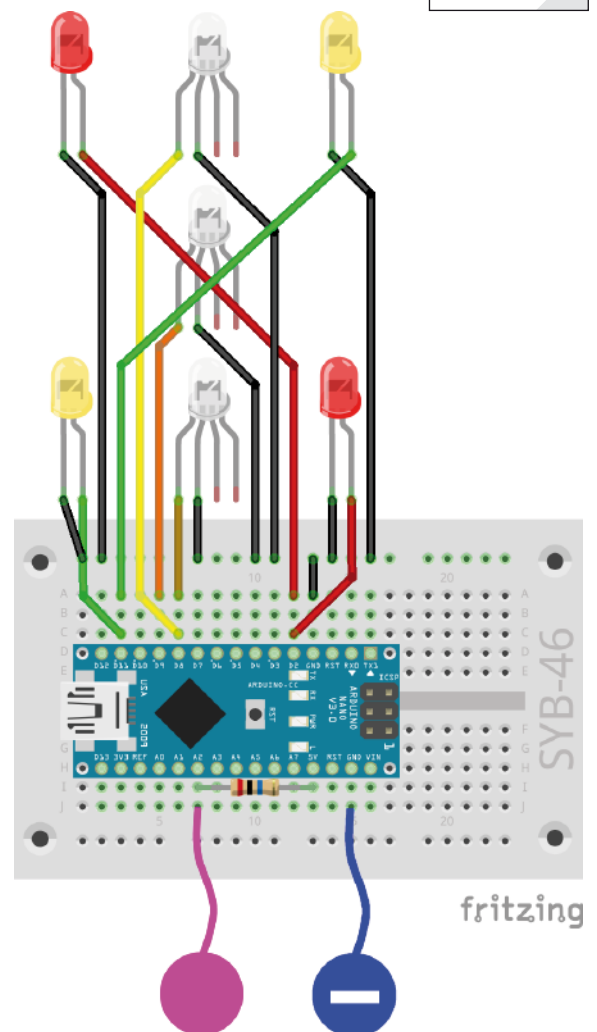
Das Programm `18mblock` würfelt beim Berühren des Sensorkontaktes eine Zahl zwischen 1 und 6 und zeigt sie als Würfelbild an.

So funktioniert das Programm

Eine Endlosschleife prüft immer wieder, ob der Sensorkontakt berührt wird. Ist dies der Fall, werden als Erstes die vier für LEDs verwendeten Pins ausgeschaltet, um das zuvor angezeigte Würfelergebnis zu löschen.

Anschließend wird in der Variablen `w` eine Zufallszahl zwischen 1 und 6 gespeichert. Für jedes mögliche Ergebnis gibt es einen eigenen `falls...`-Block, der die entsprechenden LEDs einschaltet. Sie leuchten, bis der Sensorkontakt erneut berührt wird.

18. Tag



Lauflicht mit vier LEDs, zwei RGB-LEDs und Sensorkontakten.

19. Tag

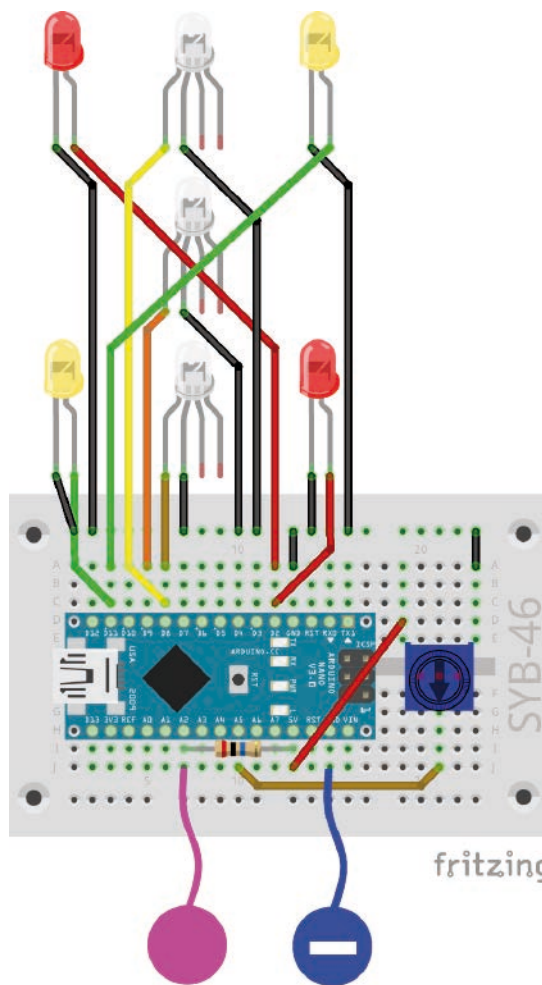
Heute im Adventskalender

• 4 x Anschlusskabel

LED-Würfel mit realistischem Würfelfeffekt

Ein wirklicher Würfel zeigt nicht sofort die endgültige Augenzahl an, sondern rollt erst noch eine kurze Zeit lang, in der man Würfelergebnisse sieht, die sich aber doch nicht bewahrheiten. Das Programm des 19. Tages simuliert das Rollen, indem der Würfel erst ein paar andere Würfelergebnisse mit minimal immer länger werdenden Pausen dazwischen anzeigt, bevor das endgültige Ergebnis erscheint.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 3 x RGB-LED mit Vorwiderstand, 1 x Potentiometer, 1 x 20-MOhm-Widerstand (rot - schwarz - blau), 2 x Knetkontakt, 4 x Drahtbrücke (unterschiedliche Längen), 14 x Anschlusskabel



Laufflicht mit vier LEDs, zwei RGB-LEDs, Potentiometer und Sensorkontakten.

Das Programm

Das Programm `19mblock` simuliert bei Berührung des Sensorkontakts einen rollenden Würfel. Über das Potentiometer lässt sich einstellen, wie viele Zwischenwerte der Würfel anzeigen soll, bis er endgültig zum Liegen kommt.

So funktioniert das Programm

Am Anfang werden, wie schon in der letzten Programmversion, die vier für die LEDs verwendeten Pins als Ausgänge definiert und ausgeschaltet. Die Hauptschleife wartet wieder, bis der Sensorkontakt berührt wird. Dann wird die Variable `zeit` auf 0.1 Sekunden gesetzt, die Wartezeit zwischen den ersten beiden Würfelanzeigen.

Anschließend wird der über das Potentiometer eingestellte Wert ausgelesen. Der analoge Eingang liefert Werte zwischen 0 und 1023. Damit der Würfel je nach Einstellung mindestens zweimal, aber nicht mehr als 12-mal rollt, wird der Eingangswert durch 100 geteilt und anschließend 2 addiert. Das Ergebnis wird als Zähler für die innere Schleife verwendet, die nacheinander die errechnete Anzahl von Würfelergebnissen anzeigt bis der Würfel zum Liegen kommt.

Am Ende jedes Durchlaufs wird die Wartezeit in der Variable `zeit` um 0.1 Sekunden verlängert, was ein langsames Ausrollen des Würfels simuliert.

Arduino Programm

```

setze led1 auf 9
setze led2 auf 2
setze led3 auf 11
setze led4 auf 8

setze digitalen Pin led1 Ausgang auf LOW
setze digitalen Pin led2 Ausgang auf LOW
setze digitalen Pin led3 Ausgang auf LOW
setze digitalen Pin led4 Ausgang auf LOW

wiederhole fortlaufend
falls lese analogen Pin (A) 2 < 200 dann
  setze zeit auf 0.1
  setze n auf lese analogen Pin (A) 5 / 100 + 2
  wiederhole n mal
    setze digitalen Pin led1 Ausgang auf LOW
    setze digitalen Pin led2 Ausgang auf LOW
    setze digitalen Pin led3 Ausgang auf LOW
    setze digitalen Pin led4 Ausgang auf LOW
    warte 0.1 Sek.
    setze w auf Zufallszahl von 1 bis 6
    falls w = 1 dann
      setze digitalen Pin led1 Ausgang auf HIGH
    falls w = 2 dann
      setze digitalen Pin led2 Ausgang auf HIGH
    falls w = 3 dann
      setze digitalen Pin led1 Ausgang auf HIGH
      setze digitalen Pin led3 Ausgang auf HIGH
    falls w = 4 dann
      setze digitalen Pin led2 Ausgang auf HIGH
      setze digitalen Pin led3 Ausgang auf HIGH
    falls w = 5 dann
      setze digitalen Pin led1 Ausgang auf HIGH
      setze digitalen Pin led2 Ausgang auf HIGH
      setze digitalen Pin led3 Ausgang auf HIGH
    falls w = 6 dann
      setze digitalen Pin led2 Ausgang auf HIGH
      setze digitalen Pin led3 Ausgang auf HIGH
      setze digitalen Pin led4 Ausgang auf HIGH
  ändere zeit um 0.1
  warte zeit Sek.

```

Würfel mit einstellbarer Verzögerung.

20. Tag

Heute im Adventskalender

• 1 x RGB-LED mit Vorwiderständen

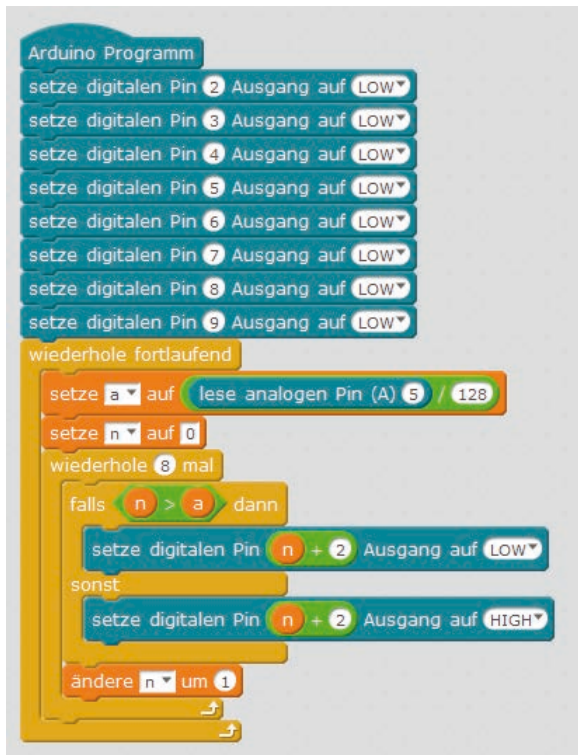
Analoge Pegelanzeige mit LEDs

Auf einer Pegelanzeige lassen sich analoge Werte auf einen Blick ablesen. Solche Anzeigen aus mehreren LEDs werden zum Beispiel bei Lautstärke- oder Temperaturreglern verwendet. Das Experiment des 20. Tages zeigt den auf dem Potentiometer eingestellten Wert über eine Reihe von LEDs an. Dabei werden die RGB-LEDs nur einfarbig verwendet. Je nach Farbe sind einer oder zwei der Anschlüsse mit dem Steckbrett verbunden.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 4 x RGB-LED mit Vorwiderstand, 1 x Potentiometer, 4 x Drahtbrücke (unterschiedliche Längen), 18 x Anschlusskabel

Das Programm

Das Programm `20mb1ock` liest den eingestellten Wert des Potentiometers am analogen Eingang A5 aus und zeigt ihn mithilfe einer Schleife auf acht LEDs an.

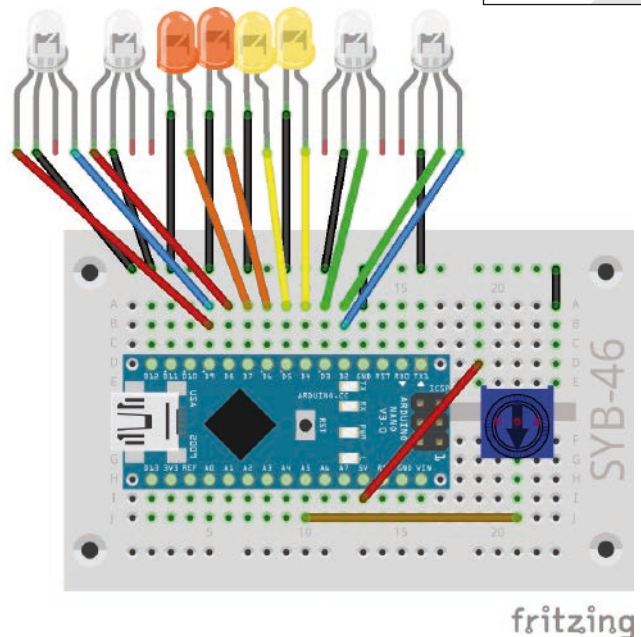


So funktioniert das Programm

In der Hauptschleife wird zuerst der analoge Wert des Potentiometers ausgelesen, durch 128 geteilt, um Werte zwischen 0 und 7 zu erhalten, und in der Variablen `a` gespeichert.

Der Zähler `n` beginnt bei 0. Dann startet wieder eine Schleife, die für die acht LEDs achtmal durchläuft. Bei jeder LED wird geprüft, ob die Nummer der LED `n` größer ist als der in `a` gespeicherte Pegelwert. Ist dies der Fall, wird die entsprechende LED ausgeschaltet. Ist die LED-Nummer nicht größer, also kleiner oder gleich dem Pegelwert, wird die LED eingeschaltet. Auf diese Weise werden immer alle LEDs geprüft, unabhängig davon, ob das Potentiometer hoch- oder heruntergedreht wird.

20. Tag



Pegelanzeige mit acht LEDs und Potentiometer.

21. Tag

21. Tag

Heute im Adventskalender

- 1 x Knete
- 1 x 20-MOhm-Widerstand (rot - schwarz - blau)

Lichteffekte mit Sensorkontakten steuern

Das Programm des 21. Tages lässt bei Berührung des einen Sensorkontakts vier LEDs zyklisch als Lauflicht aufleuchten. Bei Berührung des anderen Sensorkontakts blinken zwei RGB-LEDs in unterschiedlichen Farben.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 2 x RGB-LED mit Vorwiderstand, 2 x 20-MOhm-Widerstand (rot - schwarz - blau), 3 x Knetkontakt, 1 x Drahtbrücke, 16 x Anschlusskabel

Da die Aufbauzeichnung mit den vielen Anschlusskabeln leicht unübersichtlich wirkt, zeigen die folgenden Abbildungen den gleichen Aufbau noch einmal in zwei Teilen, nur die einzelnen LEDs und nur die RGB-LEDs.

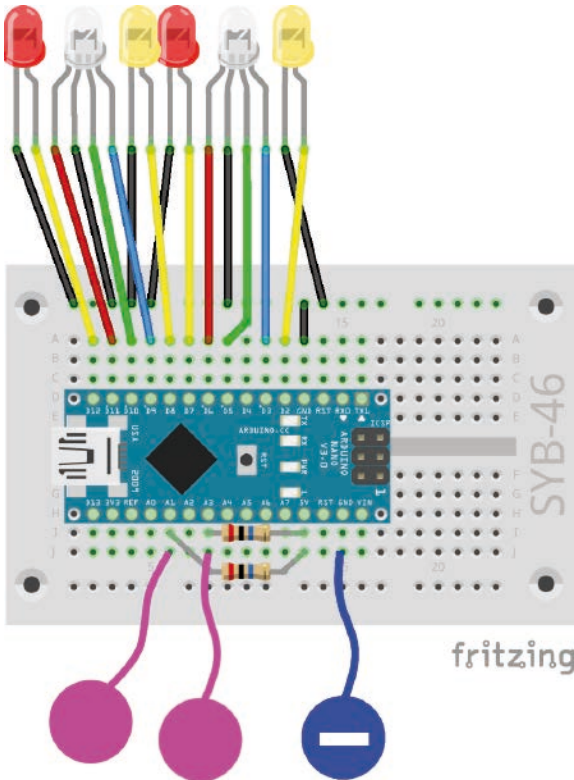
Das Programm

Das Programm `21mb10ck` nutzt die Möglichkeit, in mBlock eigene Blöcke zu generieren. In anderen Programmiersprachen werden diese als Funktionen bezeichnet.

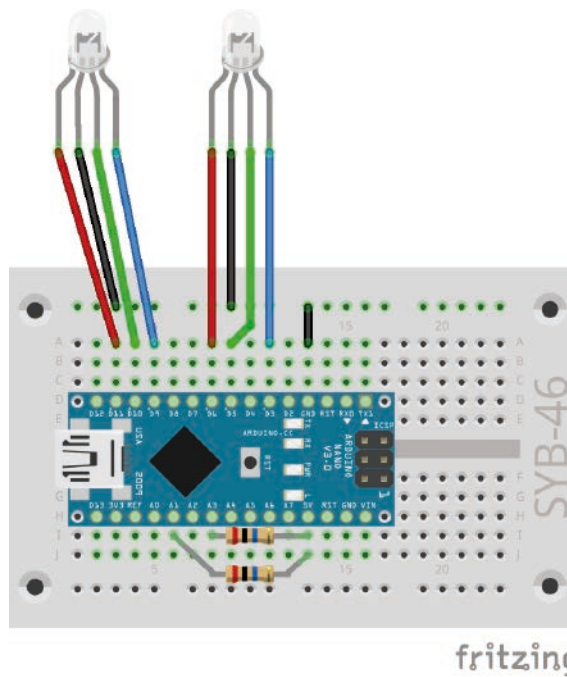
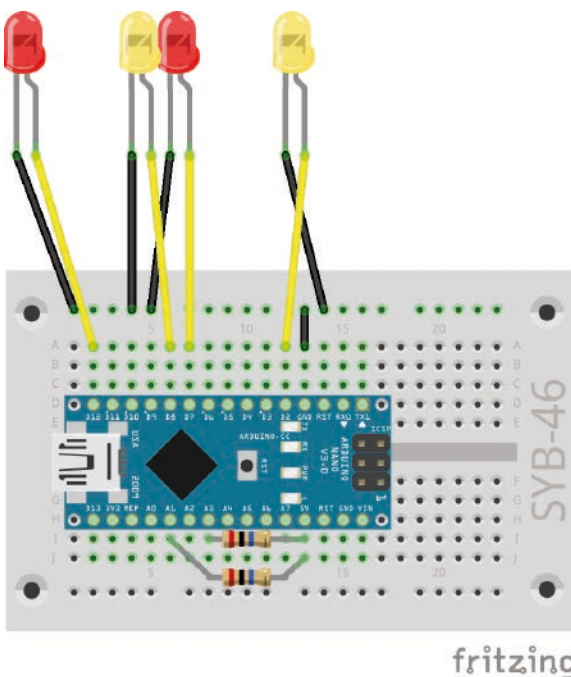
So funktioniert das Programm

Das Hauptprogramm initialisiert die Anschlusspins der LEDs und RGB-LEDs. Danach läuft eine Endlosschleife, die über selbstdefinierte Blöcke unterschiedliche Lichtmuster ausgibt.

Um einen eigenen Block zu bauen, klicken Sie auf der Blockpalette **Daten&Blöcke** auf **Neuer Block**.

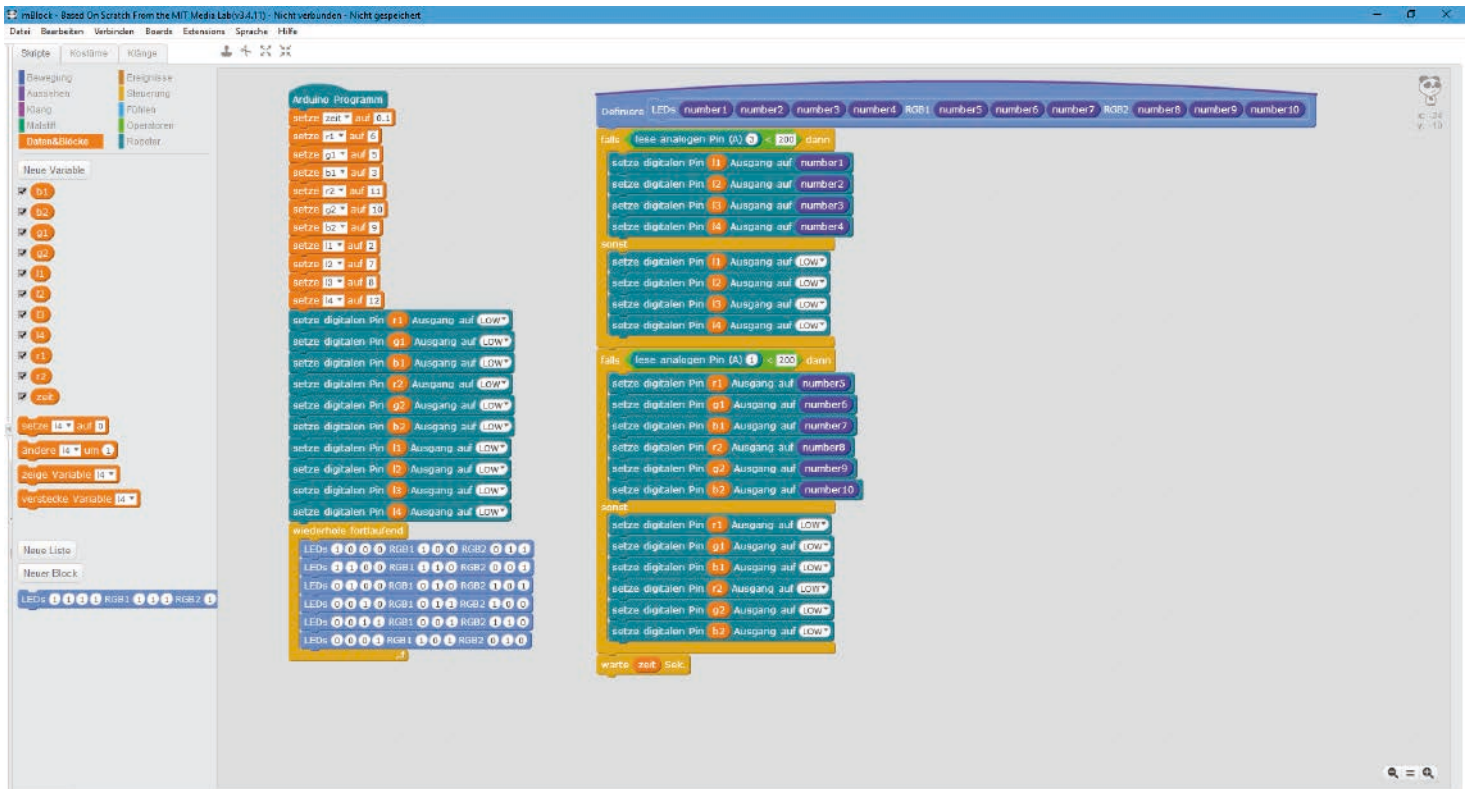


Lauflicht mit vier LEDs, zwei RGB-LEDs und Sensorkontakten.



Aufbauzeichnung in zwei Teilen.

Im Fenster **Neuer Block** legen Sie fest, was im Block später zu sehen sein soll. Dies können Beschriftungstexte, Zahlenfelder, Textfelder und Boolesche Felder sein. Anschließend bauen Sie unter die Blockdefinition die Programmblöcke, die ausgeführt werden sollen. Dabei kann der neue Block globale



Das Programm 21mBlock lässt mit einem selbstdefinierten Block LEDs blinken.

Variablen aus dem Hauptprogramm verwenden und auch eigene Parameter, die beim Aufruf des Blocks übergeben werden. Ziehen Sie diese, im Programm als **number1**, **number2**, ... bezeichnet, wie Variablen aus der Blockdefinition.

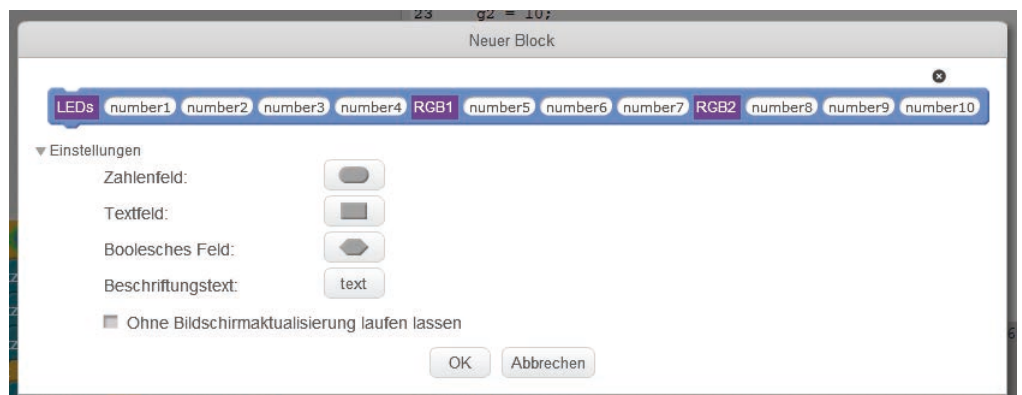
Der neue Block erscheint auf der Blockpalette **Daten&Blöcke** und kann wie jeder andere Block im Programm verwendet werden. Die Felder für die Parameter müssen mit den gewünschten Werten gefüllt werden.

Im Programm 21mBlock enthält der selbstdefinierte Block vier Felder, die mit 0 oder 1 gefüllt werden können und die einzelnen LEDs leuchten lassen. Dazu kommen zweimal drei Felder für die drei Farben der beiden RGB-LEDs, die nach dem gleichen Schema angesteuert werden.

In der Endlosschleife wird der Block sechsmal hintereinander aufgerufen, wobei jedes Mal andere Kombinationen von LEDs leuchten sollen.

Der neue Block fragt bei jedem Aufruf den Sensorkontakt am Pin **A3** ab. Wird dieser berührt, werden LEDs entsprechend der ersten vier Parameter ein- bzw. ausgeschaltet. Solange der Sensorkontakt nicht berührt wird, bleiben alle vier LEDs aus, unabhängig davon, welche Werte dem Block beim Aufruf übergeben wurden. Durch diese Methode läuft das Lauflicht im Hintergrund ständig weiter, ist aber nur zu sehen, wenn der Sensorkontakt berührt wird.

Auf die gleiche Weise leuchten die RGB-LEDs in unterschiedlichen Farben, solange der Sensorkontakt am Pin **A1** berührt wird. Am Ende des Blocks wartet das Programm die in der Variable **zeit** festgelegte Zeit. So braucht der Warte-Block nicht nach jedem einzelnen Blockaufruf im Hauptprogramm eingetragen zu werden.

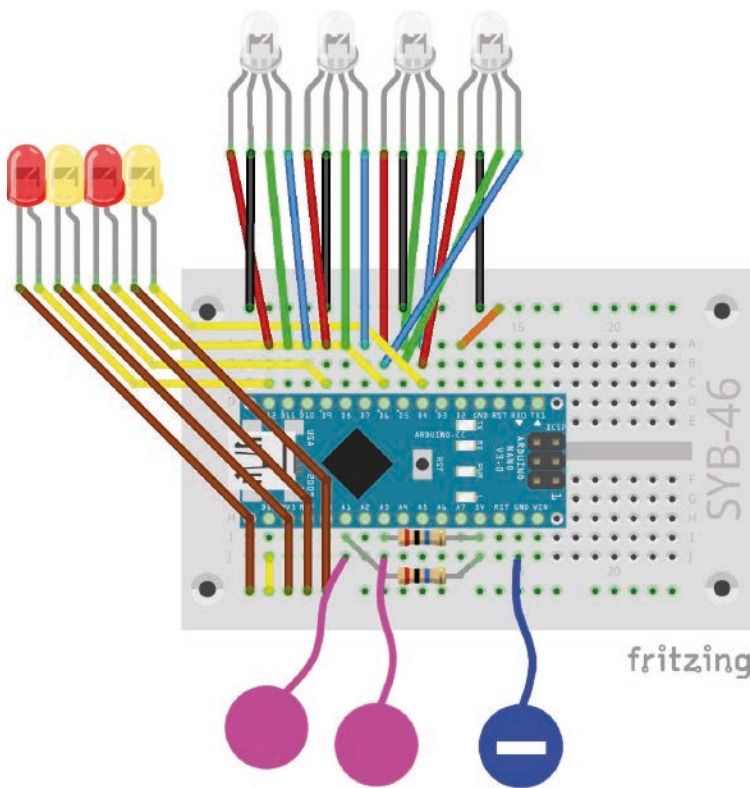


Neuen Block definieren.

22. Tag

Heute im Adventskalender

- 4 x Anschlusskabel



Laufflicht mit vier LEDs, vier RGB-LEDs und Sensorkontakten.

automatisch immer andere Farben, außer dem reinen Grün.

Die vier einzelnen LEDs sind an den gleichen Pins angeschlossen wie die roten Farben der vier RGB-LEDs, lassen sich aber durch die Kathodenschaltung getrennt schalten. Die beiden Variablen **rgb** und **led** enthalten die Pin-Nummern **2** und **13** für die Kathoden der RGB-LEDs und der einzelnen LEDs.

Anschließend ruft das Hauptprogramm in einer Endlosschleife sechsmal hintereinander den selbstdefinierten Block auf, jedes Mal mit einer anderen Kombination einzuschaltender Pins.



Das Programm 22mblock lässt mit einem selbstdefinierten Block LEDs blinken.

Mehr LEDs als Arduino-Pins

Das Experiment des 22. Tages zeigt, wie man durch Schaltung der Kathoden mehr LEDs anschließen kann, als digitale Ausgangspins verfügbar sind. Die Kathoden der LEDs sind nicht direkt mit dem GND-Pin verbunden, sondern über die beiden äußeren Schienen des Steckbretts mit den digitalen Pins 2 und 13. Diese werden auf **LOW** gesetzt, um als Masseleitung für die LEDs zu dienen und diese einzuschalten. Solange einer dieser Ausgänge auf **HIGH** steht, leuchten die mit der Kathode angeschlossenen LEDs nicht.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 4 x RGB-LED mit Vorwiderstand, 2 x 20-MOhm-Widerstand (rot - schwarz - blau), 3 x Knetkontakt, 2 x Drahtbrücke, 24 x Anschlusskabel

Das Programm

Das Programm 22mblock schaltet auch wieder über einen selbstdefinierten Block vier LEDs und vier RGB-LEDs. Auch in diesem Programm lassen sich zwei Gruppen von LEDs über Sensorkontakte ein- und ausschalten.

So funktioniert das Programm

Das Programm 22mblock funktioniert ähnlich wie das Programm des 21. Tages mit einem selbstdefinierten Block.

Das Hauptprogramm initialisiert am Anfang neun Ausgangspins **4...12** für drei RGB-LEDs. Die vierte RGB-LED ist genau umgekehrt zur dritten an die gleichen Pins angeschlossen und zeigt damit

Der neue Block funktioniert anders als im vorhergehenden Programm. Alle Anoden der LEDs werden entsprechend den im Blockaufruf übermittelten Mustern ein- bzw. ausgeschaltet. Die LEDs leuchten aber noch nicht, da die Kathoden auf **HIGH** gesetzt sind.

Jetzt werden die beiden Sensorkontakte abgefragt. Wird der Sensor am Pin **A1** berührt, wird der Pin **13** auf **LOW** gesetzt, damit leuchten die einzelnen LEDs, je nachdem ob die Anoden ein- oder ausgeschaltet sind. Auf die gleiche Weise schaltet der Sensorkontakt am Pin **A3** die RGB-LEDs.

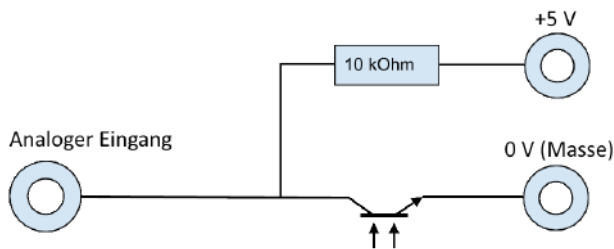
23. Tag

Heute im Adventskalender

• 1 x Fototransistor

Fototransistor

Ein Fototransistor ist ein lichtempfindliches Bauelement, das auf den ersten Blick wie eine transparente LED aussieht. Je nach Stärke des Lichteinfalls lassen sich mit der abgebildeten Schaltung an einem analogen Eingang des Nano-Boards unterschiedliche Werte erzielen. Je heller das Licht auf den Fototransistor fällt, desto geringer wird der Wert am analogen Eingang. Anders als bei LEDs wird beim Fototransistor der lange Anschluss mit Masse verbunden, nicht der kurze.

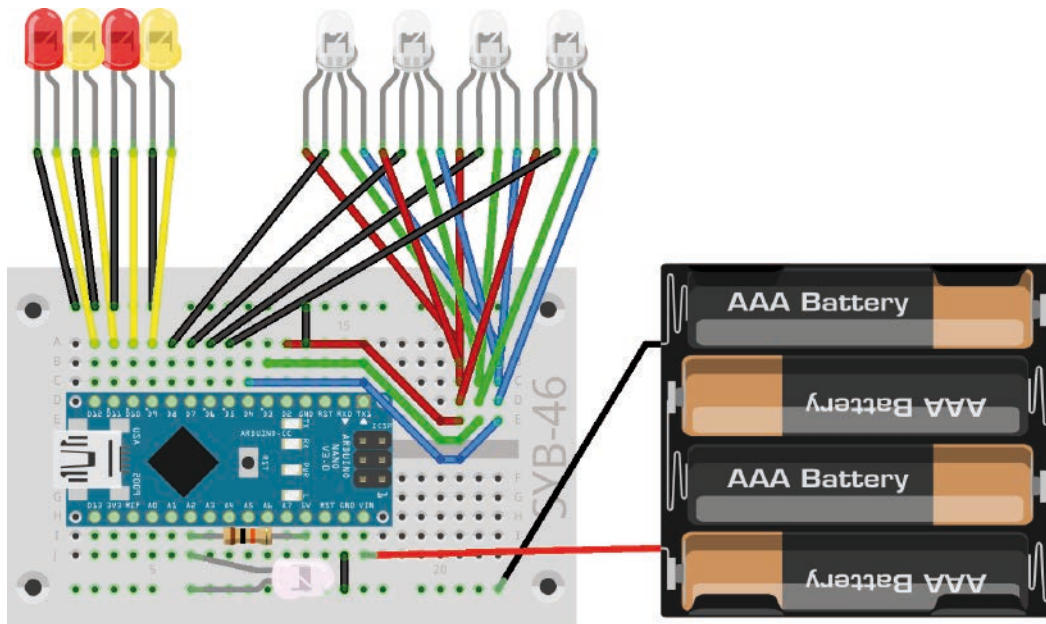


Schaltschema für einen Fototransistor.

Weihnachtliche Lichteffekte in der Dunkelheit

Das Experiment des 23. Tages lässt LEDs bunt blinken, wenn es dunkel genug ist. Haben Sie das Steckbrett auf die Rückseite des Engelsflügels geklebt, schließen Sie nach der Programmierung noch den Batteriekasten an und tragen den Engelsflügel in eine dunkle Ecke. Dann beginnen die LEDs zu blinken.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 4 x RGB-LED mit Vorwiderstand, 1 x 10-kOhm-Widerstand (braun - schwarz - orange), 5 x Drahtbrücke (unterschiedliche Längen), 24 x Anschlusskabel, 1 x Batteriekasten



fritzing

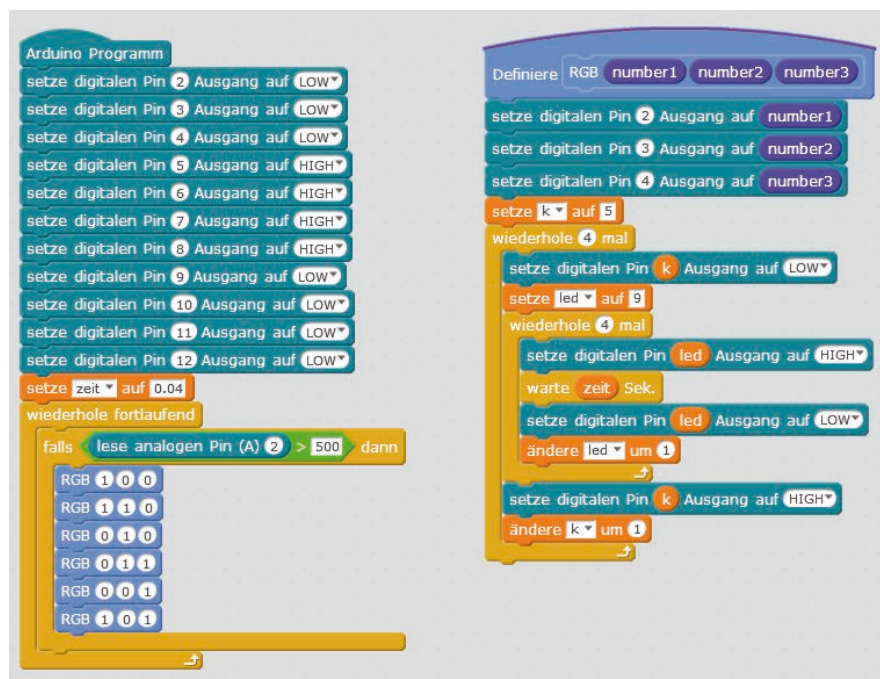
Lauflicht mit vier LEDs, vier RGB-LEDs und Fototransistor.

Die Schaltung sieht nur auf den ersten Blick unübersichtlich aus, alle roten Anschlüsse der vier RGB-LEDs sind in der Reihe 21 des Steckbretts angeschlossen und über eine Drahtbrücke mit dem Pin 2 verbunden, die grünen Anschlüsse sind in der Reihe 23 angeschlossen und mit Pin 3 verbunden, die blauen Anschlüsse sind in der Reihe 24 angeschlossen und mit Pin 4 verbunden. Die Kathoden der RGB-LEDs sind an den Pins 4, 5, 6, 7 angeschlossen.

Das Programm

Das Programm `23mb1ock` zeigt, wie durch getrenntes Schalten von Anoden und Kathoden der RGB-LEDs Anschlusspins eingespart werden können. Für vier RGB-LEDs werden nur sieben (3 x Anode, 4 x Kathode) statt 13 Pins (12 x Anode, 1 x Kathode) verwendet.

23. Tag



Das Programm 23mb1ock zeigt bei Dunkelheit Lichteffekte.

So funktioniert das Programm

Nachdem alle verwendeten Ausgangspins initialisiert worden sind, wartet die Hauptschleife des Programms darauf, dass der analoge Eingang A2 einen Wert größer als 500 annimmt. Der Fototransistor liefert je nach Helligkeit einen analogen Wert, der bei der verwendeten Schaltung umso höher ist, je dunkler die Umgebung ist. Den Wert 500 können Sie je nach Umgebung im Bereich zwischen 1 und 1023 natürlich auch beliebig verändern.

Befindet sich der Fototransistor in dunkler Umgebung, läuft sechsmal hintereinander ein selbstdefinierter Block, der die RGB-LEDs in unterschiedlichen Farben als Lauflicht leuchten lässt. Die gewünschten Farben werden als RGB-Parameter bei jedem Aufruf an den Block übergeben.

Bei jedem Aufruf des neuen Blocks werden zuerst die drei für die Anoden der RGB-LEDs verwendeten Pins entsprechend der gewünschten Farbe ein- oder ausgeschaltet. Da die Kathoden alle auf **HIGH** stehen, leuchten die RGB-LEDs aber noch nicht.

Die Variable **k** enthält die Nummer der zu schaltenden Kathode. Diese wird am Anfang für die erste RGB-LED auf **5** gesetzt. Eine Schleife läuft viermal und erhöht diese Variable jedes Mal um 1, um die vier RGB-LEDs nacheinander als Lauflicht anzusteuern. Dazu wird die entsprechende Kathode auf **LOW** gesetzt und nach kurzer Zeit wieder zurück auf **HIGH**. Während dieser Zeit leuchtet die RGB-LED in der eingestellten Farbe.

Solange eine der RGB-LEDs leuchtet, blinken die einzelnen LEDs in schneller Folge nacheinander kurz auf, was gleichzeitig die Leuchtdauer einer RGB-LED bestimmt. Hier wird eine ähnliche Schleife verwendet, die in der Variable **led** nacheinander die vier Anoden der einzelnen LEDs einschaltet und nach den in der Variable **zeit** gespeicherten 0.04 Sekunden wieder ausschaltet.

Wird der Fototransistor wieder beleuchtet, läuft die aktuell laufende Blinksequenz, die immer mit Rot beginnt und mit Lila endet, noch komplett zu Ende durch. Der Fototransistor wird erst zu Beginn des nächsten Schleifendurchlaufs wieder abgefragt.

24. Tag



Heute im Adventskalender

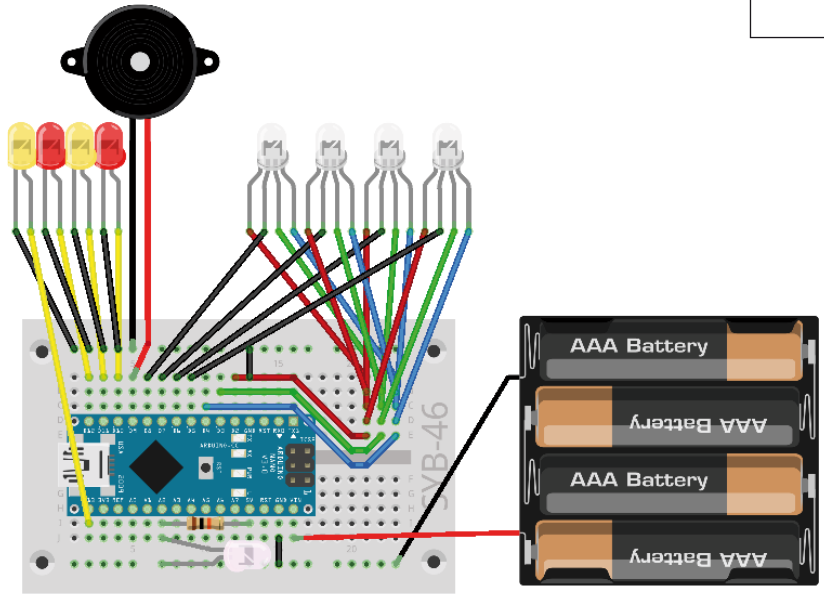
- 1 x Piezo-Summer

Piezo-Summer

Der heute im Adventskalender enthaltene Piezo-Summer macht elektrische Schwingungen hörbar. Legt man eine pulsierende Gleichspannung zwischen die beiden Pole des Summers, wird er in Schwingung versetzt. Je nach Frequenz sind einzelne Klicks oder ein durchgängiger Ton zu hören. Frequenzen von wenigen Hertz (Schwingungen pro Sekunde) nimmt das menschliche Ohr noch als einzelne Töne wahr, Frequenzen zwischen etwa 20 Hertz und 16 kHz werden als durchgehender Ton unterschiedlicher Tonhöhe wahrgenommen.

Weihnachtsbeleuchtung mit Musik

Der Piezo-Summer wird an einen PWM-Pin angeschlossen. Verändern Sie die Schaltung von gestern dazu wie in der Abbildung. Die einzelnen LEDs nutzen jetzt die Pins 10...13. So wird der PWM-fähige Pin 9 für den Piezo-Summer frei. Der restliche Schaltungsaufbau entspricht dem 23. Tag.



Weihnachtsbeleuchtung mit Musik.

Bauteile: 1 x Nano-Board, 1 x Steckbrett, 2 x LED orange mit Vorwiderstand, 2 x LED gelb mit Vorwiderstand, 4 x RGB-LED mit Vorwiderstand, 1 x 10-kOhm-Widerstand (braun - schwarz - orange), 1 x Piezo-Summer, 5 x Drahtbrücke (unterschiedliche Längen), 24 x Anschlusskabel, 1 x Batteriekasten

Das Programm

Das Programm `24mb1ock` lässt die LEDs auf dem Engelsflügel ähnlich wie im Programm des 23. Tages blinken. Dazu wird auf dem Piezo-Summer ein Weihnachtslied abgespielt.

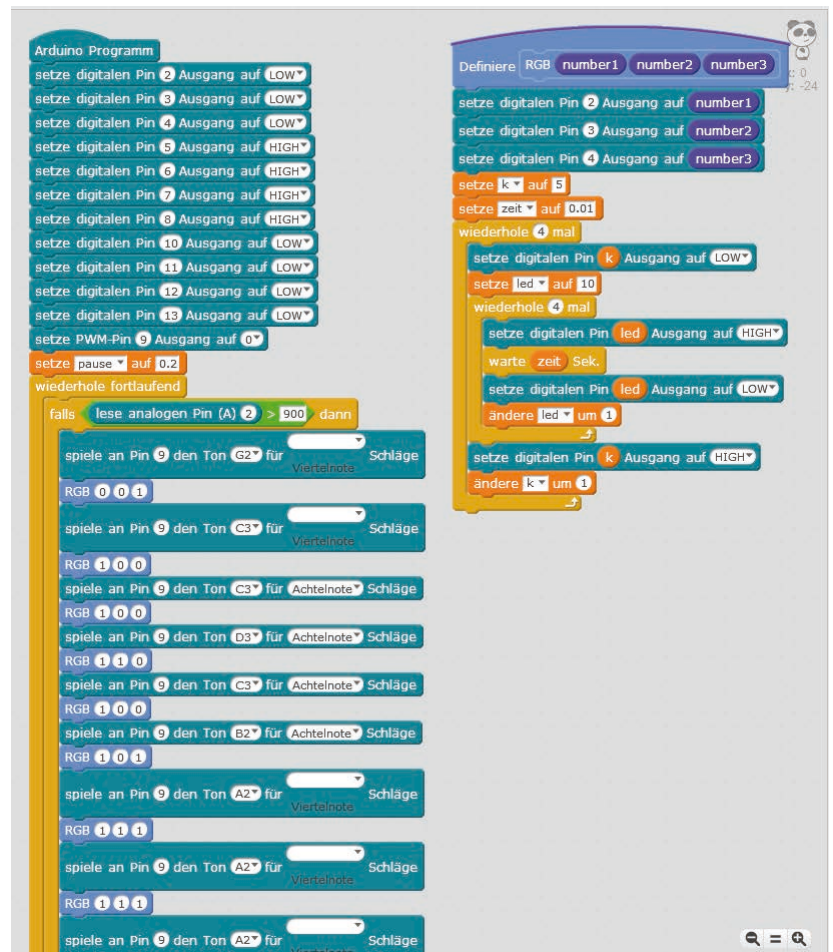
So funktioniert das Programm

Befindet sich der Engelsflügel mit dem Fototransistor im Dunkeln, werden eine Reihe von **spiele an Pin 9 den Ton ... für ... Schläge** aufgerufen. Dieser Block spielt einen bestimmten Ton auf einem Piezo-Summer an einem PWM-Pin ab, ohne dass man die genaue Frequenz für das PWM-Signal kennen muss. Wegen eines Designfehlers in **mBlock 3** werden die Blöcke für Viertelnoten zweizeilig angezeigt. Es handelt sich aber um den gleichen Block, der auch für die anderen Notenlängen verwendet wird. Zusätzlich wird der selbstdefinierte Block aufgerufen, der einen kurzen Lauflichteffekt mit einer für den Ton charakteristischen Farbe auf den RGB-LEDs zeigt. Am Ende des Liedes macht das Programm eine kurze Pause, damit der nächste Liedanfang nicht nahtlos nach dem Ende des vorherigen Liedes folgt.

We wish you a merry Christmas

Wer diese Melodie in der Vorweihnachtszeit fast täglich in Einkaufszentren und auf Weihnachtsmärkten hört, vermutet dahinter ein modernes amerikanisches Weihnachtslied. Tatsächlich handelt es sich aber um ein Lied aus der englischen Weihnachtstradition, das bis ins Jahr 1500 zurückverfolgt werden kann und somit heute urheberrechtsfrei genutzt werden kann.

Frohe Weihnachten!



Die ersten Blöcke des Programms zeigen, wie das Weihnachtslied abgespielt wird.

Vorsichtsmaßnahmen

Auf keinen Fall irgendwelche Pins miteinander verbinden und abwarten, was passiert.

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

Einige Pins sind direkt mit Anschlüssen des Mikrocontrollers verbunden, ein Kurzschluss kann den Arduino komplett zerstören. Verbindet man über einen Schalter oder eine LED zwei Pins miteinander, muss immer ein Schutzwiderstand dazwischengeschaltet werden.

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

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.

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.

© 2018 Franzis Verlag GmbH, Richard-Reitzner-Allee 2, 85540 Haar

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.

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.

Arduino ist ein eingetragenes Markenzeichen der Arduino AG