

Impressum

© 2021 Franzis Verlag GmbH, Richard-Reitzner-Allee 2, 85540 Haar bei München
Änderungen, Innovationen und Druckfehler vorbehalten
www.franzis.de

2021/01

Autor: Christian Immler
Produktmanagerin: Maria Siegmantel
Copy Editor: Claudia Fliedner
Coverdesign: Thomas Preischl
Design Inhalt: DTP-Satz A. Kugge, München
GTIN: 4019631673442

Bildnachweis:

Screenshots erstellt auf <https://scratch.mit.edu/>
Zeichnungen erstellt mit <http://fritzing.org>
Cover: /shutterstock.com
Motiv Lebkuchenhaus: /shutterstock.com
Alle anderen Abbildungen: Christian Immler

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

Die meisten Produktbezeichnungen 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 Bauanleitungen und Tipps wurden mit der größtmöglichen Sorgfalt entwickelt, geprüft und getestet. Trotzdem können Fehler im Buch und im Bausatz 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.



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 das Produkt deshalb genau so auf, wie es in der Anleitung beschrieben wird. Das Produkt darf nur zusammen mit der Anleitung und diesem Hinweis 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.

Sicherheits- und Warnhinweise

Hinweise an die betreuenden Erwachsenen

Hinweise und Regeln beim Aufbau der Bastelprojekte sind unbedingt einzuhalten. Nicht alle Kinder haben die gleichen Fertigkeiten. Auch in einer vergleichbaren Altersgruppe sollte der betreuende Erwachsene für die Kinder geeignete Experimente auswählen. Die Versuchsanweisungen helfen dabei zu beurteilen, ob Experimente geeignet sind. Vor dem Experiment sollte der betreuende Erwachsene mit den Kindern die erforderlichen Warnungen und Sicherheitsvorkehrungen durchsprechen.

ACHTUNG! Nicht für Kinder unter 3 Jahren geeignet. Es besteht Erstickungsgefahr, da Kleinteile verschluckt oder eingeatmet werden können.

ACHTUNG! Ausschließlich für Kinder von mindestens 10 Jahren geeignet. Anweisungen für Eltern oder andere verantwortliche Personen sind beigefügt und müssen beachtet werden. Verpackung und Anleitung müssen aufbewahrt werden, da sie wichtige Informationen enthalten.

ACHTUNG! Dieses Spielzeug hat funktionsbedingte scharfe Spitzen. Verletzungsgefahr!

ACHTUNG! Keine Experimente an Steckdosen durchführen! Die 230 Volt des Stromnetzes sind lebensgefährlich! Alle Experimente dieses Experimentierpakets dürfen nur unter Anschluss des Mikrocontrollers an einen PC durchgeführt werden. Dann besteht keine Gefahr bei Berührung elektrisch leitender Teile.

ACHTUNG! Benutzung nur unter unmittelbarer Aufsicht von Erwachsenen.

ACHTUNG! Augenschutz und LEDs: Blicken Sie nicht aus geringer Entfernung direkt in die LEDs, denn ein direkter Blick kann Netzhautschäden verursachen! Die scheinbare Helligkeit der LED gibt einen falschen Eindruck von der tatsächlichen Gefahr für die Augen.

ACHTUNG! Kurzschlüsse vermeiden! Eine direkte Verbindung zwischen Minus- und Pluspol muss unbedingt vermieden werden, weil dabei Drähte und Batterien heiß werden können. Im Extremfall können Drähte glühend heiß werden. Es besteht Brand- und Verletzungsgefahr.

Inhaltsverzeichnis

Adventskalender programmieren lernen.....	6
Tag 1.....	7
Heute im Adventskalender	7
mBlock 5.....	10
Aufgabe	10
So funktioniert das Programm	10
Tag 2.....	15
Heute im Adventskalender	15
Nano-Board – Arduino-kompatible Platine	16
Nano vorbereiten	17
Softwareinstallation in Kürze.....	17
Aufgabe	20
LED blinkt	21
So funktioniert das Programm	21
Tag 3.....	22
Heute im Adventskalender	22
Aufgabe	25
Das Haus in mBlock 5.....	26
Klick auf das Fenster schaltet das Licht ein	28
Tag 4.....	31
Heute im Adventskalender	31
Aufgabe	31
LED blinkt mit einstellbarer Geschwindigkeit	32
So funktioniert das Programm	33
Tag 5.....	34
Heute im Adventskalender	34
Aufgabe	34
Wechselblinklicht.....	34
So funktioniert das Programm	34
Tag 6.....	35
Heute im Adventskalender	35
LEDs dimmen	35
Aufgabe	36
Das Programm	36
So funktioniert das Programm	37

Tag 7	38
Heute im Adventskalender	38
Aufgabe	38
Das Programm	38
So funktioniert das Programm	39
Tag 8	42
Heute im Adventskalender	42
Aufgabe	42
Das Programm	42
So funktioniert das Programm	43
Tag 9	44
Heute im Adventskalender	44
Aufgabe	44
Das Programm	44
So funktioniert das Programm	45
Tag 10	46
Heute im Adventskalender	46
Aufgabe	47
Das Programm	48
So funktioniert das Programm	49
Tag 11	50
Heute im Adventskalender	50
Das Programm	51
So funktioniert das Programm	51
Tag 12	52
Heute im Adventskalender	52
Aufgabe	52
Das Programm	52
So funktioniert das Programm	53
Tag 13	55
Heute im Adventskalender	55
Aufgabe	56
Das Programm	57
So funktioniert das Programm	57

Tag 14.....	58
Heute im Adventskalender	58
Aufgabe	58
Das Programm	59
So funktioniert das Programm	59
Tag 15.....	60
Heute im Adventskalender	60
Aufgabe	60
Das Programm	60
So funktioniert das Programm	62
Tag 16.....	63
Heute im Adventskalender	63
Aufgabe	63
Das Programm	64
So funktioniert das Programm	64
Tag 17.....	65
Heute im Adventskalender	65
Aufgabe	65
Das Programm	65
So funktioniert das Programm	66
Tag 18.....	67
Heute im Adventskalender	67
Aufgabe	68
Das Programm im Arduino-Modus	68
So funktioniert das Programm	68
Programm auf den Nano übertragen.....	69
Tag 19.....	70
Heute im Adventskalender	70
Aufgabe	70
Das Programm	70
So funktioniert das Programm	70

Tag 20	71
Heute im Adventskalender	71
Aufgabe	71
Das Programm	72
So funktioniert das Programm	73
Tag 21	74
Heute im Adventskalender	74
Aufgabe	74
Taster mit Pull-up-Widerstand	74
Das Programm	75
So funktioniert das Programm	76
Tag 22	77
Heute im Adventskalender	77
Aufgabe	77
Steuerpult für das Haus im Vollbildmodus	78
Die Figuren	78
Die Variablen	81
Das Programm	81
So funktioniert das Programm	83
Tag 23	85
Heute im Adventskalender	85
Aufgabe	85
Weihnachtliche Lichteffekte in der Dunkelheit.....	86
So funktioniert das Programm	86
Tag 24	87
Heute im Adventskalender	87
Bunte Lichteffekte und Weihnachtslieder.....	87
Das Programm	88
Die Variablen	89
So funktioniert das Programm	90

Adventskalender programmieren lernen

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.

Und dank grafischer Programmiersprachen braucht man sich beim Programmieren keine Syntaxregeln mehr zu merken. Programme werden einfach am Bild-

schirm aus vorgefertigten Bausteinen zusammengesetzt. Damit lassen sich auch anspruchsvolle Programmieraufgaben spielerisch lösen.

Dieser Adventskalender bringt jeden Tag ein neues Programmierprojekt für dein eigenes kleines Smart Home ans Licht, das bis Weihnachten in vollem Glanz erstrahlen wird.

Hinweis:

Unter <https://www.franzis.de/programmieren-adventskalender> findest du die Website zu diesem Kalender mit allen Downloads. Bei Fragen und Problemen kannst du dich dort auch an unser Support-Team wenden.

Tag 1

HEUTE IM ADVENTSKALENDER

VORLAGEN FÜR DAS HAUS

Haus backen (1. Teil)

Auf der Rückseite des Adventskalenders findest du die Ausschneidevorlagen für das Haus. Hier kannst du schon erkennen, wie es aussehen wird.

Bitte beachte: Dein Lebkuchenhaus kommt in Kontakt mit Elektro-Bauteilen und darf deswegen leider **nicht gegessen** werden. Du kannst anstelle von Lebkuchen auch Salzteig verwenden. Salzteig funktioniert so ähnlich wie Knetmasse, lässt sich aber im Ofen backen und wird dann fest. Beide Anleitungen findest du auf den nächsten Seiten. Da beide Rezepte unterschiedliche Back- und Ruhezeiten haben, sind die Anleitungen leicht unterschiedlich. Davon nicht verwirren lassen!

Bitte lass dir in jedem Fall beim Backen von einem Erwachsenen helfen!

Lebkuchen

Für den Teig brauchst du:

- 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

Bringe die Butter in einem Topf auf kleiner Flamme zum Schmelzen. Sie darf nicht anbrennen. Rühre dabei den Honig und den Zucker mit ein, sodass es eine gleichmäßige Masse ergibt. Lass die Masse abkühlen und rühre die Eier, das Lebkuchengewürz und etwas Salz unter die Masse.

Mische Mehl (lass noch ein bisschen übrig, um am nächsten Tag die Arbeitsfläche zu bestäuben), Backpulver und Kakao und gib alles nach und nach unter Rühren zu der Masse. Dabei entsteht ein immer festerer Teig, den du zum Schluss mit der Hand kneten kannst. Knete so lange, bis alles gleichmäßig gemischt ist, und lass den Teig dann in einer mit einem Tuch abgedeckten Schüssel über Nacht ruhen.

Salzteig

Für den Teig brauchst du:

4 Tassen	Mehl
2 Tassen	Salz
2 Tassen	Wasser

Du kannst ganz normale Tassen verwenden, sie haben in etwa eine Füllmenge von 200 bis 250 ml. Solange du für alle Mengen mit der gleichen Tasse abmisst, ist die genaue Füllmenge egal.

Mische in einer Schüssel zunächst Mehl und Salz.



Gib nach und nach unter Rühren das Wasser hinzu und verknete alles zu einem festen Teig, bei dem alles gleichmäßig vermischt ist. Zum Kneten braucht man ein bisschen Kraft, deswegen kann das auch ein Erwachsener übernehmen.



Rolle den Teig auf einer mit Mehl bestäubten Arbeitsfläche etwa einen Zentimeter dick aus. Schneide die Vorlagen für die Teile des Hauses aus der Rückseite dieses Adventskalenders aus, lege sie auf die Lebkuchenplatte und schneide die Teile aus. Beachte dabei die Hinweise auf der Rückseite der Vorlagen. Du brauchst eine Bodenplatte, zwei Dächer, eine Vorderseite und eine Rückseite. Die Bodenplatte kann natürlich auch größer sein, wenn du um das Haus herum noch einen Garten gestalten möchtest.



Wenn du möchtest, kannst du aus den beiden Dachteilen Fenster ausstechen. Achte darauf, dass die Fenster mittig liegen und die Stege zwischen den Öffnungen nicht zu schmal sind. Sie brechen sonst leicht.

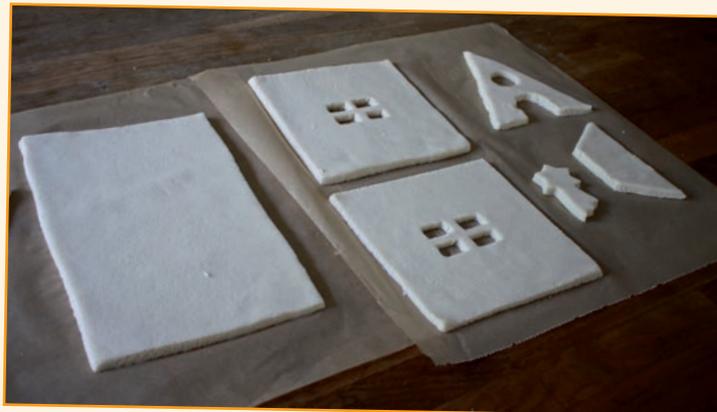


Die Vorderseite des Häuschens bekommt ein Fenster und eine Tür. Drücke für die spätere Türklingel eine kleine Aussparung in die Türöffnung. Drücke dazu zum Beispiel die Rückseite eines Messers vorsichtig in den Teig. Die Aussparung sollte etwa einen halben Zentimeter hoch und breit sein.



Aus dem übrigen Teig kannst du Bäume oder andere Figuren ausstechen. Knete noch einen kleinen Stern oder stich ihn aus, der sich später auf dem Haus drehen kann.

Lass alle Teile über Nacht trocknen.



MBLOCK 5



Für die Projekte im Adventskalender verwenden wir die einfach zu erlernende Programmiersprache *mBlock 5*. *mBlock* basiert auf dem bekannten Scratch. Lade dir die PC-Version von *mBlock 5* kostenlos bei www.mblock.cc/download herunter und installiere sie. *mBlock 5* funktioniert mit Windows 10, 8,1 und 7.

Starte nach der Installation das Programm *mBlock*. Jetzt erscheint die Programmieroberfläche mit dem typischen M-Panda, der Symbolfigur von *mBlock*.

Sollte *mBlock* nicht automatisch auf Deutsch starten, fahre auf der Menüleiste oben über das Weltkugelsymbol. Dort kannst du die Sprache einstellen.

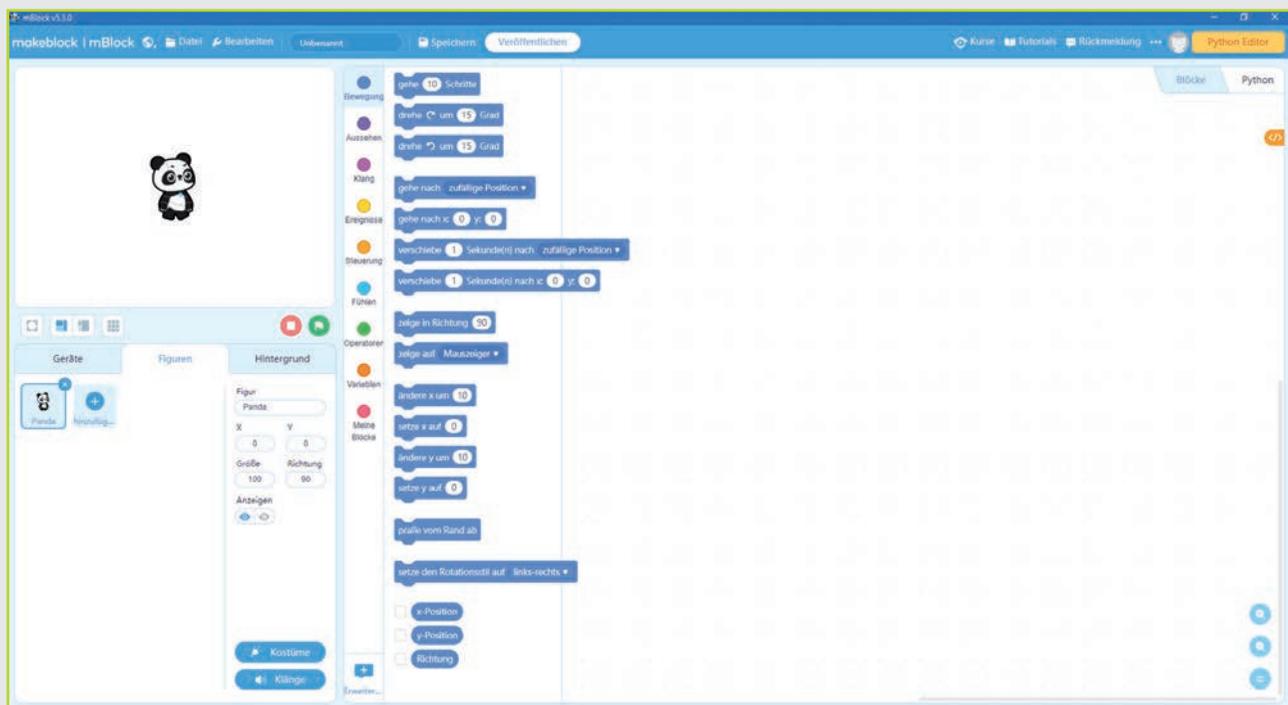
In *mBlock 5* braucht man keinen Programmcode zu schreiben. Die Programme werden aus puzzleteilartigen Blöcken zusammengesetzt. Diese stehen auf Blockpaletten, die farblich nach Themen geordnet sind, zur Auswahl bereit.

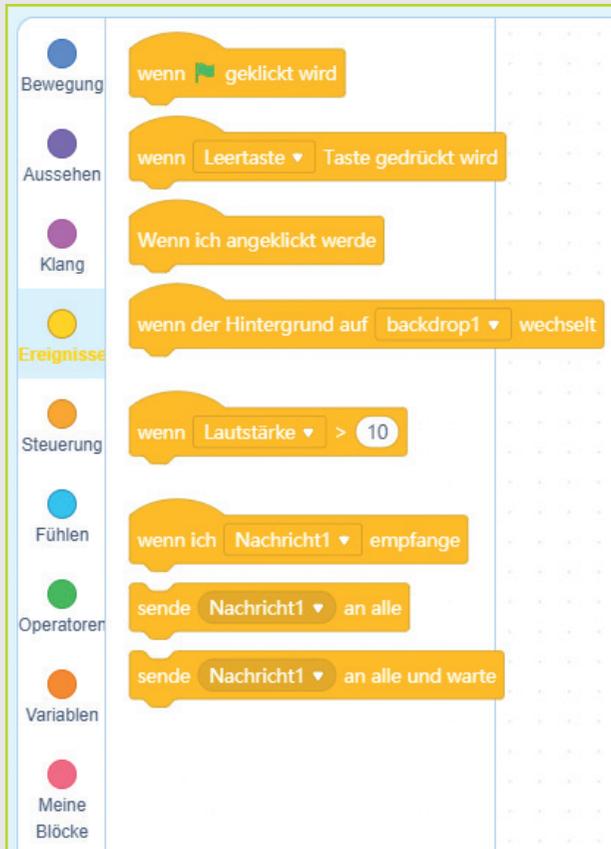
AUFGABE

In einem ganz einfachen Programm soll die Figur auf der sogenannten Bühne hin- und herlaufen und dabei ihre Farbe verändern. Heute brauchen wir noch keine Elektronik.

SO FUNKTIONIERT DAS PROGRAMM

1. Klicke in der Blockpalette auf das gelbe Symbol *Ereignisse*. Damit werden die Blöcke angezeigt, die auf bestimmte Ereignisse reagieren.

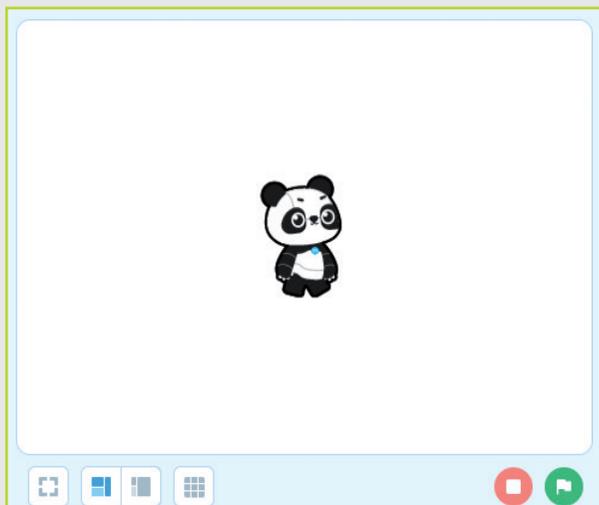




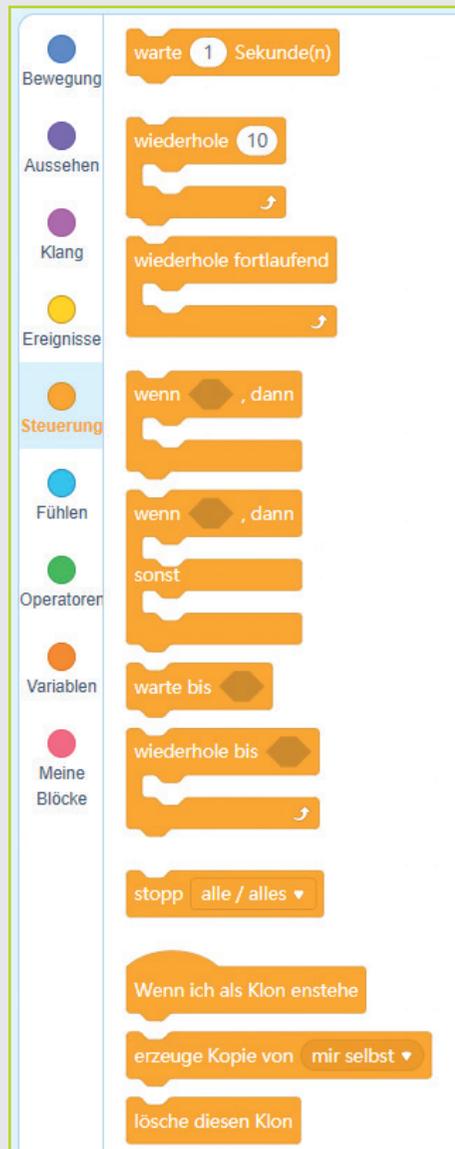
- Ziehe den abgebildeten Block *wenn Fahne geklickt wird* aus der Blockpalette in das graue Skriptfenster rechts.



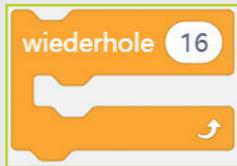
- Rechts unterhalb der Bühne ist ein weißes Fähnchen auf grünem Grund zu sehen. Es dient üblicherweise dazu, ein Programm zu starten. Die Bühne ist das Fenster links oben, auf dem sich in der Mitte die M-Panda-Figur befindet.



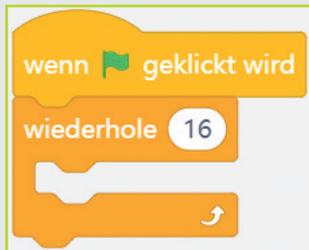
- Der Block *wenn Fahne geklickt wird* bewirkt, dass die folgenden Blöcke ausgeführt werden, wenn du auf das grüne Fähnchen klickst. Der Block ist oben rund, passt also unter keinen anderen Block. Er muss immer als Erstes gesetzt werden.
- Klicke jetzt in der Blockpalette auf das orange-farbene Symbol *Steuerung*.



- Die Bewegung der Figur wird aus einzelnen Schritten zusammengesetzt. Die Figur geht 16 Schritte in eine Richtung, dreht sich und geht wieder 16 Schritte zurück – und so weiter. Ziehe für die Wiederholungsschleife den Block *wiederhole ...* in das Skriptfenster und docke ihn unten an den dort bereits vorhandenen Block an. Wenn du zwei Blöcke nahe genug aneinanderziehst, verbinden sie sich automatisch.



7. Schreibe in das Zahlenfeld im Block *wiederhole ...* die Zahl 16.



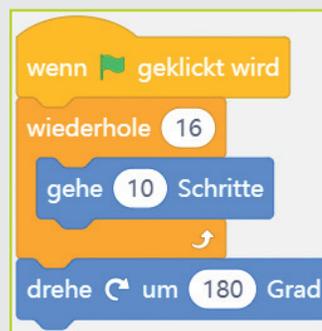
8. Bis jetzt haben wir zwar die Wiederholung programmiert, aber noch keine wirkliche Bewegung. Um eine Bewegung zu programmieren, klicke in der Blockpalette auf das blaue Symbol *Bewegung*.



9. Ziehe den Block *gehe ... Schritte* in das Skriptfenster und docke ihn innerhalb der Schleife an.



10. Nachdem die Figur 16-mal 10 Schritte gegangen ist, soll sie sich um 180 Grad drehen und wieder zurückgehen. Ziehe dazu einen Block *drehe um ... Grad* unter die Wiederholung. Ändere den Zahlenwert auf 180 Grad.



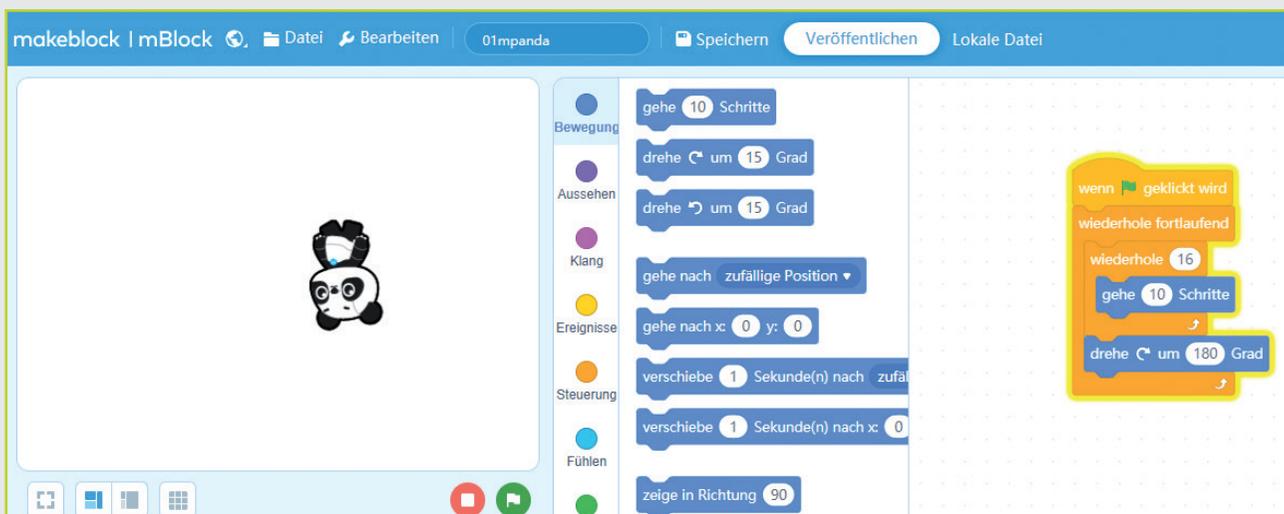
11. Bis jetzt geht die Figur 16-mal einen 10er-Schritt, dreht sich und bleibt stehen. Sie soll aber immer wieder hin- und zurückgehen. Trenne dazu zunächst die *wiederhole*-Schleife vom darüberliegenden Block *wenn Fahne geklickt wird*. Ziehe dann aus der Blockpalette *Steuerung* den Block *wiederhole fortlaufend* unter den Block *wenn Fahne geklickt wird*. Ziehe anschließend die Schleife *wiederhole 16* in die Klammer der *wiederhole fortlaufend*-Schleife.



12. Probiere aus, ob das Programm auch wie erwartet funktioniert. Klicke dazu unter der Bühne auf das grüne Fähnchen. Die Figur geht hin und zurück, allerdings steht sie auf dem Rückweg auf dem Kopf, und die gewünschte Farbveränderung erfolgt auch noch nicht. 
13. Halte das Programm mit einem Klick auf das rote Stoppsymbol neben dem grünen Fähnchen wieder an. 
14. Jede Figur kann einen eigenen Rotationsstil bekommen. Dieser legt fest, ob sich eine Figur im Kreis

dreht oder hin- und herbewegt. Um den Rotationsstil festzulegen, ziehe einen Block *setze den Rotationsstil auf ...* aus der Blockpalette *Bewegung* an den Anfang des Programms noch vor die *wiederhole fortlaufend*-Schleife. Wähle hier im Listenfeld den Rotationsstil *links-rechts*. Anstelle einer kreisförmigen Drehung dreht sich die Figur jetzt nur noch nach links oder rechts.

15. Um den Farbwechsel zu programmieren, klicke in der Blockpalette auf das violette Symbol *Aussehen*.

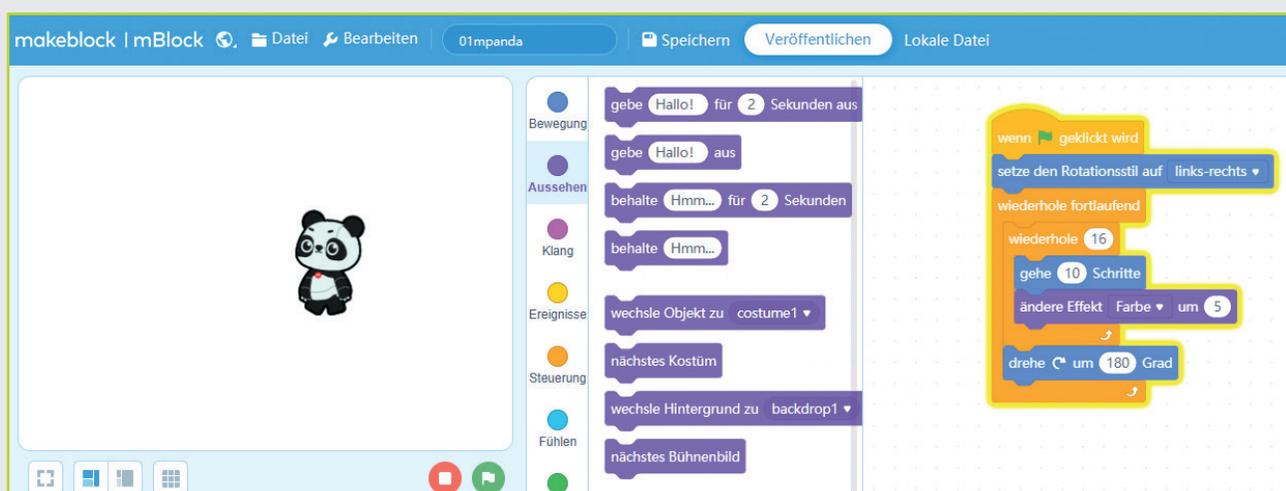




16. Ziehe einen Block *ändere Effekt ... um ...* in die *wiederhole 16*-Schleife unter den Block *gehe 10 Schritte*, damit sich die Farbe bei jedem Schritt etwas verändert. Schreibe in das Zahlenfeld des *ändere Effekt um ...*-Blocks die Zahl 5 und wähle im Listenfeld den Effekt *Farbe*.
17. Starte das Programm mit einem Klick auf das grüne Fähnchen erneut. Nun sollte alles wie erwartet laufen. 

Die Programme zum Adventskalender

Die verwendeten Programme gibt es auf der Webseite zu diesem Adventskalender: <https://www.franzis.de/programmieren-adventskalender>. Dort findest du einen Download-Link. Entpacke das ZIP-Archiv in ein persönliches Verzeichnis auf der Festplatte. Die Programme sind nach Tagen nummeriert. Das Programm von heute findest du dort unter dem Namen `01mpanda.mblock`. Im Menü *Datei* wählst du dann *Öffne Dateien von Deinem Computer*, um ein Programm zu öffnen. Du kannst die Programme aber auch einfach jeden Tag anhand der Abbildungen selbst zusammenbauen.



Tag 2

HEUTE IM ADVENTSKALENDER

- Nano-Board (Arduino-kompatible Platine)

Haus backen (2. Teil)

Lebkuchen

Nachdem der Teig über Nacht geruht hat, rolle ihn auf einer mit Mehl bestäubten Arbeitsfläche etwa einen Zentimeter dick aus. Schneide die Vorlagen für die Teile des Hauses aus der Rückseite dieses Adventskalenders aus, lege sie auf die Lebkuchenplatte und schneide die Teile aus. Beachte dabei die Hinweise auf der Rückseite. Du brauchst eine Bodenplatte, zwei Dächer, eine Vorderseite und eine Rückseite. Die Bodenplatte kann natürlich auch größer sein, wenn du um das Haus herum noch einen Garten gestalten möchtest.

Bitte einen Erwachsenen um Hilfe.

Aus dem übrigen Teig kannst du Bäume oder andere Figuren ausstechen. Knete noch einen kleinen Stern oder stich ihn aus, der sich später auf dem Haus drehen kann.



Lege die Teile vorsichtig auf ein Backblech mit Backpapier und backe sie bei 175 °C etwa 10 bis 15 Minuten.

Lass die Teile nach dem Backen über Nacht gut abkühlen.



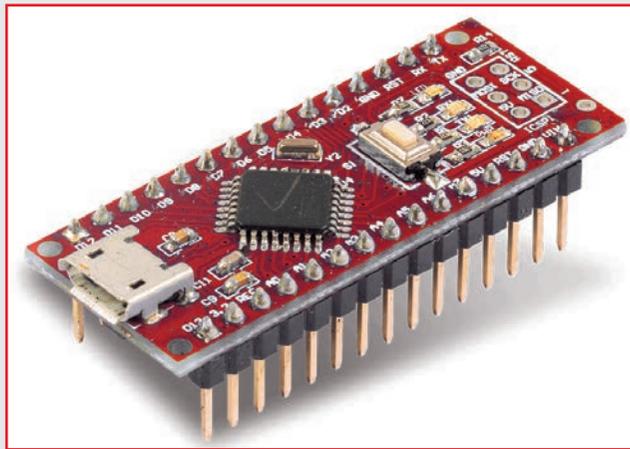
Salzteig

Bitte wieder einen Erwachsenen um Hilfe. Lege deine getrockneten Teile vorsichtig auf ein Backblech mit Backpapier und backe sie zunächst bei 60 °C etwa eine Stunde. Klemme dabei einen Holzlöffel in die Backofentür, sodass sie leicht geöffnet ist. Drehe die Teile nun einmal um, damit sie gleichmäßig von allen Seiten trocknen, erhöhe die Temperatur auf 150 °C und backe alles eine weitere Stunde fertig. Stich zwei Zahnstocher in den Stern, bevor du ihn backst. Du brauchst sie später, um den Stern in den Servo zu stecken.

Hast du Bäume oder andere Figuren aus Salzteig gemacht, backe einen der Bäume nicht mit. Lass ihn einfach nur trocknen. Du kannst ihn später bei der Türklingel verwenden. Lass alle Teile nach dem Backen über Nacht gut trocknen.

NANO-BOARD – ARDUINO-KOMPATIBLE PLATINE

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



Die mit D2 bis 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 bis A7 beschrifteten Pins sind analoge Eingänge. Davon kann eine Spannung zwischen 0 V und 5 V angelegt werden. Der Nano wertet sie aus und kann darauf mit bestimmten Aktionen antworten.

Der Name Arduino

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

Vorsichtsmaßnahmen

Auf keinen Fall sollte man irgendwelche Pins auf dem Nano miteinander verbinden und abwarten, was passiert.

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

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

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

NANO VORBEREITEN

Um den Nano in Betrieb zu nehmen, braucht man Folgendes:

- PC mit Windows 10, 8.1, 7
- MicroUSB-Kabel
- Treiber

Die Verbindung zwischen PC und Nano wird über ein MicroUSB-Kabel hergestellt. Du brauchst dir nicht extra ein solches Kabel zu besorgen, die meisten (besonders ältere) Smartphones verwenden diesen Steckertyp. Das Kabel wird gleichzeitig zur Stromversorgung wie auch zur Datenübertragung verwendet.

USB-Anschluss

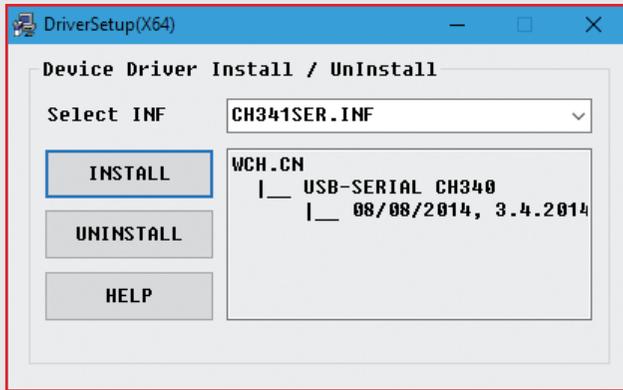
Schließe den Nano nach Möglichkeit an einem USB-2.0-Anschluss des PCs an, da es an USB-3.0-Anschlüssen eher zu Verbindungsproblemen kommen kann. Außerdem führen USB-3.0-Anschlüsse, bedingt durch die höhere Stromstärke, schneller zu einer Überhitzung des Nano. Zur besseren Unterscheidung sind USB-3.0-Anschlüsse meistens blau.

SOFTWAREINSTALLATION IN KÜRZE

Hier die Treiberinstallation in vier Schritten. Beende dazu mBlock, falls es gerade läuft:

1. Entpacke das ZIP-Archiv mit der heruntergeladenen Software in einen beliebigen Ordner auf der Festplatte – wenn du das nicht bereits getan hast. 
2. Entpacke das ZIP-Archiv 67006-9-nano-board-Treiber.zip in einen eigenen Ordner.

3. Schließe den Nano über das USB-Kabel an und starte dann die Treiberinstallation mit der Datei CH341SER.EXE aus dem Unterordner Windows des Treiberordners. Zur Installation musst du eine Abfrage der Windows-Benutzerkontensteuerung bestätigen.
4. Klicke im Installationsdialog auf *INSTALL* und warte, bis bestätigt wird, dass der Treiber installiert wurde.



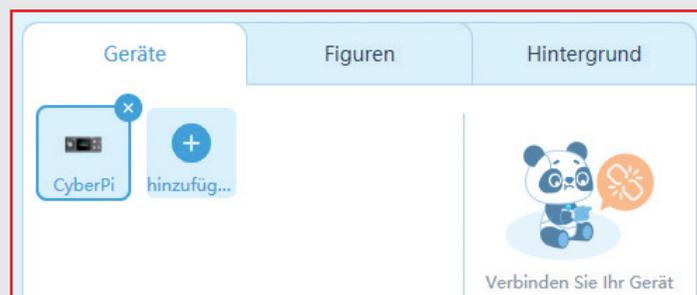
5. Starte jetzt mBlock erneut. Die Verbindung mit dem Nano wird standardmäßig von der Windows-Defender-Firewall blockiert. mBlock benötigt eine Berechtigung, die du mit einem Klick auf *Zugriff zulassen* in dem Dialogfeld, das automatisch erscheint, gewähren musst.

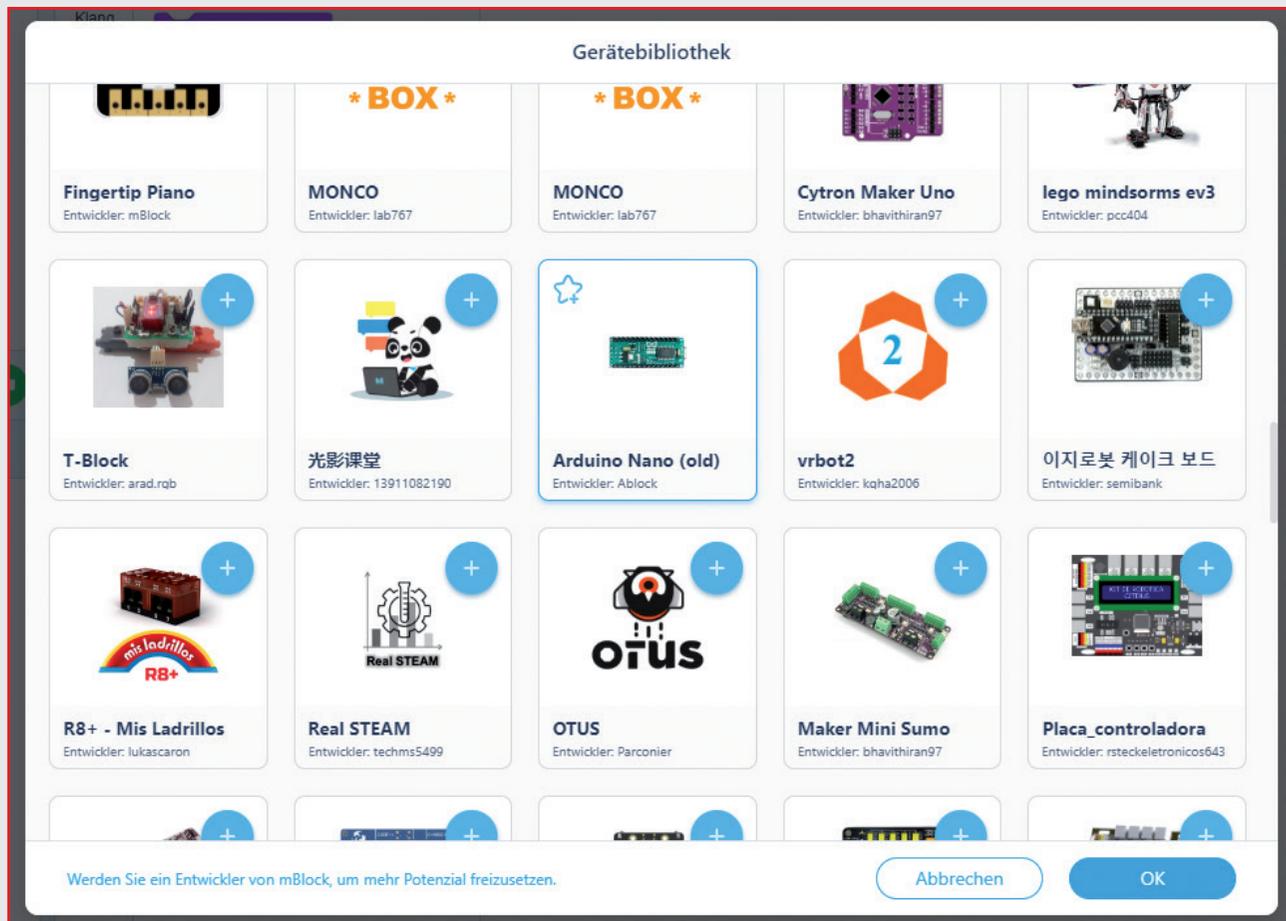
6. Falls du eine andere Firewall auf dem PC installiert hast, schalte sie aus. Viele externe Firewalls blockieren Verbindungen über USB-Kabel und bieten keine Möglichkeit, diese freizugeben, ohne die Firewall komplett abzuschalten.

Nach dem Treiber muss innerhalb von mBlock noch das richtige Gerät installiert werden.

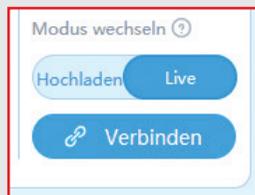
7. Unter *Geräte* im linken unteren Fenster ist ein Gerät *CyberPi* vorinstalliert. Entferne es mit einem Klick auf das x-Symbol.

8. Klicke auf das blaue Symbol *hinzufügen*. Jetzt öffnet sich die *Gerätebibliothek*. Suche hier das Gerät *Arduino Nano (old)*, markiere es, klicke auf das blaue Plusymbol, um die Software herunterzuladen, und installiere sie dann mit einem Klick auf *OK*.

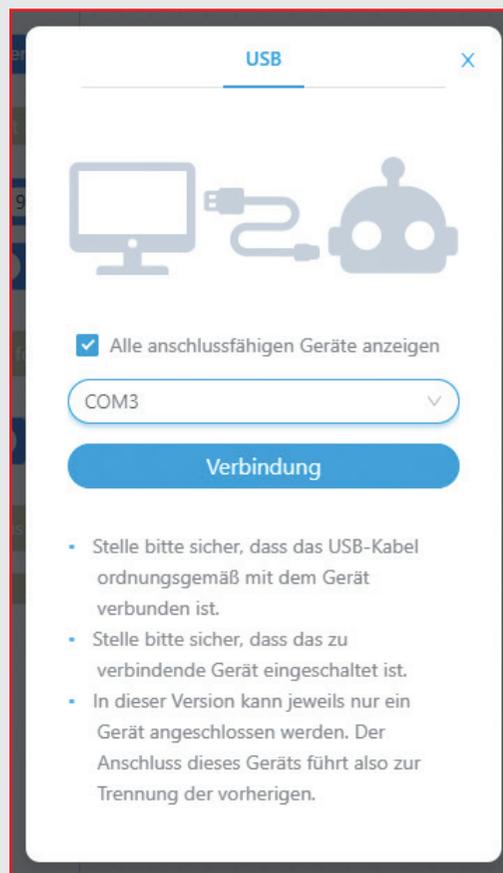




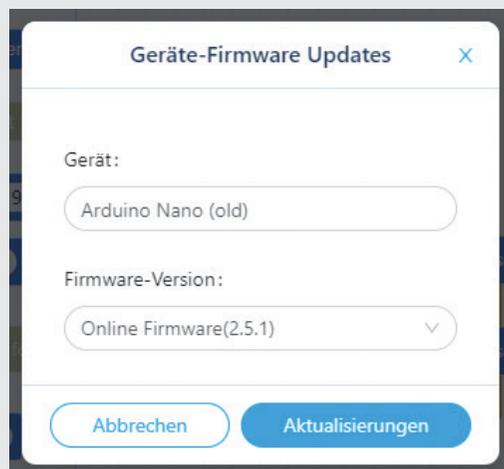
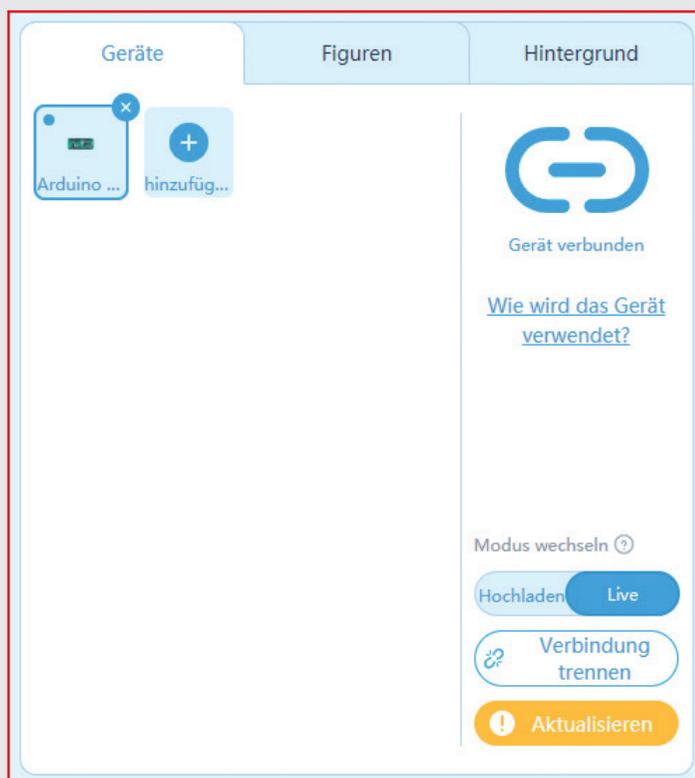
9. Schiebe im Fenster *Geräte* den Schalter unter *Modus wechseln* nun auf *Live* und klicke auf *Verbinden*.



10. Setze im nächsten Fenster ein Häkchen bei *Alle anschlussfähigen Geräte anzeigen*. Im Listenfeld werden ein oder mehrere COM-Ports zur Auswahl angeboten. Solange nur ein COM-Port angezeigt wird, wähle diesen aus. Werden mehrere angezeigt, wähle diesen aus und klicke auf *Verbindung*.



Wenn der richtige COM-Port ausgewählt wurde, wird der Nano nun als verbunden angezeigt.



11. Damit ein Programm aus mBlock heraus den Nano steuern kann, muss noch eine spezielle Firmware installiert werden. Dazu erscheint ein orangefarbener Button *Aktualisieren*. Klicke darauf und anschließend auf *Firmware aktualisieren*.
12. Im nächsten Fenster werden das Gerät *Arduino Nano (old)* sowie die aktuelle Firmwareversion angezeigt. Klicke auf *Aktualisierungen*.

13. Nachdem die Firmware aktualisiert wurde, klicke noch einmal auf *Verbinden*. Jetzt kann der Nano im interaktiven Modus von mBlock verwendet werden, um elektronische Bauteile vom PC aus zu steuern.

Firmware für mBlock neu installieren

Wenn du zwischendurch ein Programm im Arduino-Modus von mBlock oder ein anderes Arduino-Programm verwendet hast, musst du, die Firmware erneut aktualisieren, bevor du mBlock wieder interaktiv benutzen kannst.

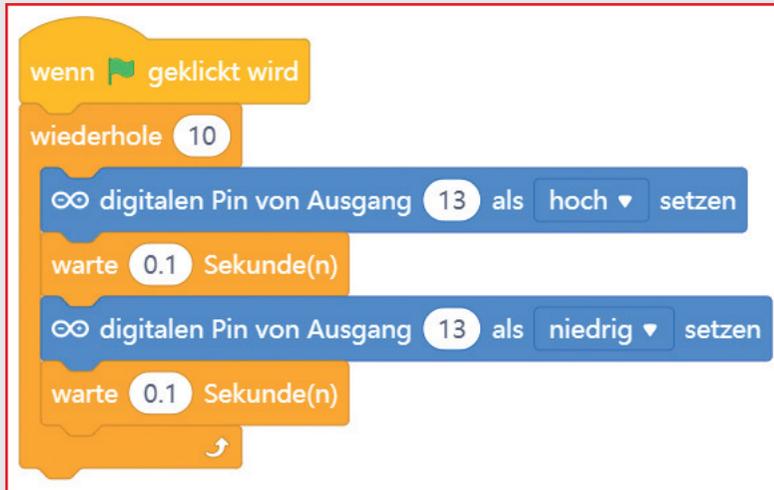
AUFGABE

Die mit „LED“ bezeichnete LED auf dem Nano soll zehnmal blinken.

LED BLINKT

Das Programm startet wie die meisten Programme in mBlock mit dem Block *wenn Fahne geklickt wird*. Danach folgt eine bereits bekannte Schleife *wiederhole 10*.

Der Block *digitalen Pin von Ausgang ... als ... setzen* von der Palette *Pin* setzt einen der digitalen Pins des Nano-Boards auf einen der Logikwerte *HIGH* oder *LOW* (in mBlock als *hoch* und *niedrig* bezeichnet). Die Werte können rechts im Listenfeld ausgewählt werden.



warte ... Sekunden von der Palette *Steuerung* lässt das Programm eine bestimmte Zeit bis zum nächsten Schritt warten. Erst nachdem diese Wartezeit abgelaufen ist, wird der nächste Programmschritt abgearbeitet.

Die LED an Pin D13

Für Statusanzeigen ohne Zusatzhardware hat der Nano eine eigene eingebaute LED, die als Pin D13 gesteuert wird.

Nano ausschalten

Der Nano hat keinen Ausschalter, du kannst einfach die Stromversorgung trennen, und er schaltet sich ab. Beim nächsten Einschalten startet automatisch das zuletzt gespeicherte Programm. Das Gleiche passiert, wenn man den Taster auf dem Nano drückt.

SO FUNKTIONIERT DAS PROGRAMM

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



Klicke auf das grüne Fähnchen. Wie erwartet, blinkt die mit „LED“ bezeichnete LED auf dem Nano zehnmal.

Nachdem die LED an Pin D13 eingeschaltet ist, wird 0,1 Sekunden gewartet – so lange leuchtet die LED. Danach wird die LED an Pin D13 wieder ausgeschaltet. Jetzt wartet das Programm erneut 0,1 Sekunden. Danach beginnt das Ganze wieder von vorne.



Das Programm ist unter dem Namen `02blink.mblock` bei den Downloads zu finden.

Dezimalpunkt statt Komma

mBlock verwendet, wie viele chinesische oder amerikanische Programme, den Punkt als Dezimaltrennzeichen, nicht das in Deutschland übliche Komma. Bedenke das bei der Eingabe von Zahlenwerten.

Tag 3

HEUTE IM ADVENTSKALENDER

- 1 Steckbrett (SYB 46)
- 1 LED gelb mit Vorwiderstand

Haus backen (3. Teil) Lebkuchen

Klebe die Teile des Hauses mit Zuckerguss zusammen. Für den Zuckerguss brauchst du:

- 1 Eiweiß
- 100 g Puderzucker

Schlage mit Hilfe eines Erwachsenen das Eiweiß schaumig und gib nach und nach den Puderzucker dazu, bis eine gleichmäßige klebrige Masse entsteht. Damit kannst du die Teile zusammenkleben.

Salzteig

Klebe mit Hilfe eines Erwachsenen die Teile mit Klebstoff zusammen. Gut geeignet ist ein flüssiger Bastelklebstoff oder, falls vorhanden, Heißkleber. Beachte die Anwendungshinweise des Klebstoffs, damit alles gut hält.

Das Haus bauen

Egal ob Salzteig oder Lebkuchen: Bitte einen Erwachsenen um Hilfe – es werden vier Hände gebraucht!

Klebe zuerst die Vorderseite auf die Bodenplatte. Halte die Vorderseite ein wenig fest, bis sie angetrocknet ist. Nun folgt das erste Dach. Halte die Vorderseite beim Kleben des Daches fest, damit nichts verrutscht.



Klebe nun die Rückseite an den anderen Teilen fest. Spare nicht mit Klebemittel und halte wieder alle Teile gut fest, bis sie angetrocknet sind.



Das zweite Dach macht das Häuschen komplett.

Lass oben am Dachfirst einen Spalt frei, um LEDs oder Drähte hindurchzustecken. Auf der Rückseite befindet sich eine Öffnung, durch die du später deine Bauteile ins Häuschen stellen kannst.

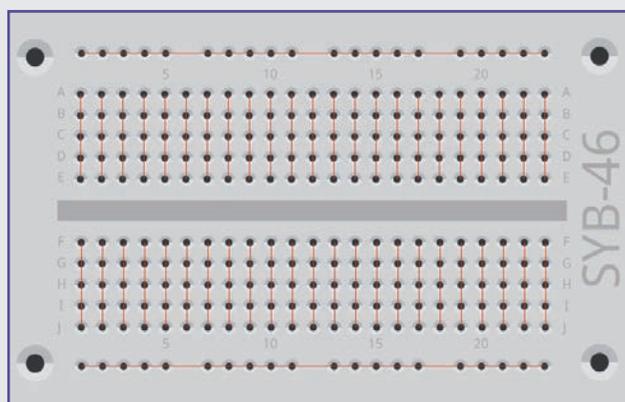
Zum Schluss kannst du das Haus noch beliebig verzieren. Das Salzteig-Häuschen kannst du mit Acrylfarben bemalen, wenn du möchtest.



STECKBRETT

Für den schnellen Aufbau elektronischer Schaltungen, ohne löten zu müssen, findest du heute ein Steckbrett im Adventskalender. Dort können elektronische Bauteile direkt in ein Lochraster eingesteckt werden.

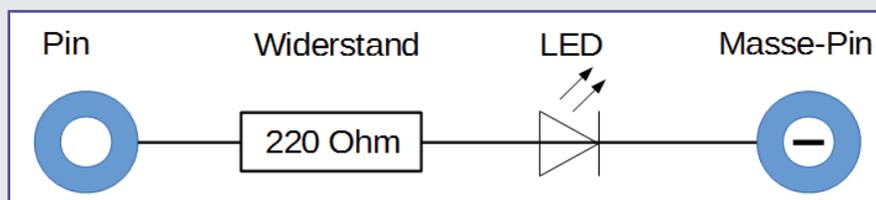
Bei diesem Steckbrett sind alle äußeren Längsreihen über Kontakte (X und Y) miteinander verbunden. Diese Kontaktreihen werden oft als Plus- und Minuspol zur Stromversorgung der Schaltungen genutzt. In den anderen Kontaktreihen sind jeweils fünf Kontakte (A bis E und F bis J) quer miteinander verbunden, 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 (englisch *Light Emitting Diode*, 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 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 Massepin – ein Vorwiderstand eingebaut werden. Dessen Wert errechnet sich aus dem maximalen Durchflussstrom durch die LED und beträgt bei Verwendung der Kontakte am Nano 220 Ohm. 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 ist der Minuspol, die Kathode. Einfach zu merken: Das Pluszeichen hat einen Strich mehr als das Minuszeichen und macht damit den Draht quasi optisch etwas länger. Außerdem sind die meisten LEDs auf der Minusseite abgeflacht, vergleichbar mit einem Minuszeichen. Leicht zu merken: Kathode = kurz = Kante.

AUFGABE

Eine LED im Haus soll vom PC eingeschaltet werden.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 LED gelb mit Vorwiderstand

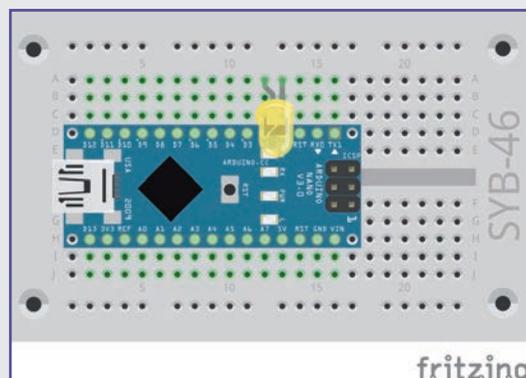
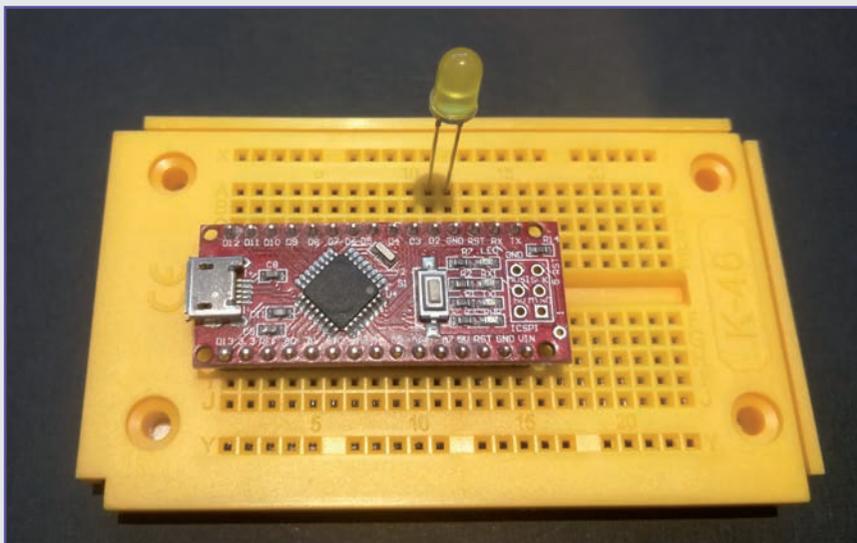
Stecke das Nano-Board wie in der Abbildung zu sehen auf das Steckbrett. Auf einem neuen Steckbrett lässt sich das Nano-Board oft nur schwer einstecken. Wende etwas gleichmäßigen Druck an, aber benutze keinerlei Werkzeug, also keinen Hammer oder Ähnliches. Das könnte die Elektronik beschädigen.

Achte 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.

Schiebe das Steckbrett ins Haus. In dunkler Umgebung dringt der Lichtschein der eingeschalteten LED später aus dem Fenster des Hauses.

Die Pins auf dem Nano

Alle Pins mit „D“ am Anfang sind digitale Ein- oder Ausgänge, die die Werte *wahr* oder *falsch* (ein oder aus, hoch oder niedrig) 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 Pluspins. An Pin 3.3 liegen +3,3 V Spannung an, an Pin 5V liegen +5 V Spannung an. Der Nano im Adventskalender benötigt für ein logisches *Wahr*-Signal +5 V, einige anderen Platinen nur +3,3 V.

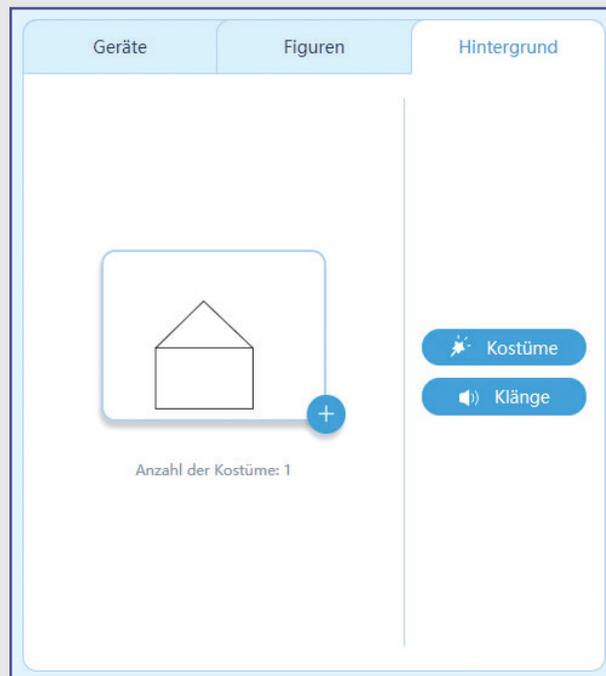


DAS HAUS IN MBLOCK 5

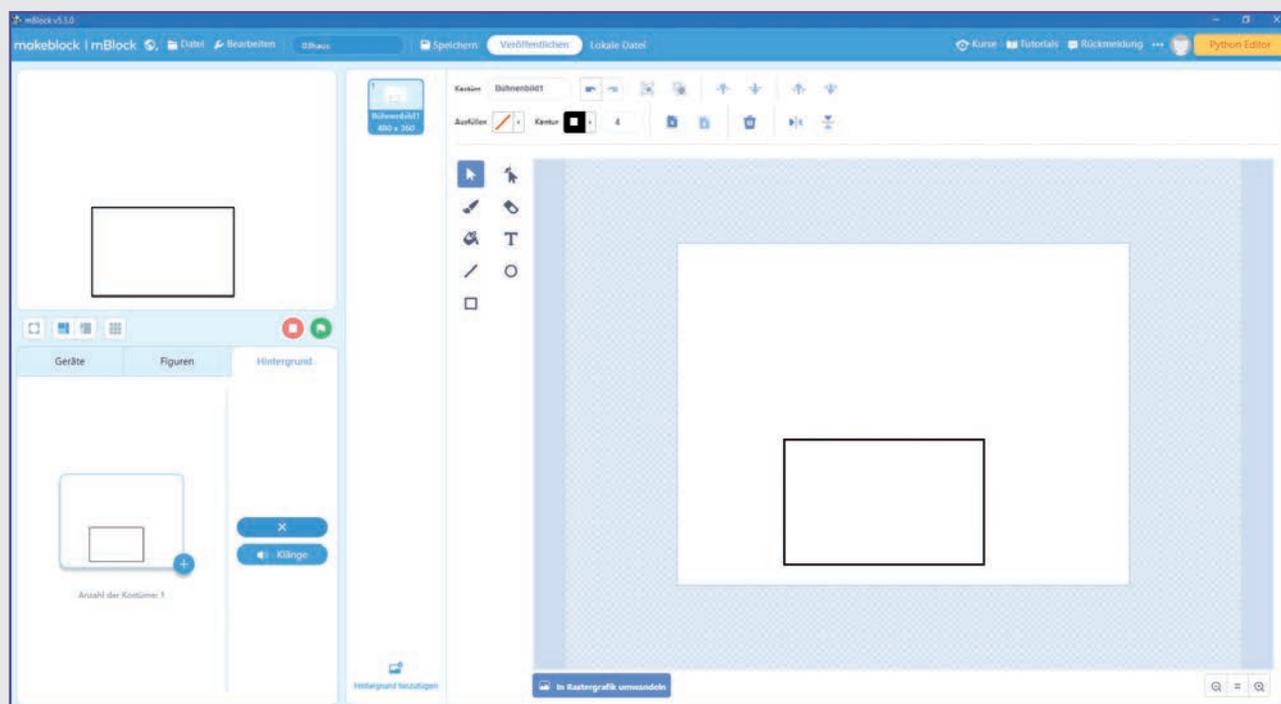
Auf der Bühne in mBlock 5 soll ein Haus zu sehen sein. Bei einem Klick auf das Fenster gehen das Licht im gezeichneten Haus sowie die LED im Lebkuchenhaus an.

Jedes Programm kann dazu verschiedene Hintergrundbilder enthalten.

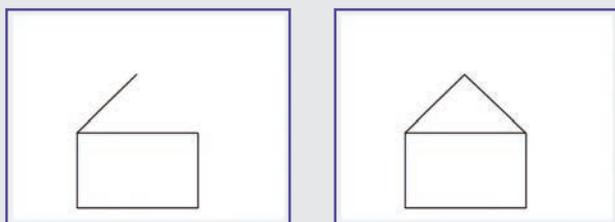
1. Wenn du im Fenster unten links auf die Registerkarte *Hintergrund* schaltest, kannst du das Hintergrundbild bearbeiten. mBlock 5 enthält ein Grafikprogramm für Vektorgrafik und Rastergrafik, mit dem sich Hintergrundbilder und neue Figuren gestalten lassen. Klicke auf den blauen Button *Kostüme*, um dieses Grafikprogramm zu öffnen.



2. Zeichne mit dem Rechteckwerkzeug ein Rechteck auf die Bühne. Schalte für das Rechteck den Füllmodus aus. Klicke dazu auf das Symbol *Ausfüllen* und dann unterhalb der Farbreger auf den roten Schrägstrich.



3. Zeichne anschließend mit dem Linienwerkzeug zwei Linien für das Dach.



4. Verlasse das Grafikprogramm mit einem Klick auf das blaue x-Symbol auf der Registerkarte *Hintergrund*.



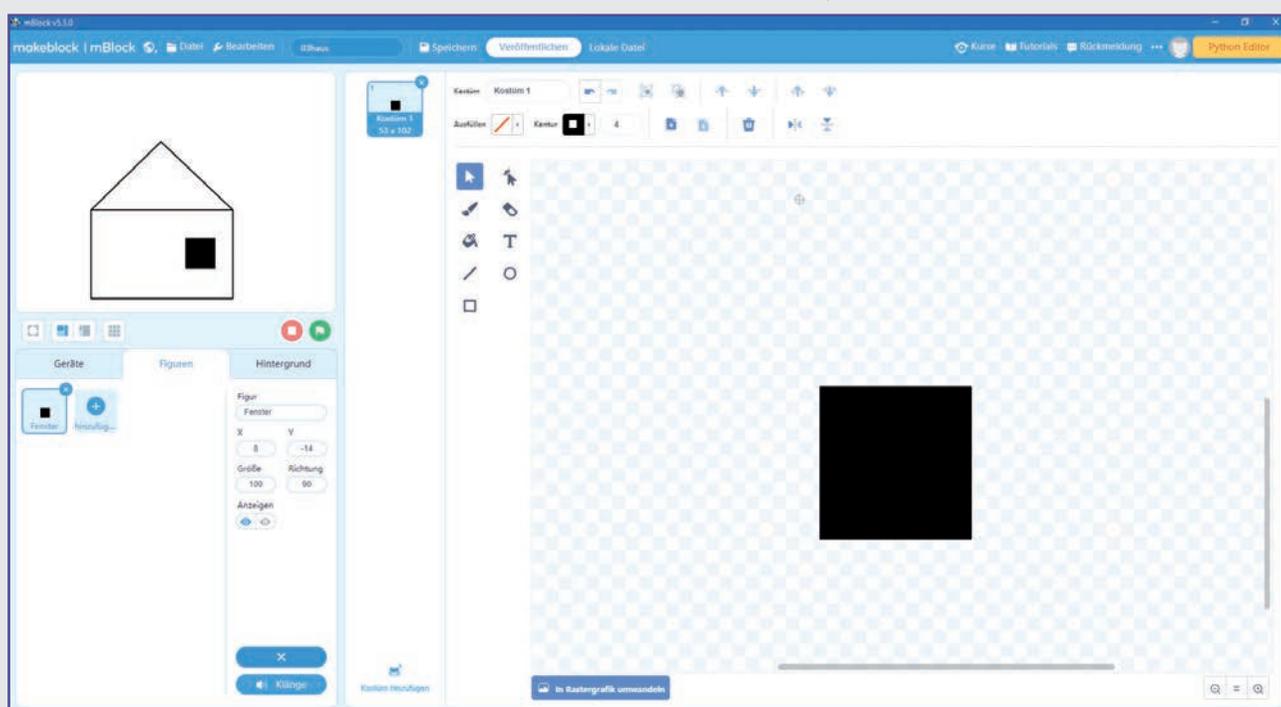
5. Klicke mit der rechten Maustaste auf der Figurenpalette unten links auf die M-Panda-Figur und dann auf das blaue x-Symbol an dieser Figur, um sie zu löschen. Diese Figur wird für das Programm nicht gebraucht.
6. Zeichne stattdessen eine neue Figur für das Fenster. Sie bewegt sich nicht, wird aber als Lichtschalter gebraucht. Klicke dazu auf das Symbol *hinzufügen* in der Figurenpalette.

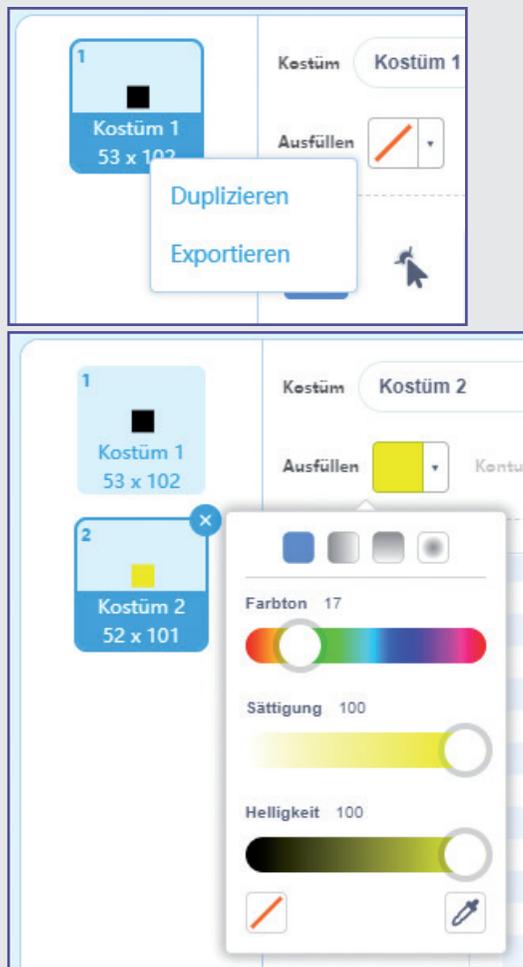


7. Zeichne nach einem Klick auf *Kostüme* ein schwarzes Quadrat. Wenn du beim Anklicken des Rechteckwerkzeugs die [Umschalt]-Taste gedrückt hältst, wird automatisch ein Quadrat gezeichnet. Verschiebe das Quadrat im Grafikbereich so, dass es auf der Bühne an der passenden Stelle auf dem Haus erscheint.

8. Benenne die Figur in der Figurenpalette in *Fenster* um.

9. Jede Figur kann mehrere Kostüme haben und darüber Form, Farbe und Aussehen verändern. Klicke mit der rechten Maustaste auf das markierte *Kostüm 1* und wähle im Kontextmenü *Duplizieren*.





- Klicke jetzt auf das neu duplizierte Kostüm und fülle dort das Quadrat mit dem Farberwerkzeug mit gelber Farbe. Im Feld *Ausfüllen* kannst du eine Farbe einstellen oder irgendwo auf dem Bildschirm mit der Pipette eine auswählen.

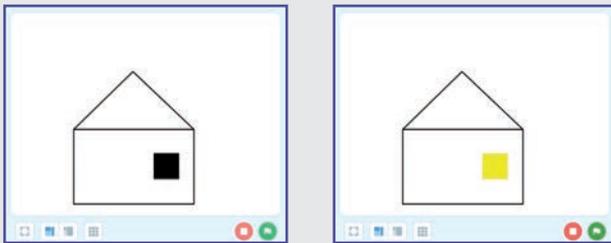
Das schwarze *Kostüm 1* stellt das Fenster mit ausgeschaltetem Licht, das gelbe *Kostüm 2* das Fenster mit eingeschaltetem Licht dar.

KLICK AUF DAS FENSTER SCHALTET DAS LICHT EIN

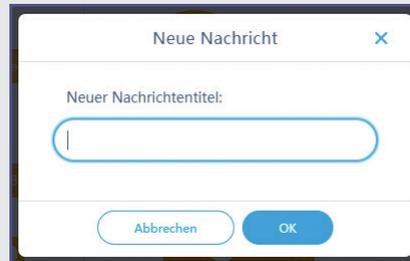
- Nachdem das Haus vorbereitet ist, schließe das Grafikprogramm mit dem x-Symbol, um die Programmblöcke zusammenzubauen. Jeder Figur und auch dem Hintergrund können eigene Programmblöcke zugeordnet werden. Da das Programm ablaufen soll, wenn man auf das Fenster klickt, müssen die Programmblöcke auch dieser Figur zugeordnet sein. Achte darauf, dass die Figur *Fenster* auf der Figurenpalette mit dem blauen Rand markiert ist, während du im Programmbereich die Programmblöcke zusammenbaust.
- Das Programm besteht aus zwei Blöcken: Wenn du auf das grüne Fähnchen klickst,  wird die Starteinstellung aufgebaut. Der Block *wechsle Objekt zu Kostüm 1* aus der Blockpalette *Aussehen* schaltet das Fenster im gezeichneten Haus auf Dunkel, egal wie es vorher aussah. Wähle dazu im Listenfeld dieses Blocks das *Kostüm 1* aus.
- Ziehe noch einen Block *Wenn ich angeklickt werde* von der Blockpalette *Ereignisse* in den Programmbereich. Dieser Block ist wie der Block *wenn Fahne geklickt wird* oben rund, kann also nicht an einen anderen Block angehängt werden, sondern läuft völlig unabhängig. Alle Blöcke, die darunterhängen, werden ausgeführt, wenn man auf die aktuelle Figur, in diesem Fall also auf das Fenster, klickt.
- Das Fenster soll im Haus für fünf Sekunden gelb leuchten und danach wieder schwarz werden. Dazu werden erneut zwei Blöcke *wechsle Objekt zu Kostüm ...* verwendet.



5. Klicke auf das Fenster im Haus auf der Bühne. Es erscheint in Gelb und wechselt nach fünf Sekunden wieder auf Schwarz.



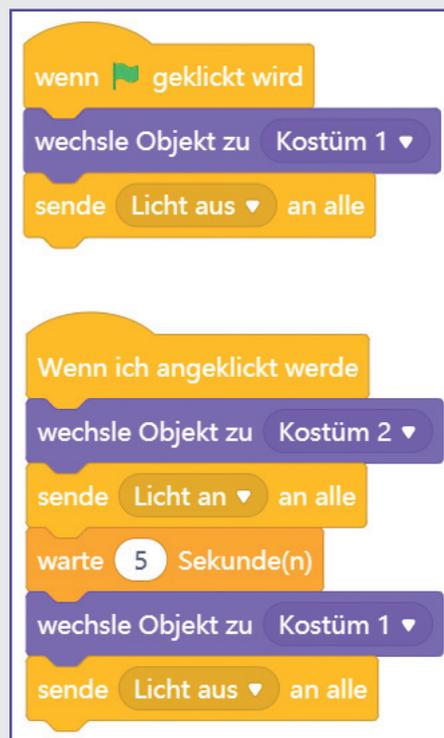
6. Zusätzlich soll die LED ein- und ausgeschaltet werden. Dazu muss das Programm der Figur *Fenster* mit weiteren Programmblöcken des Geräts *Arduino Nano (old)* „sprechen“. mBlock verwendet zu diesem Zweck sogenannte Nachrichten. Um ganz am Anfang die LED auszuschalten, falls sie eingeschaltet ist, ziehe einen Block *sende ... an alle* aus der Blockpalette *Ereignisse* in das Programm. Dieser Block sendet eine Nachricht an alle Figuren und Geräte. Diese können daraufhin mit ihren Aktionen beginnen.
7. Klicke auf das Listenfeld im Block und dann auf *Neue Nachricht*. Schreibe im nächsten Fenster die Nachricht *Licht aus*. Schreibe in gleicher Weise auch eine Nachricht *Licht an*.



8. Die Nachrichten werden im Programm gespeichert und können in jedem verwendeten Block dieses Typs im Listenfeld ausgewählt werden.



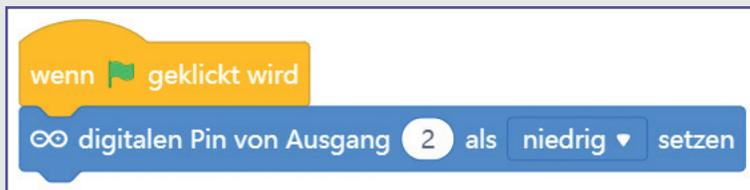
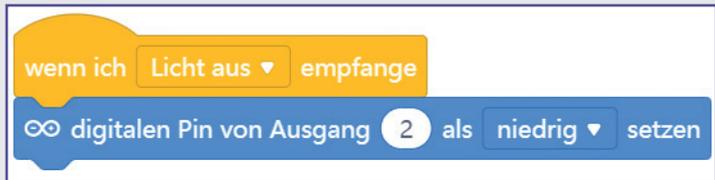
9. Baue auch in den Programmteil, der bei Klick auf das Fenster gestartet wird, Blöcke ein, die die Nachrichten *Licht an* und *Licht aus* senden.



10. Jetzt muss das Gerät noch auf die Nachrichten der Figur reagieren. Schalte dazu unten links auf die Registerkarte *Geräte*. Auch hier soll bei Klick auf das grüne Fähnchen die LED erst einmal ausge-

schaltet werden für den Fall, dass sie aus irgendeinem Grund eingeschaltet ist.

11. Wir verwenden dazu den bereits bekannten Block *digitalen Pin von Ausgang ... als niedrig setzen* von der Blockpalette *Pin*. Trage in das Zahlenfeld eine 2 ein und setze das Listenfeld am Ende auf *niedrig*.



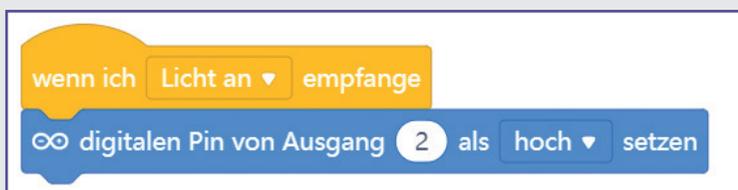
im Listenfeld des Blocks *digitalen Pin von Ausgang ... als ... setzen* die Option *niedrig*, um die LED wieder auszuschalten.

12. Jetzt braucht das Gerät noch Programmblöcke, die die Nachrichten auswerten und die LED ein- oder ausschalten. Ziehe dazu einen Block *wenn ich ... empfange* aus der Blockpalette *Ereignisse* in das Programmfenster des Geräts. Wähle im Listenfeld die Nachricht *Licht an* aus.

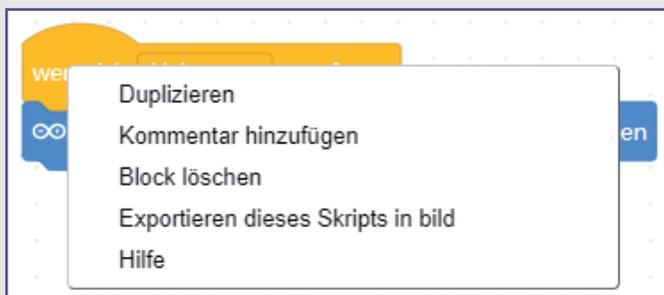
16. Starte das Programm mit einem Klick auf das grüne Fähnchen. Klickst du jetzt auf das Fenster, schaltet es sich auf Gelb, und die LED im Haus leuchtet. Nach fünf Sekunden geht beides wieder aus.

13. Hänge einen Block *digitalen Pin von Ausgang ... als niedrig setzen* von der Blockpalette *Pin* darunter. Trage in das Zahlenfeld eine 2 ein und setze das Listenfeld am Ende auf *hoch*.

Das Programm ist unter dem Namen 03haus.mblock bei den Downloads zu finden.



14. Die Blöcke zum Ausschalten der LED sind sehr ähnlich. Deshalb kannst du sie einfach duplizieren und brauchst sie nicht erneut aus den Blockpaletten zu ziehen. Klicke mit der rechten Maustaste auf den Block *wenn ich Licht an empfange* und wähle im Kontextmenü *Duplizieren*. Jetzt hängt eine Kopie der beiden Blöcke am Mauszeiger. Ziehe sie in einen freien Bereich des Programmfensters.



Tag 4

HEUTE IM ADVENTSKALENDER

- 1 Verbindungskabelpaar

VERBINDUNGSKABEL

Heute ist ein Verbindungskabelpaar im Adventskalender. Diese Leitungen werden verwendet, um die LEDs, die auf dem Hausdach stecken, mit dem Steckbrett zu verbinden. Die Verbindungskabel haben am einen Ende kleine Drahtstecker, mit denen sie sich leicht in das Steckbrett stecken lassen. Am anderen Ende befinden sich Klemmen, in die die Drähte einer LED gesteckt werden können.

Stecke eine LED von außen durch den Spalt am First zwischen den beiden Dachflächen. Schließe dann im Haus die Klemme an und stecke die anderen Kabelenden in das Steckbrett.

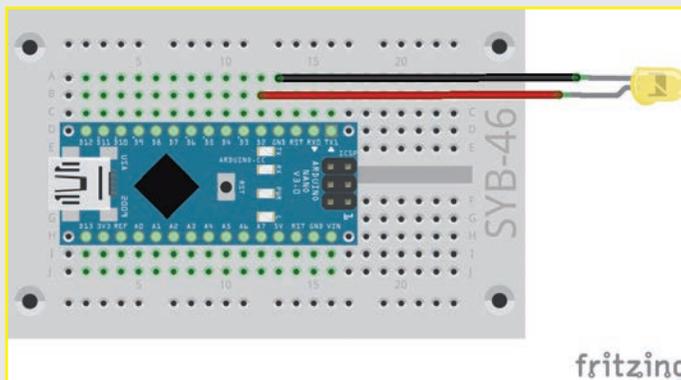
Die LED ist über die Verbindungskabel mit dem Steckbrett verbunden. Achte beim Aufbau der Schaltung darauf, dass die Kathode (kurzer Draht) der LED mit dem GND-Pin des Nano-Boards verbunden ist, die Anode (langer Draht) ist in dieser Schaltung mit Pin D2 auf dem Nano verbunden.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 LED gelb mit Vorwiderstand
- 1 Verbindungskabelpaar

AUFGABE

Eine LED auf dem Hausdach soll mit einstellbarer Geschwindigkeit blinken.



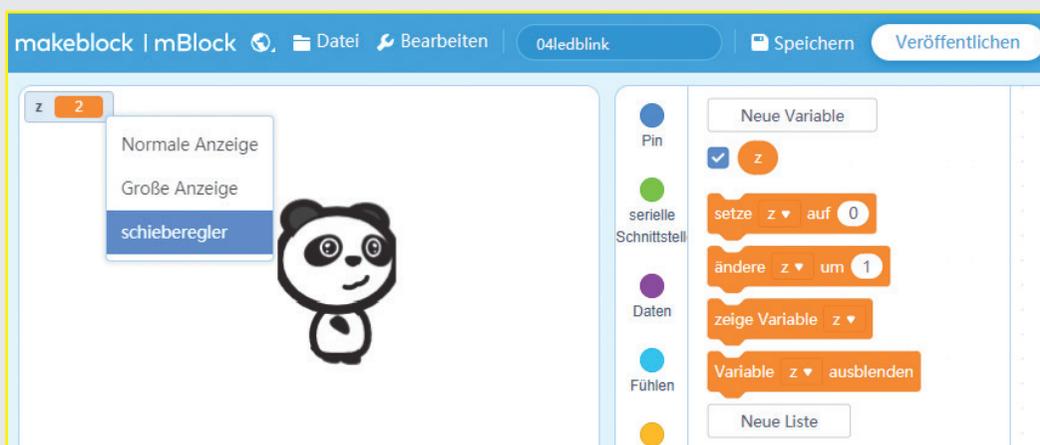
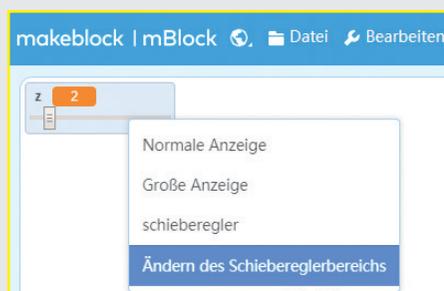
Die Fritzing-Zeichnung zeigt dir den Schaltungs-Aufbau ohne das Haus.

LED BLINKT MIT EINSTELLBARER GESCHWINDIGKEIT

Das Programm des 4. Tages lässt wieder eine LED blinken. Allerdings kann die Geschwindigkeit mit einer Variablen im Programm eingestellt werden.

1. Lege zuerst auf der Blockpalette *Variablen* mit dem Button *Neue Variable* eine Variable *z* an, die die Wartezeit beim Blinken festlegt.
2. Solange das Häkchen links neben dem Variablen-symbol auf der Blockpalette *Variablen* eingeschaltet ist, erscheint die Variable auf der Bühne und zeigt dort immer den aktuellen Wert. Klicke mit der rechten Maustaste auf die Variable auf der Bühne und wähle im Kontextmenü *Schieberegl*. Es erscheint ein Schieberegler, mit dem der Variablenwert, während das Programm läuft, eingestellt werden kann.

3. Klicke noch einmal mit der rechten Maustaste auf die Variable auf der Bühne und wähle diesmal im Kontextmenü *Ändern des Schiebereglerbereichs*. Trage die Werte *1* und *10* ein.



Variablen in mBlock 5

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 Variablen-speicher automatisch wieder geleert. Variablen müssen in mBlock 5 erst einmal auf der Blockpalette *Variablen* mit dem Button *Neue Variable* angelegt werden, bevor man sie nutzen kann. Anschließend kannst du 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 Variablen zur Verfügung.

4. mBlock 5 rechnet zwar mit Fließkommazahlen, die Schieberegler funktionieren aber nur mit ganzzahligen Werten. Deshalb wird der eingestellte Wert anschließend mit dem Divisionsblock ... / ... aus der Blockpalette *Operatoren* durch 10 geteilt. Je nach eingestelltem Wert soll die LED zwischen 0,1 Sekunden und 1 Sekunde leuchten und danach genauso lange ausgeschaltet bleiben.
5. Baue jetzt das Programm wie in der Abbildung zusammen.

Nach dem Anlegen erscheint eine Variable als orange-farbenes Symbol auf der Blockpalette *Variablen*. Von dort kannst du sie überall an die Stelle in das Programm ziehen, an der sonst eine Zahl eingetragen werden müsste.



SO FUNKTIONIERT DAS PROGRAMM



Nach Klick auf das grüne Fähnchen startet eine Endlosschleife. Die *wiederhole fortlaufend*-Schleife lässt die LED endlos blinken. Anstelle einer vom Programm fest vorgegebenen Zeit zwischen dem Umschalten wird die Variable *z* verwendet. Diese kannst du, während das Programm läuft, jederzeit mit

dem Schieberegler verändern. Die LED blinkt entsprechend schneller oder langsamer.



Das Programm ist unter dem Namen `041edblink.mblock` bei den Downloads zu finden.

Tag 5

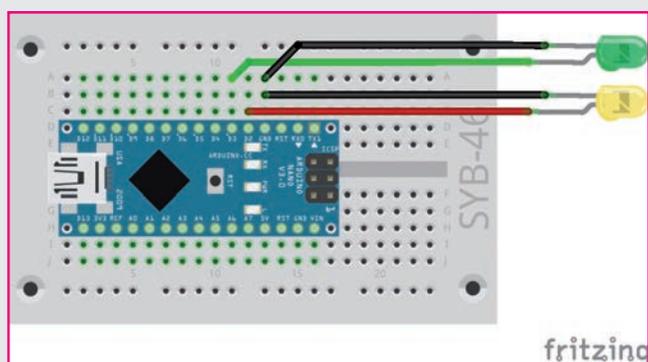
HEUTE IM ADVENTSKALENDER

- 1 LED grün mit Vorwiderstand
- 1 Verbindungskabelpaar

AUFGABE

Zwei LEDs auf dem Hausdach sollen abwechselnd mit einstellbarer Geschwindigkeit blinken.

WECHSELBLINKLICHT



Die LEDs sind an den Pins D2 und D3 angeschlossen. Achte auch diesmal darauf, dass die beiden Kathoden (kurzer Draht) der LEDs mit dem GND-Pin des Nano-Boards verbunden sind.

Das Programm ähnelt bis auf die zusätzlichen Blöcke für die zweite LED dem Programm des 4. Tages. Diese neuen Blöcke funktionieren genau entgegengesetzt. Wird die eine LED eingeschaltet, wird die andere ausgeschaltet, damit beide LEDs abwechselnd blinken.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 LED gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 2 Verbindungskabelpaare

wenn geklickt wird

wiederhole fortlaufend

digitalen Pin von Ausgang 2 als hoch setzen

digitalen Pin von Ausgang 3 als niedrig setzen

warte z / 10 Sekunde(n)

digitalen Pin von Ausgang 2 als niedrig setzen

digitalen Pin von Ausgang 3 als hoch setzen

warte z / 10 Sekunde(n)

SO FUNKTIONIERT DAS PROGRAMM

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

Nachdem die LED an Pin D2 eingeschaltet wurde, wird die LED an Pin D3 ausgeschaltet. Jetzt wartet das Programm die in der Variablen z eingestellte Zeit.

Danach wird auf die gleiche Weise die LED an Pin D3 ein- und die an Pin D2 ausgeschaltet. Nach einer weiteren Wartezeit beginnt der Zyklus von vorn.



Das Programm ist unter dem Namen 051ledblink.mblock bei den Downloads zu finden.

Tag 6

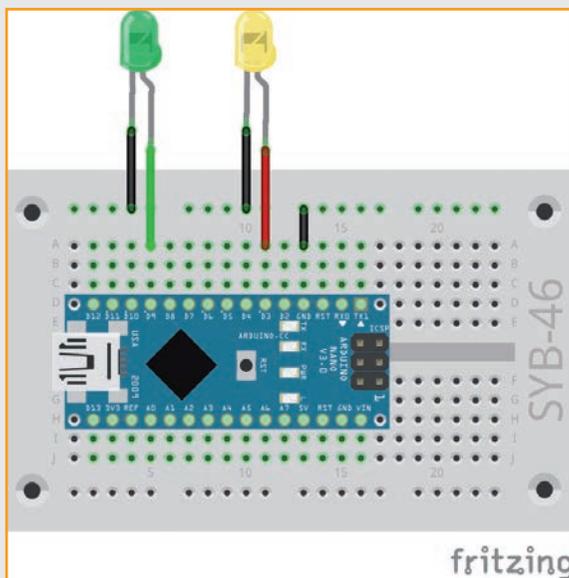
HEUTE IM ADVENTSKALENDER

- Schaltdraht (isoliert)

SCHALTDRAHT

Heute ist Schaltdraht im Adventskalender enthalten. Damit kannst du kurze Verbindungsbrücken herstellen, mit denen dicht nebeneinanderliegende Kontaktreihen auf der Steckplatine verbunden werden. Solche Drahtbrücken liegen flach auf dem Steckbrett und stören beim Aufbau der anderen Bauteile weniger als lange Verbindungskabel.

Schneide 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. Entferne mit einem scharfen Messer an beiden Enden auf einer Länge von etwa einem halben Zentimeter die Isolierung.



Die Schaltung von heute zeigt den typischen Schaltungsaufbau auf dem Steckbrett. Eine der langen Kontaktleisten wird als Masseleitung verwendet, die mit dem GND-Pin auf dem Nano über eine Drahtbrücke verbunden ist. Achte diesmal beim Aufbau der Schaltung darauf, dass die Kathoden (kurzer Draht) der LEDs

Bauteile

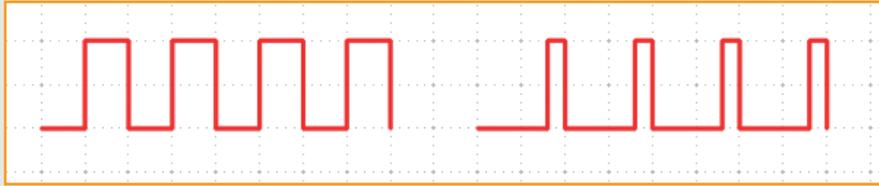
- 1 Nano-Board
- 1 Steckbrett
- 1 LED gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 2 Verbindungskabelpaare
- 1 Drahtbrücke

in der Masseleiste stecken, die Anoden (langer Draht) sind in dieser Schaltung mit den Pins D9 und D3 auf dem Nano verbunden.

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*, Licht an oder aus. Das Gleiche gilt für die als Ausgänge definierten digitalen Pins. Demnach könnte man eine LED theoretisch nicht dimmen.

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* (+5 V) wird verändert. Das Tastverhältnis gibt das Verhältnis der Länge des eingeschalteten Zustands zur Gesamtdauer eines Schaltzyklus an.



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

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

Pins für PWM-Signale

Die Pins D3, D5, D6, D9, D10 und D11 sind auf den Schaltbildern mit einem ~-Symbol gekennzeichnet. Diese Pins können für PWM verwendet werden.

AUFGABE

Zwei LEDs sollen gleichmäßig heller und dunkler werden.

Der gleichmäßige Zyklus zeigt deutlich, dass ein Mensch die Helligkeit von Licht bei verschiedenen Farben unterschiedlich wahrnimmt.

DAS PROGRAMM

Das Programm des 6. Tages dimmt die LEDs an den Pins D3 und D9 zyklisch heller und dunkler. Dabei werden einige neue Programmblöcke benötigt.

```

wenn geklickt wird
  setze schritt auf 10
  setze hell auf 0
  wiederhole fortlaufend
    PWM-Ausgang 3 als hell festlegen
    PWM-Ausgang 9 als hell festlegen
    ändere hell um schritt
  wenn hell = 0 oder hell = 250, dann
    setze schritt auf 0 - schritt
  warte zeit / 10 Sekunde(n)
  
```

SO FUNKTIONIERT DAS PROGRAMM

Das Programm verwendet drei Variablen:

- *zeit* legt die Geschwindigkeit beim Dimmen fest.
- *hell* bezeichnet den PWM-Wert für die Helligkeit der LEDs.
- *schritt* gibt die Schrittweite beim Dimmen an.

Nachdem die Variablen auf ihre Grundeinstellung gesetzt wurden, beginnt eine Endlosschleife. Als Erstes wird bei jedem Schleifendurchlauf der aktuelle Wert der Variablen *hell* als PWM-Wert auf den Pins D3 und D9 ausgegeben. Auf der Blockpalette *Pin* gibt es einen eigenen Block für PWM-Pins. Anschließend wird der Wert der Variablen *hell* um den Wert *schritt* erhöht.

Im nächsten Schritt wird mit einem *wenn ... dann ...*-Block von der Blockpalette *Steuerung* überprüft, ob der Wert von *hell* die Grenzen 0 oder 250 erreicht hat. In diesem Fall wird ein *oder*-Block aus der Blockpalette *Operatoren* 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 *wenn*-Blocks wird ausgeführt. Blöcke mit Abfragen sind durch spitze Enden

gekennzeichnet. Sie können ineinander verschachtelt und in Steuerungsblöcke eingebaut werden, die Felder in dieser Form enthalten.

Zwei Gleichheitsabfragen prüfen, ob der Wert der Variablen *hell* den Wert 0 oder 250 erreicht hat. Trifft eines davon zu, wird die Variable *schritt* auf ihren negativen Wert gesetzt, um die Richtung umzukehren. Da mBlock 5 keine Möglichkeit bietet, das Vorzeichen einer Variablen umzukehren, verwenden wir den Minusoperator '-' und subtrahieren den Wert der Variablen von 0, was das gleiche Ergebnis liefert.

Zum Schluss wartet das Programm in jedem Schleifendurchlauf die in der Variablen *zeit* gespeicherten Zehntelsekunden. Der Wert der Variablen wird mit einem *... / ...*-Block durch 10 geteilt. Anschließend startet die Endlosschleife neu und liefert der LED einen neuen PWM-Wert. Die Variable *zeit* kann, während das Programm läuft, über einen Schieberegler verändert werden, um die LEDs schneller oder langsamer hell und dunkel werden zu lassen.



Das Programm ist unter dem Namen `061edpwm.mblock` bei den Downloads zu finden.

Tag 7

HEUTE IM ADVENTSKALENDER

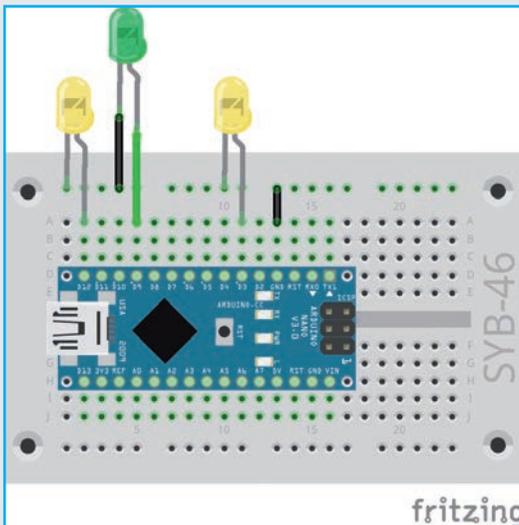
- 1 LED gelb mit Vorwiderstand

AUFGABE

Drei LEDs werden über ein Steuerpult auf dem PC gesteuert. Das Licht im Haus (gelbe LED im Fenster) soll für eine einstellbare Zeit leuchten. In einem zweiten Fenster flackert ein Kamin (zweite gelbe LED). Auf dem Dach blinkt eine grüne LED mit einstellbarer Geschwindigkeit.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 2 LEDs gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 1 Verbindungskabelpaar
- 1 Drahtbrücke



DAS PROGRAMM

Das Programm zeigt auf der Bühne ein Haus als Steuerpult für die Beleuchtung. Es enthält drei Figuren: die beiden Fenster sowie einen Kreis für die LED auf dem Dach. Jede Figur hat zwei Kostüme, die den ausgeschalteten und den eingeschalteten Zustand der LED kennzeichnen. Bei jeder Figur werden auf der Bühne die zugehörigen Variablen angezeigt. Überall dort, wo es sinnvoll ist, haben diese Variablen Schieberegler, um zum Beispiel Blinkzeiten einzustellen.



Bei Klick auf das linke Fenster leuchtet die LED an Pin D12 für eine einstellbare Zeit. Beim Klick auf das rechte Fenster flackert eine LED an Pin D3 mithilfe eines PWM-Signals. Die Flackergeschwindigkeit lässt

sich einstellen. Bei Anklicken des Kreises auf dem Dach blinkt die LED an Pin D9. Dabei lassen sich die Dauer und die Blinkzeit einstellen.

SO FUNKTIONIERT DAS PROGRAMM



Jede der drei Figuren hat ihre eigenen Programmblöcke. Ein Block legt bei Klick auf das grüne Fähnchen Grundeinstellungen fest. Dabei wird die entsprechende LED ausgeschaltet und das erste Kostüm gewählt. Ein weiterer Block startet den Lichteffekt bei Anklicken der Figur. Zum Schalten der LEDs werden wieder Nachrichten verwendet, die von der jeweiligen Figur an das Gerät gesendet werden. Dort werden die Nachrichten ausgewertet, um die LEDs ein- oder auszuschalten.

ERSTES FENSTER

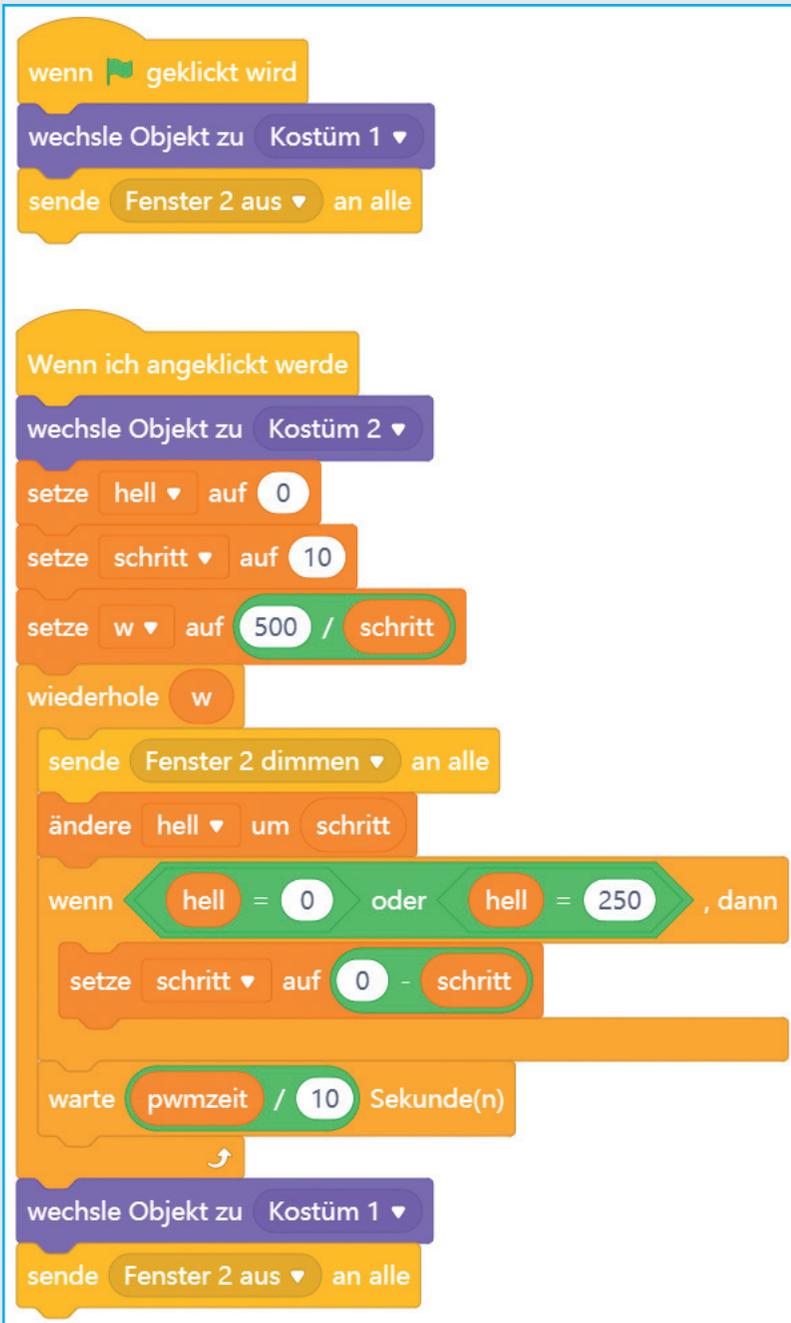
Bei Klick auf das grüne Fähnchen wird das Fenster schwarz angezeigt und die gelbe LED über die Nachricht *Fenster 1 aus* ausgeschaltet. Das Gerät wertet diese Nachricht aus und schaltet die LED an Pin D12 ein.

Bei Klick auf das Fenster wechselt es das Kostüm und erscheint in Gelb. Die LED wird eingeschaltet. Anschließend wartet das Programm die eingestellte Zeit, bis das Fenster wieder schwarz ist und die LED ausgeschaltet wird. Die Variable *zeit1* gibt an, wie lange die LED im Fenster nach dem Anklicken eingeschaltet bleibt. Die Zeit kann jederzeit über den Schieberegler *zeit1* verändert werden. Es werden ganze Sekunden verwendet.

ZWEITES FENSTER

Bei Klick auf das zweite Fenster wird eine Schleife gestartet, die ein PWM-Signal für die LED an Pin D3 zyklisch verändert. Der Programmaufbau entspricht weitgehend dem Programmaufbau des 6. Tages.





Die Nachricht *Fenster 2 dimmen* startet auf dem Gerät einen Block *PWM-Ausgang 3 als hell festlegen*, wobei die Variable *hell* direkt aus dem Programm der Figur *Fenster2* übernommen wird. Bei Anlegen der Variablen muss der Schalter *für alle Figuren* aktiviert sein, damit diese Variable von allen Figuren und Geräten genutzt werden kann.

Die Variable *pwmzeit*, die die Wartezeit zwischen zwei Veränderungen des PWM-Signals und damit die Flackergeschwindigkeit festlegt, kann über einen Schieberegler bestimmt werden, während das Programm läuft. Diese Variable wird in Zehntelsekunden eingestellt und im Programm durch 10 geteilt.

LED AUF DEM DACH

Ein Klick auf das Symbol der LED auf dem Dach lässt die LED an Pin D9 blinken. Dabei lassen sich die Gesamtzeit *zeit2* und auch die Zeit, die festlegt, wie lange die LED bei jedem Blinkvorgang eingeschaltet ist – die *blinkzeit* –, einstellen.



Um die Anzahl der Wiederholungen zu berechnen, wird zunächst die Gesamtzeit durch die Blinkzeit geteilt. Die Blinkzeit, die in Zehntelsekunden eingestellt wird, wird dabei vorher durch 10 geteilt. Das Ergebnis wird noch einmal durch 2 geteilt, da jeder Schleifendurchlauf zwei Blinkzeiten dauert. Während der ersten Blinkzeit ist die LED an Pin D9 eingeschaltet, während der zweiten ist sie aus.



Das Programm ist unter dem Namen *07haus.mblock* bei den Downloads zu finden.

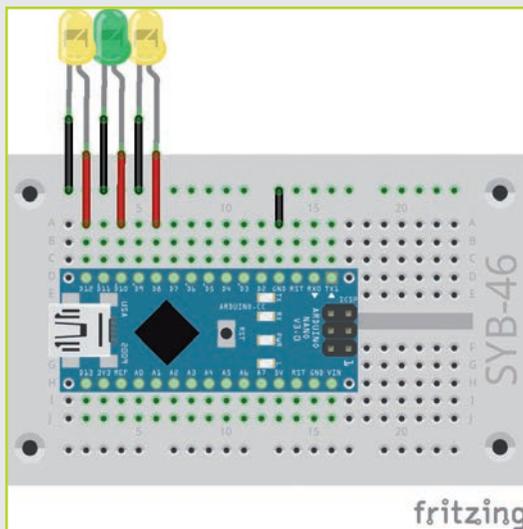
Tag 8

HEUTE IM ADVENTSKALENDER

- 1 Verbindungskabelpaar

AUFGABE

Lauflichter sind immer wieder beliebte Effekte, nicht nur für Werbung und Partyräume. Drei LEDs sollen als Lauflicht nacheinander aufleuchten. Über eine Variable steuert man die Geschwindigkeit des Lauflichts.



DAS PROGRAMM

Das Programm lässt die drei LEDs an den Pins D8, D10 und D12 zyklisch als Lauflicht aufleuchten. Die Pinnummern haben jeweils einen Abstand von 2 voneinander. So können sie einfach in einer Schleife hochgezählt werden.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 2 LEDs gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 3 Verbindungskabelpaare
- 1 Drahtbrücke

```
wenn  geklickt wird
  digitalen Pin von Ausgang 8 als niedrig setzen
  digitalen Pin von Ausgang 10 als niedrig setzen
  digitalen Pin von Ausgang 12 als niedrig setzen
  wiederhole fortlaufend
    setze n auf 8
    wiederhole 3
      digitalen Pin von Ausgang n als hoch setzen
      warte zeit / 10 Sekunde(n)
      digitalen Pin von Ausgang n als niedrig setzen
      ändere n um 2
```

SO FUNKTIONIERT DAS PROGRAMM

Bei der Verwendung von Variablen kann mBlock 5 nicht gleich am Anfang erkennen, welche Pins als Ausgänge verwendet werden sollen, und initialisiert diese erst bei ihrer ersten Verwendung. Daher kann man von keinem eindeutig definierten Zustand der LEDs zum Programmstart ausgehen. Um dies zu vermeiden, schaltet das Programm am Anfang alle LEDs mit den Blöcken *digitalen Pin von Ausgang ... als niedrig setzen* aus und initialisiert sie damit.

Danach startet die Endlosschleife, die in jedem Durchlauf die Variable n auf 8 setzt, die Nummer der ersten LED. In jedem Durchlauf der Endlosschleife läuft das Lauflicht über alle drei LEDs.

Eine innere Schleife lässt die drei LEDs nacheinander kurz aufleuchten. Nach jeder LED wird die Variable n

um 2 erhöht, und damit wird die nächste LED angesteuert. Nachdem alle LEDs einmal geleuchtet haben, startet der nächste Durchlauf der Endlosschleife mit der LED an Pin D8.

Die Leuchtzeit der einzelnen LEDs wird über die Variable *zeit* interaktiv eingestellt. Diese Variable kann über einen Schieberegler von 1 bis 10 Zehntelsekunden eingestellt werden. Der Wert legt die Zeit fest, die jede einzelne LED leuchtet. Im Programm wird die Variable *zeit* jedes Mal durch 10 geteilt.



Das Programm ist unter dem Namen `08lauflicht.mblock` bei den Downloads zu finden.

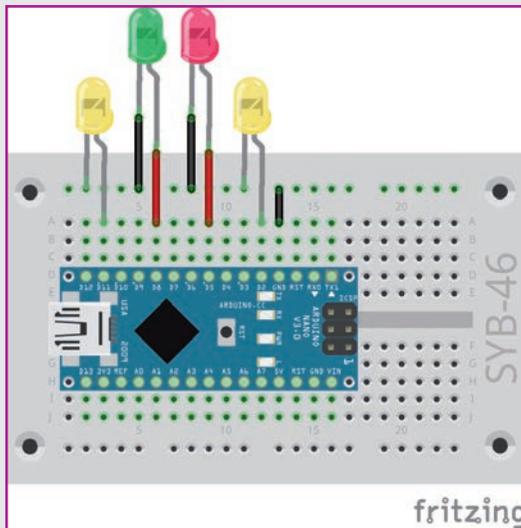
Tag 9

HEUTE IM ADVENTSKALENDER

- 1 LED rot mit Vorwiderstand

AUFGABE

Die vier LEDs (zwei auf dem Dach, zwei im Haus) sollen in zufälliger Reihenfolge blinken.



Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 2 LEDs gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 1 LED rot mit Vorwiderstand
- 2 Verbindungskabelpaare
- 1 Drahtbrücke

Die LEDs haben diesmal einen Abstand von je drei Pins, damit die beiden gelben besser hinter die Fenster des Hauses passen.

DAS PROGRAMM

Das Programm lässt vier LEDs zufällig abwechselnd blinken.

wenn geklickt wird

digitalen Pin von Ausgang **2** als **niedrig** setzen

digitalen Pin von Ausgang **5** als **niedrig** setzen

digitalen Pin von Ausgang **8** als **niedrig** setzen

digitalen Pin von Ausgang **11** als **niedrig** setzen

wiederhole fortlaufend

setze **n** auf wähle eine zufällige Zahl zwischen **0** und $3 * 3 + 2$

digitalen Pin von Ausgang **n** als **hoch** setzen

warte **zeit / 10** Sekunde(n)

digitalen Pin von Ausgang **n** als **niedrig** setzen

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 und sich auch ohne jede Regelmäßigkeit ändern, 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.

SO FUNKTIONIERT DAS PROGRAMM

Am Anfang werden die vier Pins der LEDs ausgeschaltet und dabei gleichzeitig initialisiert. Danach startet eine Endlosschleife.

Im ersten Schritt jedes Durchlaufs der Endlosschleife wird eine Zufallszahl benötigt, die einen der vier verwendeten Pins D2, D5, D8 oder D11 bezeichnet. Da der Block *wähle eine zufällige Zahl zwischen ... und ...* nur Zufallszahlen in einem begrenzten, fortlaufend nummerierten Bereich erzeugen kann, brauchen wir eine Umrechnungsformel: Zunächst wird die Variable *n* auf eine Zufallszahl zwischen 0 und 3 gesetzt. Diese Zahl wird mit 3 multipliziert, und anschließend wird noch 2 hinzuaddiert. Das Ergebnis ist eine der Zahlen 2, 5, 8 oder 11 und gibt die Pinnummer der einzuschaltenden LED an.

Die Geschwindigkeit des Farbwechsels wird wie im letzten Programm über die einstellbare Variable *zeit* gesteuert, die in jedem Schleifendurchlauf einmal abgefragt und durch 10 geteilt wird, um Zeiten in Zehntelsekunden einstellen zu können.

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



Das Programm ist unter dem Namen 091ed4blink.mblock bei den Downloads zu finden.

HEUTE IM ADVENTSKALENDER

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

WIDERSTÄNDE UND IHRE FARBCODES

Widerstände werden unter anderem zur Strombegrenzung an empfindlichen elektronischen Bauteilen sowie als Vorwiderstände für LEDs verwendet. In welcher Richtung ein Widerstand eingebaut wird, ist egal. Bei LEDs dagegen spielt die Einbaurichtung eine wichtige Rolle.

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 die winzig kleinen 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.

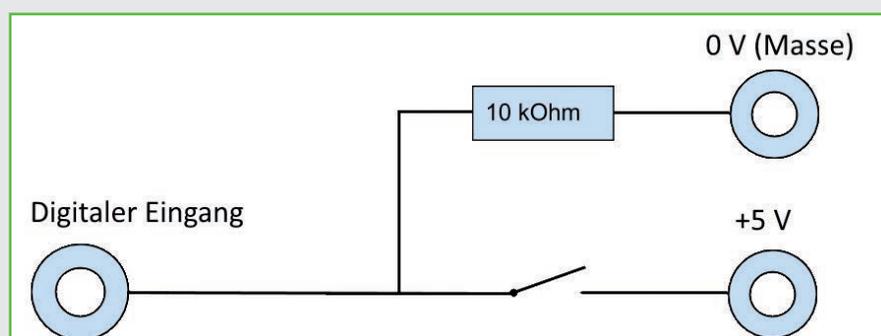
Farbe	Widerstandswert in Ohm			
	1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multiplikator)	4. Ring (Toleranz)
Silber			$10^{-2} = 0,01$	$\pm 10\%$
Gold			$10^{-1} = 0,1$	$\pm 5\%$
Schwarz		0	$10^0 = 1$	
Braun	1	1	$10^1 = 10$	$\pm 1\%$
Rot	2	2	$10^2 = 100$	$\pm 2\%$
Orange	3	3	$10^3 = 1.000$	
Gelb	4	4	$10^4 = 10.000$	
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. Dazu verwenden wir im heutigen Projekt einen Taster, der als Türklingelknopf in die Giebelwand neben der Tür gesteckt und wie die LEDs von innen mit einem Verbindungskabelpaar am Steckbrett angeschlossen wird. Solange der Taster gedrückt ist, sind beide 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, könnte 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.

AUFGABE

Bei Drücken des Tasters soll das Licht einige Sekunden lang leuchten.

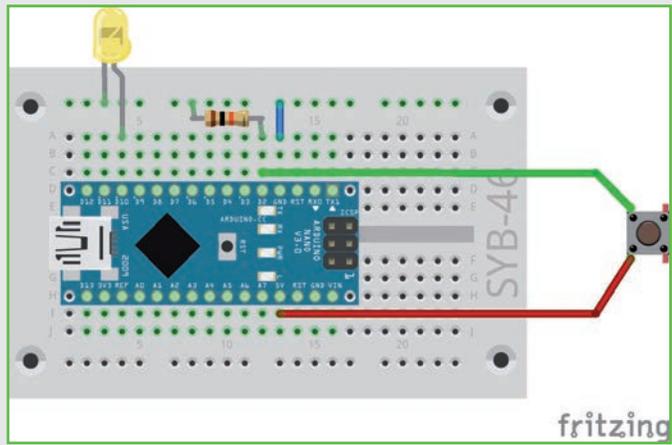
Der Taster wird zur Türklingel am Haus. Schließe den Taster über zwei Verbindungskabel im Haus am Steckbrett an. Das funktioniert genau wie bei den LEDs. Lege den Taster nun vorsichtig in die Aussparung an der Tür. Achtung, dafür braucht es etwas Fingerspitzengefühl. Du kannst den Taster und die Kabel auch in einen der Bäume im Garten pieksen. Die nicht gebackenen Salzteig-Teile sollten dafür noch weich genug sein. Oder lass den Taster einfach lose aus der Tür hängen. Das funktioniert genauso!

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 LED gelb mit Vorwiderstand
- 1 Taster
- 1 Widerstand 10 kOhm (braun, schwarz, orange)
- 1 Verbindungskabelpaar
- 1 Drahtbrücke



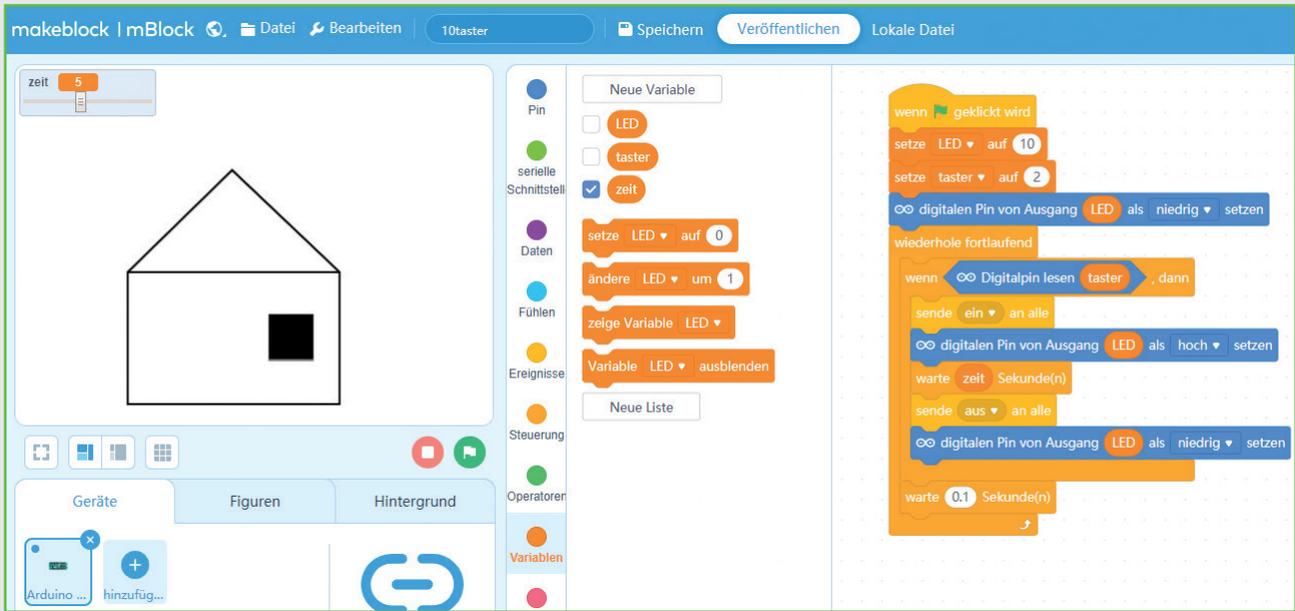
Die Kabel für den Taster lassen sich prima durch einen Lebkuchen-Baum stecken.



DAS PROGRAMM

Bei Drücken des Tasters am digitalen Pin D2 leuchtet die LED die eingestellte Zeit. Das Fenster verändert dabei auch sein Kostüm. Bei Anklicken des Fensters passiert in diesem Programm nichts. Das Hauptprogramm ist diesmal bei dem Gerät gespeichert, da

es ständig läuft und auf den Taster wartet. Es sendet Nachrichten an die Figur, die nichts anderes tut, als das Kostüm zu wechseln. Die beiden Kostüme haben die Namen *Licht an* und *Licht aus*.





SO FUNKTIONIERT DAS PROGRAMM

Das Drücken eines Tasters – also genauer der Zustand eines digitalen Eingangs – kann nicht als Ereignis abgefragt werden wie zum Beispiel das Anklicken eines Objekts. Es gibt keinen Block in mBlock 5, der bei Drücken eines Tasters automatisch startet. Aus diesem Grund verwenden wir eine Endlosschleife. Sie prüft in jedem Durchlauf, ob der Taster gedrückt wird. Nach einer Zehntelsekunde erfolgt der nächste Durchlauf.

Der Block *wenn ... dann* von der Blockpalette *Steuerung* führt alle Blöcke innerhalb der Klammer aus, wenn die Bedingung wahr ist. Ziehe dazu einen Block *Digitalpin lesen ...* von der Blockpalette *Pin* in das Feld für die Bedingung. Ziehe dann noch die Variable *taster* in

den Block, um den Taster abzufragen, dessen Pin in dieser Variablen gespeichert ist.

Die Blöcke innerhalb der Klammer des *wenn ... dann*-Blocks sind bereits bekannt. Sie senden eine Nachricht an die Figur, die daraufhin das Kostüm des Fensters umschaltet. Danach wird die LED eingeschaltet. Nach der eingestellten Zeit wird beides wieder zurückgesetzt. Die Zeit kann über einen Schieberegler interaktiv in Sekunden eingestellt werden.

Das Programm ist unter dem Namen `10taster.mblock` bei den Downloads zu finden.

Tag 11

HEUTE IM ADVENTSKALENDER

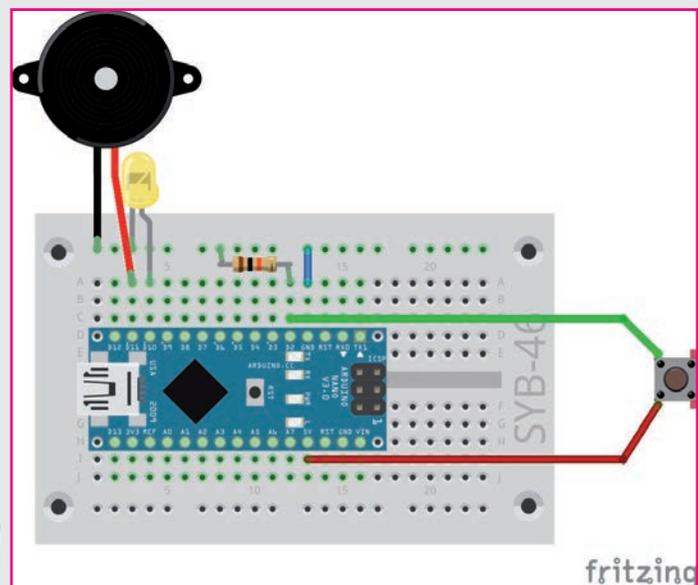
- 1 Piezo-Summer

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.

AUFGABE

Der Taster ist jetzt die Türklingel am Haus. Stecke die beiden Drähte von außen durch die beiden Löcher neben der Haustür und schließe den Taster über ein Verbindungskabelpaar im Haus am Steckbrett an. Drückt jemand auf die Klingel, geht im Haus das Licht an, und die Klingel ertönt.

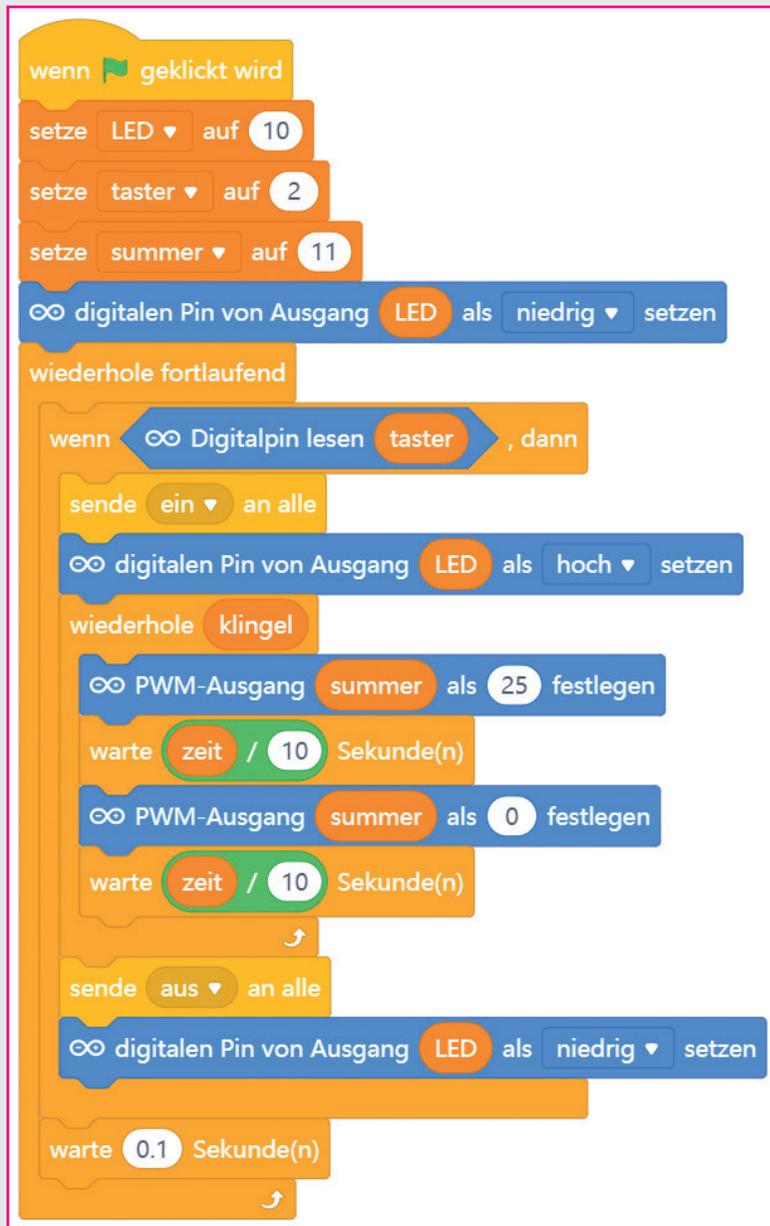


Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 LED gelb mit Vorwiderstand
- 1 Taster
- 1 Widerstand 10 kOhm (braun, schwarz, orange)
- 1 Piezo-Summer
- 1 Verbindungskabelpaar
- 1 Drahtbrücke

DAS PROGRAMM

Wie im Programm von gestern leuchtet auch im heutigen Programm bei Druck auf den Taster die LED eine Zeit lang. Zusätzlich lässt der Piezo-Summer einige Töne erklingen, wobei Tondauer und Wiederholung über Schieberegler einstellbar sind.



SO FUNKTIONIERT DAS PROGRAMM

Das Programm wartet – wie das Programm von gestern – in einer Endlosschleife darauf, dass der Taster gedrückt wird.

Nachdem die LED eingeschaltet ist, startet eine *wiederhole fortlaufend*-Schleife, die auf dem Piezo-Summer Pieptöne ausgibt. Wie viele Töne das sein sollen, können Sie über den Schieberegler der neuen Variablen *klingel* festlegen.

In jedem Schleifendurchlauf wird am in der Variablen *summer* gespeicherten Pin ein PWM-Signal ausgegeben. Die Dauer wird mit dem Schieberegler der Varia-

blen *zeit* in Zehntelsekunden festgelegt und deshalb wieder einmal durch 10 geteilt. Nach dem Ton wird das PWM-Signal ausgeschaltet. Das Programm wartet noch einmal die gleiche Zeit, bevor die Schleife den nächsten Ton abspielt.

Zum Schluss wird die LED ausgeschaltet, und die Hauptschleife wartet darauf, dass der Taster erneut gedrückt wird.



Das Programm ist unter dem Namen `11klingel.mblock` bei den Downloads zu finden.

Tag 12

HEUTE IM ADVENTSKALENDER

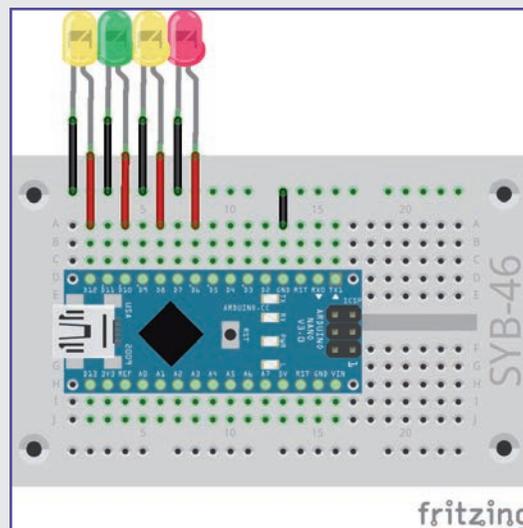
- 1 Verbindungskabelpaar

AUFGABE

Bei Druck auf die Pfeiltaste nach rechts auf der Tastatur sollen vier LEDs auf dem Dach des Hauses nacheinander als Lauflicht leuchten. Die Pfeiltaste nach links lässt das Lauflicht in umgekehrter Richtung laufen, bei Drücken der [Leertaste] blinken alle vier LEDs gleichzeitig. In Wirklichkeit leuchten sie ganz schnell nacheinander auf, das menschliche Auge nimmt dies aber als fast gleichzeitig wahr.

Bauteile

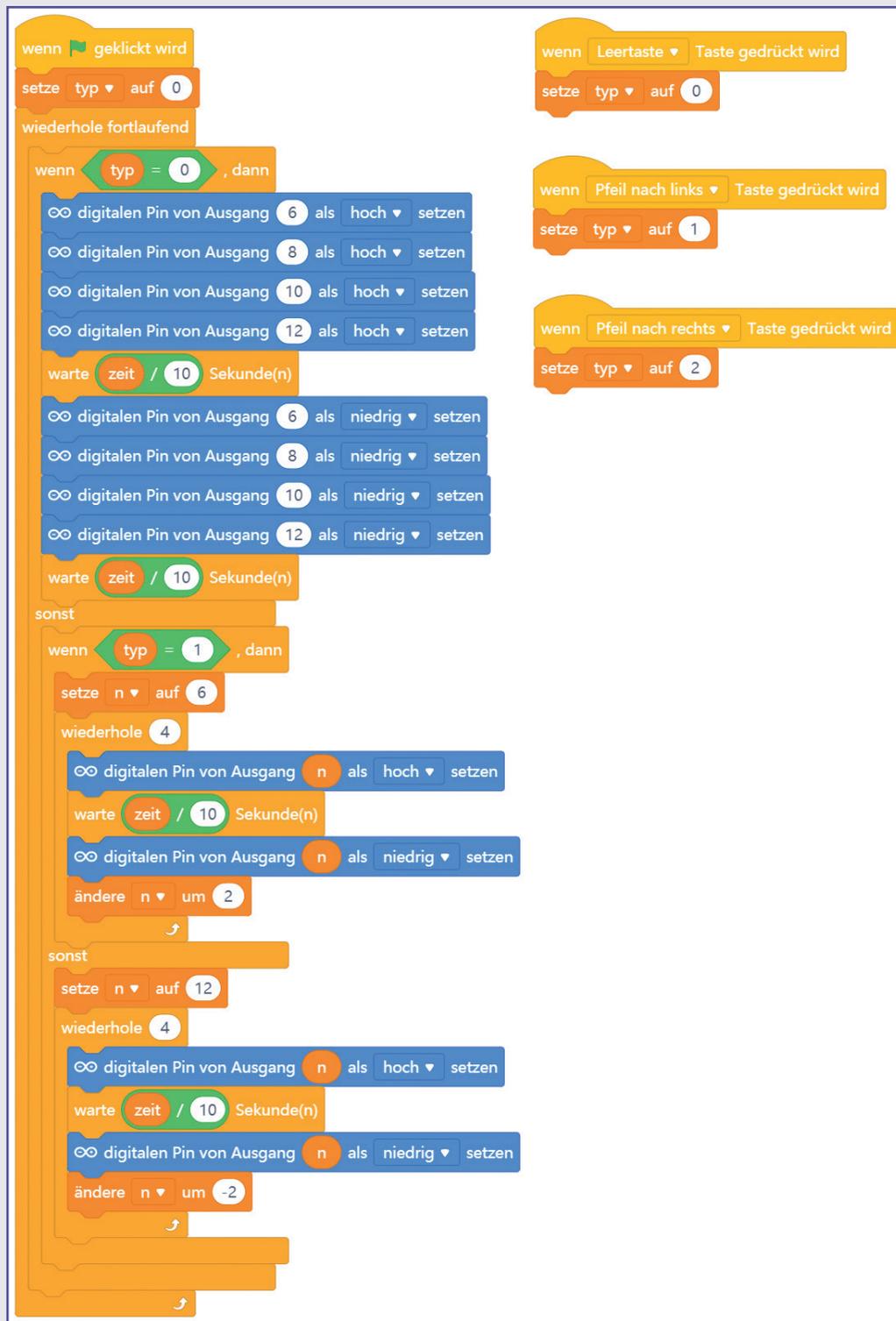
- 1 Nano-Board
- 1 Steckbrett
- 2 LEDs gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 1 LED rot mit Vorwiderstand
- 4 Verbindungskabelpaare
- 1 Drahtbrücke



DAS PROGRAMM



Das Programm besteht aus vier voneinander unabhängigen Programmblöcken. Das Hauptprogramm schaltet bei Klick auf das grüne Fähnchen alle vier LEDs aus. Die anderen drei Blöcke reagieren auf verschiedene Tasten auf der Tastatur.



SO FUNKTIONIERT DAS PROGRAMM

Der Block *wenn Taste ... gedrückt* von der Blockpalette *Ereignisse* startet automatisch, wenn eine bestimmte Taste gedrückt wurde. Du brauchst also nicht wie bei Tastern an digitalen Eingängen mit einer Endlosschleife darauf zu warten, dass die Taste gedrückt wird.

Bei Drücken auf das grüne Fähnchen wird die Variable *typ* auf 0 gesetzt. Dieser Typ legt eines von drei möglichen Blinkmustern fest. Ein Druck auf die `Leertaste`, die Pfeiltaste nach links oder die nach rechts setzt die Variable *typ* auf 0, 1 oder 2.

In einer Endlosschleife wird diese Variable ständig abgefragt, und abhängig davon läuft einer der Lauflichttypen genau einmal ab. Wurde keine andere Taste gedrückt, läuft das gleiche Lauflicht ein weiteres Mal. Hat der Benutzer eine andere der drei Tasten gedrückt, startet ein anderer Lauflichttyp.

ALLE LEDES BLINKEN (TYP = 0)

Nach Drücken der [Leertaste] werden die vier LEDs eingeschaltet. Bedingt durch die langsame Kommunikation mit dem Arduino, sieht man die LEDs nacheinander aufleuchten. Nach Ablauf der in der Variablen *zeit* festgelegten Zeit werden die LEDs wieder ausgeschaltet. Diese Variable kann, während das Programm läuft, über einen Schieberegler interaktiv in Zehntelsekunden eingestellt werden. Nach einer weiteren Wartezeit wiederholt sich die Hauptschleife mit der Abfrage der Variablen *typ*.

LAUFLICHT NACH LINKS (TYP = 1)

Bei Drücken der Pfeiltaste nach links läuft das Lauflicht nach links in aufsteigender Reihenfolge der Pinnummern von der rechten LED an Pin D6 über D8 und D10 bis zur linken LED an Pin D12. Dazu läuft in der Endlosschleife eine weitere Schleife, die viermal wiederholt wird und in jedem Durchlauf die in der Variablen *n* gespeicherte Pinnummer um 2 erhöht, nachdem die betreffende LED kurz ein- und wieder ausgeschaltet wurde. Nach vier Durchläufen der inneren Schleife setzt die Endlosschleife die Pinnummer *n* wieder auf 6. Danach wird die Hauptschleife mit der Abfrage der Variablen *typ* wiederholt.

LAUFLICHT NACH RECHTS (TYP = 2)

Das Lauflicht nach rechts läuft fast identisch. Es startet aber mit der LED ganz links an Pin D12. Dann wird in jedem Durchlauf die in der Variablen *n* gespeicherte Pinnummer bis zur LED an Pin D6 ganz rechts um 2 verringert.



Das Programm ist unter dem Namen `12lauflicht.mblock` bei den Downloads zu finden.

Tag 13

HEUTE IM ADVENTSKALENDER

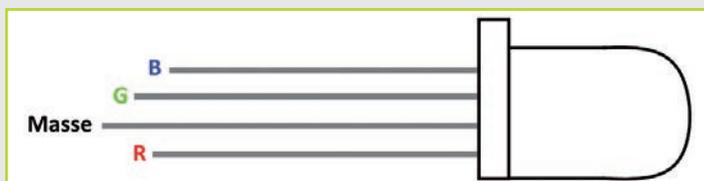
- 1 RGB-LED mit Vorwiderständen

RGB-LEDS

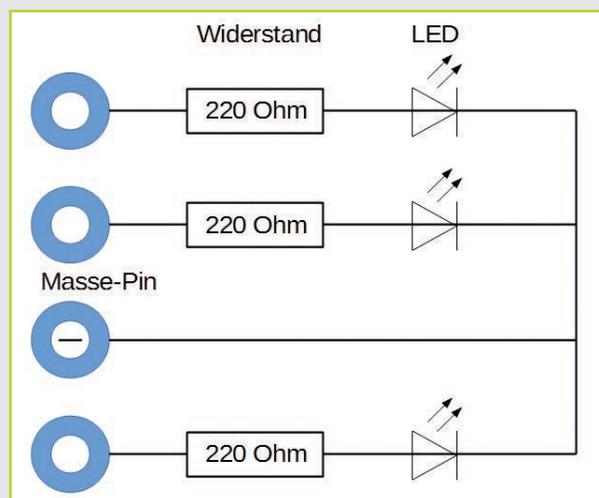
Eine normale LED leuchtet immer nur in einer Farbe. Die im Adventskalender verwendeten RGB-LEDs können wahlweise in mehreren Farben leuchten. Bei ihnen befinden sich im Prinzip drei LEDs in verschiedenen Farben in einem transparenten Gehäuse. Jede dieser drei LEDs hat eine eigene Anode, über die sie mit einem Anschlusspin des Nano-Boards verbunden wird. Die Kathode, die mit der Masseleitung verbunden wird, ist nur einmal vorhanden. Deshalb hat eine RGB-LED vier Anschlussdrähte.

Die Anschlussdrähte der RGB-LEDs sind unterschiedlich lang, um sie eindeutig kenntlich zu machen. Im Gegensatz zu normalen LEDs ist die Kathode hier der längste Draht.

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



Anschlusspins einer RGB-LED.



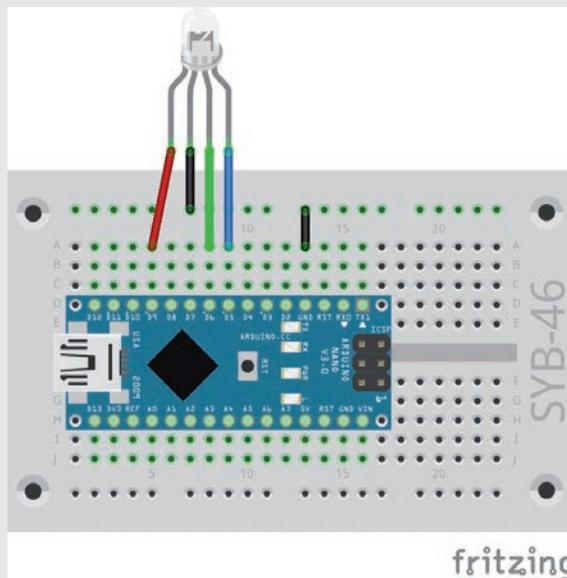
Schaltplan für eine RGB-LED mit drei Vorwiderständen.

AUFGABE

Die RGB-LED soll nacheinander in den Grundfarben und den Mischfarben leuchten.

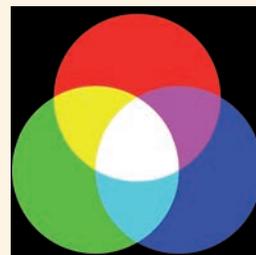
Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 RGB-LED mit Vorwiderständen
- 2 Verbindungskabelpaare
- 1 Drahtbrücke



Additive Farbmischung

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



DAS PROGRAMM

In einer Endlosschleife werden nacheinander verschiedene digitale Pins ein- und ausgeschaltet. In diesem Fall handelt es sich um die drei Farbkomponenten der RGB-LED.

```
wenn geklickt wird
  setze r auf 9
  setze g auf 6
  setze b auf 5
  digitalen Pin von Ausgang r als niedrig setzen
  digitalen Pin von Ausgang g als niedrig setzen
  digitalen Pin von Ausgang b als hoch setzen
  wiederhole fortlaufend
    digitalen Pin von Ausgang r als hoch setzen
    warte 10 Sekunde(n)
    digitalen Pin von Ausgang b als niedrig setzen
    warte 10 Sekunde(n)
    digitalen Pin von Ausgang g als hoch setzen
    warte 10 Sekunde(n)
    digitalen Pin von Ausgang r als niedrig setzen
    warte 10 Sekunde(n)
    digitalen Pin von Ausgang b als hoch setzen
    warte 10 Sekunde(n)
    digitalen Pin von Ausgang g als niedrig setzen
    warte 10 Sekunde(n)
```

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* und *b* bezeichnen die drei Anschlusspins der RGB-LED. So ist im Programm leicht zu erkennen, welche Farbkomponenten ein- oder ausgeschaltet werden. Außerdem hat diese Methode den Vorteil, dass sie leicht auf andere Pins umgebaut werden kann. Die Pinnummern

sind nur an einer Stelle im Programm, bei der Zuweisung der Variablen, eingetragen.

Die Geschwindigkeit des Farbwechsels wird auch hier über die Variable *zeit* gesteuert, die mithilfe eines Schiebereglers in Zehntelsekunden eingestellt werden kann und dann bei jedem Farbwechsel abgefragt wird.

 Das Programm ist unter dem Namen `13rgbled.mblock` bei den Downloads zu finden.

Tag 14

HEUTE IM ADVENTSKALENDER

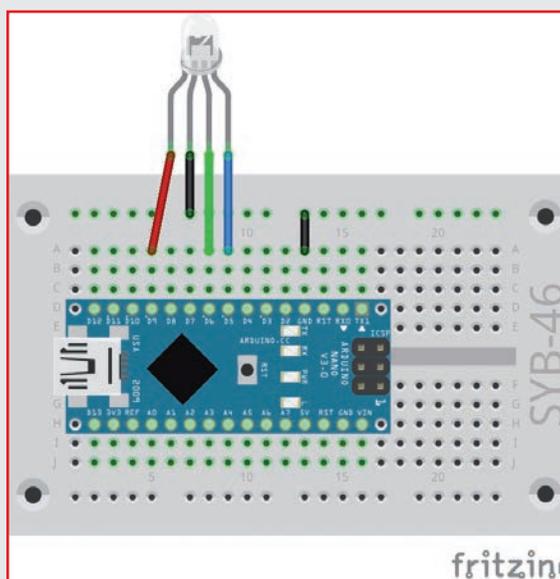
- 1 Verbindungskabelpaar

AUFGABE

Über PWM-Signale lassen sich auf RGB-LEDs noch deutlich mehr Farben anzeigen. Über drei Schieberegler sollen die drei Farbkomponenten unabhängig voneinander eingestellt werden können.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 RGB-LED mit Vorwiderständen
- 2 Verbindungskabelpaare
- 1 Drahtbrücke



Der Schaltungsaufbau ist der gleiche wie am vorherigen Tag.

DAS PROGRAMM

Die Variablen *r*, *g* und *b* bezeichnen wieder die drei Anschlusspins der RGB-LED. Die Variablen *rot*, *grün* und *blau* zeigen auf der Bühne Schieberegler, mit denen sich die drei Farbkomponenten einstellen lassen.



SO FUNKTIONIERT DAS PROGRAMM

Am Anfang werden die drei Farben der RGB-LEDs auf 0 gesetzt, also ausgeschaltet. Erst wenn die Schieberegler betätigt werden, leuchtet die RGB-LED.

Die Endlosschleife setzt die PWM-Pins der drei Farbkomponenten auf die über die Schieberegler eingestellten Werte. Danach wartet sie kurz eine Hundertstelse-

kunde und liest die Werte der Schieberegler wieder neu ein. So kann die Farbe der RGB-LED vom Benutzer in Echtzeit verändert werden.



Das Programm ist unter dem Namen `14rgbpwm.mblock` bei den Downloads zu finden.

Tag 15

HEUTE IM ADVENTSKALENDER

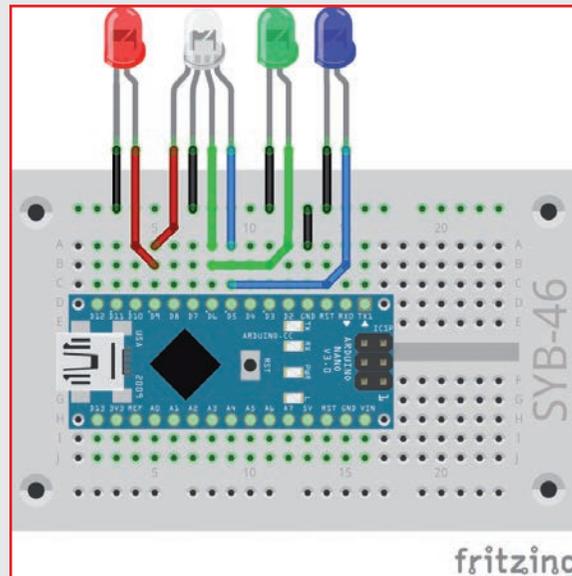
- 1 LED blau mit Vorwiderstand

AUFGABE

Das Programm soll eine RGB-LED in allen Farben des Farbspektrums zyklisch leuchten lassen. Drei einfarbige LEDs zeigen die eingestellten Farbkomponenten einzeln an.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 RGB-LED mit Vorwiderständen
- 1 LED blau mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 1 LED rot mit Vorwiderstand
- 5 Verbindungskabelpaare
- 1 Drahtbrücke



An den verwendeten PWM-Pins D5, D6 und D9 sind jeweils eine Farbkomponente der RGB-LED sowie eine einfarbige LED angeschlossen.

DAS PROGRAMM

Die Hauptschleife des Programms errechnet nacheinander die RGB-Werte für alle Farbwerte des HSV-Farb-

spektrums. Bei jedem Wert leuchtet die RGB-LED 0,05 Sekunden in der entsprechenden Farbe.

```

wenn geklickt wird
  setze zeit auf 0.05
  setze r auf 9
  setze g auf 6
  setze b auf 5
  PWM-Ausgang r als 0 festlegen
  PWM-Ausgang g als 0 festlegen
  PWM-Ausgang b als 0 festlegen
  wiederhole fortlaufend
    setze h auf 0
    wiederhole 360
      setze x auf 1 - Betrag von h / 60 mod 2 - 1 * 255
      wenn h < 360, dann
        setze rot auf 255
        setze gruen auf 0
        setze blau auf x
      wenn h < 300, dann
        setze rot auf x
        setze gruen auf 0
        setze blau auf 255
      wenn h < 240, dann

```

```

  setze rot auf 0
  setze gruen auf x
  setze blau auf 255
  wenn h < 180, dann
    setze rot auf 0
    setze gruen auf 255
    setze blau auf x
  wenn h < 120, dann
    setze rot auf x
    setze gruen auf 255
    setze blau auf 0
  wenn h < 60, dann
    setze rot auf 255
    setze gruen auf x
    setze blau auf 0
  PWM-Ausgang r als rot festlegen
  PWM-Ausgang g als gruen festlegen
  PWM-Ausgang b als blau festlegen
  warte zeit Sekunde(n)
  ändere h um 1

```

HSV- und RGB-Farbsystem

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

Durch eine einfache Veränderung des H-Werts können alle Farben des Farbspektrums in voller Intensität beschrieben werden, wenn man die beiden anderen Werte auf Maximum einstellt. Das Grafikmodul in mBlock 5, mit dem man Kostüme und Hintergründe zeichnen kann, verwendet eine HSV-Palette, um eine bestimmte Farbe mithilfe ihrer HSV-Parameter auszuwählen.



SO FUNKTIONIERT DAS PROGRAMM

Die drei Variablen *r*, *g* und *b* enthalten die als PWM-Ausgänge definierten Anschlusspins für die RGB-LED. Zusätzlich werden die drei Variablen *rot*, *gruen* und *blau* angelegt, in denen während der Berechnung die PWM-Werte für die drei Farbkomponenten der RGB-LED eingetragen werden.

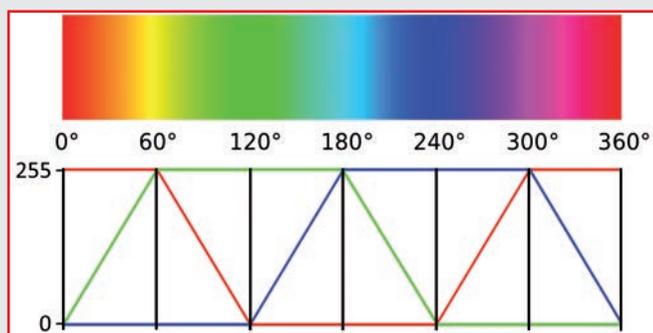
Die Hauptschleife zählt den H-Wert entsprechend den Gradzahlen auf einem Farbkreis zwischen 0 und 360 hoch und errechnet daraus die drei Farbkomponenten R, G und B. Die Werte S = Saturation (Sättigung) und V = Value (Helligkeitswert) werden automatisch als Maximalwerte gesetzt. Wie die Grafik zeigt,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

die sechs 60-Grad-Bereiche jeweils eigene lineare Verlaufskurven.

Nachdem die RGB-Werte errechnet und in den Variablen *rot*, *gruen* und *blau* gespeichert wurden, 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 Grad beginnt der Zyklus von Neuem bei 0.



Das Programm ist unter dem Namen `15rgbpwm.mblock` bei den Downloads zu finden.



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

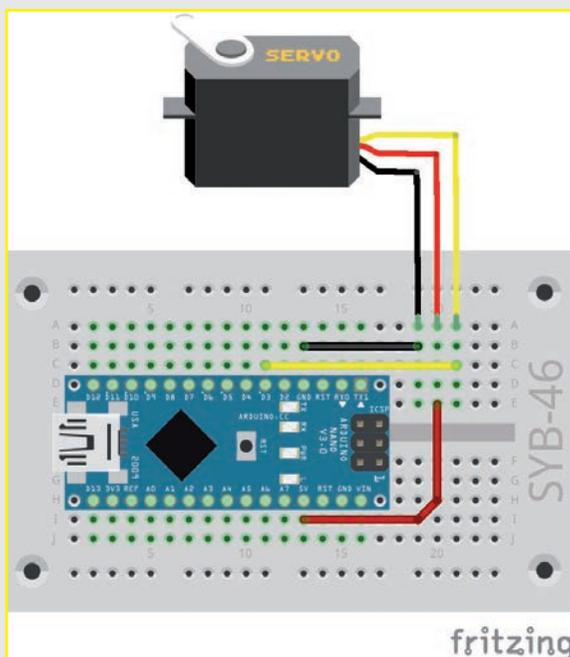
Tag 16

HEUTE IM ADVENTSKALENDER

- 1 Servo
- 1 Pinleiste 3 polig

SERVO

Ein Servo ist ein kleiner Motor mit einem Arm, an dem Gegenstände befestigt werden können. Mit einem Steuerungssignal kann er um einen bestimmten Winkel gedreht werden. Die meisten Servos haben einen Drehbereich von 180 Grad, drehen also nicht, wie klassische Motoren, nacheinander viele Runden in Folge, sondern können mit dem Steuersignal in eine bestimmte Position gebracht werden.



Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 Servo
- 1 Pinleiste 3 polig
- 3 Drahtbrücken

Der Servo wird in unseren Experimenten über GPIO-Verbindungskabel und eine Pinleiste auf dem Steckbrett am Nano angeschlossen. Diese Pinleiste passt genau in das Anschlusskabel des Servos. Servos haben drei Anschlussleitungen. Die rote und die schwarze Leitung dienen der Stromversorgung (Rot = + und Schwarz = -), über die gelbe Leitung empfängt der Servo die Steuerungssignale. Sie ist an Pin D3 angeschlossen.

AUFGABE

Die M-Panda-Figur soll den Drehwinkel des Servos steuern. Diese Figur kann auf der Bühne mit der Maus bewegt werden.

Die heutige Aufgabe lässt sich am besten ohne das Haus erledigen. Nimm das Steckbrett also aus dem Häuschen. Morgen baust du den Servo dann ins Haus ein.

DAS PROGRAMM

mBlock 5 bietet einen speziellen Block *set Servo-Pin ... angle as ...* zur Servosteuerung. In anderen Programmiersprachen wird es über PWM-Signale gelöst. Dieser Block ist in der deutschen mBlock-5-Einstellung nicht übersetzt.

Das Bühnenbild zeigt einen Kreisbogen. Verschiebe die M-Panda-Figur darauf hin und her, um den Drehwinkel des Servos entsprechend einzustellen. Die Figur verwendet dazu eigene Blöcke, die die Position der Figur auf der Bühne ständig abfragen.



SO FUNKTIONIERT DAS PROGRAMM

In jedem Durchlauf der Endlosschleife im Programm der Figur wird die Variable *winkel* auf den für den Servo berechneten Winkel gesetzt und auf der Bühne angezeigt. Der Block *X-Position von M-Panda* von der Blockpalette *Fühlen* liefert die aktuelle x-Position der M-Panda-Figur. Ob sich diese Figur tatsächlich auf dem Kreisbogen befindet oder einfach waagrecht verschoben wird, spielt dabei keine Rolle.

Diese x-Position kann auf der Bühne zwischen -240 und 240 liegen. Zwei Blöcke von der Blockpalette *Operatoren* rechnen den Wert in einen Winkel zwischen 0 und 180 Grad um.

Das Programm für das Gerät läuft ebenfalls in einer Endlosschleife. Der Block *set Servo-Pin ... angle as ...* von der Blockpalette *Pin* stellt den Servo auf den ausgerechneten und in der Variablen *winkel* gespeicherten Winkel ein. Dafür braucht man keine Nachricht von der Figur an das Gerät zu senden, da die Variable für alle Figuren und Geräte gilt und einfach nur in jedem Schleifendurchlauf ausgelesen wird.



Das Programm ist unter dem Namen *16servo*. *mBlock* bei den Downloads zu finden.

Tag 17

HEUTE IM ADVENTSKALENDER

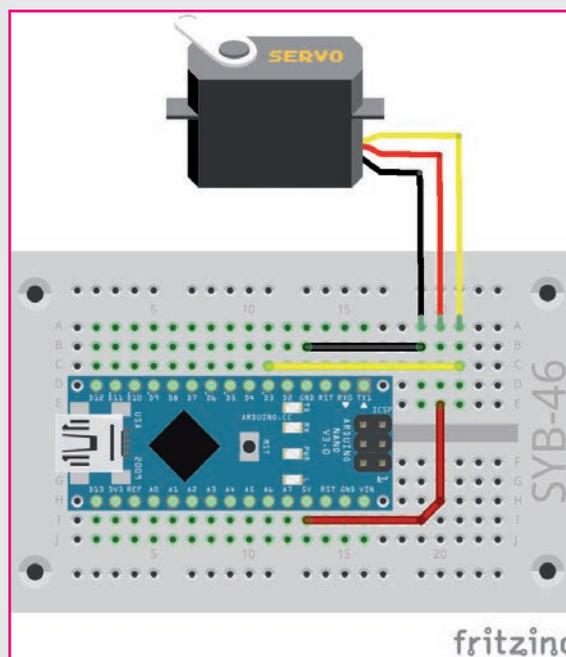
- 1 Drahtgestänge für Servo

AUFGABE

Der Stern auf dem Haus soll mit dem Servo gedreht werden.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 Servo
- 1 Pinleiste 3 polig
- 3 Drahtbrücken



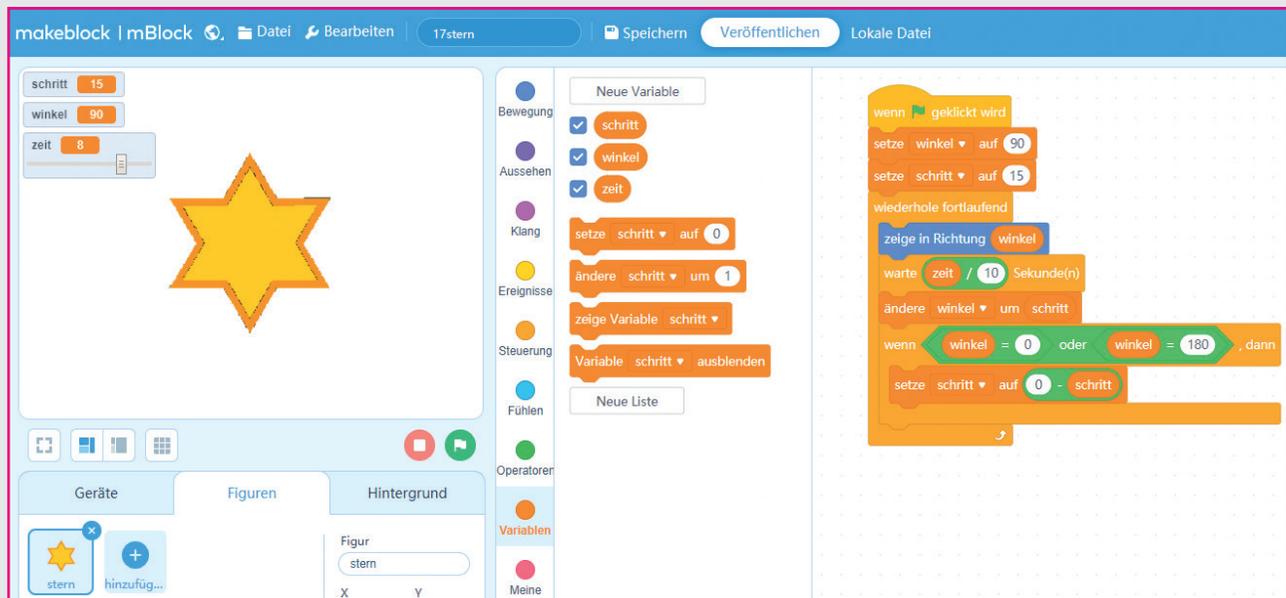
Der Schaltungsaufbau ist der gleiche wie am vorherigen Tag.



DAS PROGRAMM

Das Programm verwendet eine neue Figur, einen Stern. Er dreht sich schrittweise in einem Winkel zwischen 0

und 180 Grad. Der Servo dreht den Stern auf dem Haus im gleichen Winkel.



SO FUNKTIONIERT DAS PROGRAMM

Das Programm verwendet drei Variablen:

schrift gibt den Winkel eines Schritts beim Drehen an.

winkel bezeichnet den Winkel, in dem der Stern und der Servo stehen.

zeit legt die Geschwindigkeit fest, also die Wartezeit in Zehntelsekunden.



Das Programm enthält eine neue Figur in Form eines Sterns. Diese Figur ist als Grafik *stern.svg* im Download enthalten. Lösche die Figur *M-Panda* und klicke auf das blaue Symbol *hinzufigen*. Klicke im nächsten Fenster oben links auf *Hochladen* und lade den Stern aus dem Download ins Programm.

Das Programm für diese Figur enthält weitgehend bekannte Elemente. Nachdem die Variablen *winkel* und *schrift* in eine Grundeinstellung gebracht wurden, startet eine Endlosschleife, die in jedem Durchlauf den Stern in den durch die Variable *winkel* festgelegten Winkel bringt. Dazu wird der Block *zeige in Richtung ...* aus der Blockpalette *Bewegung* verwendet. Er dreht eine Figur in einen bestimmten absoluten Winkel – anders als die *drehe ... um ... Grad*-Blöcke, die die Figur relativ zum aktuellen Winkel weiter im oder gegen den Uhrzeigersinn drehen.

Anschließend wartet das Programm die eingestellte Zeit. Da die Variable *zeit* wie schon in früheren Programmen in Zehntelsekunden eingestellt wird, wird sie wieder durch 10 geteilt.

Nach dieser Pause wird der Winkel um den vorgegebenen Schritt von 15 Grad erhöht. Wird eine der Grenzen von 0 Grad oder 180 Grad erreicht, wird die Drehrichtung umgekehrt. Danach wiederholt sich die Schleife.

Das Gerät verwendet ebenfalls eine Endlosschleife, die Pin D3, an den das Servo angeschlossen ist, auf den in der Variablen *winkel* gespeicherten Winkel setzt. Auch hier sind wieder keine Nachrichten oder Wartezeiten in der Schleife nötig. Das wird alles durch das Programm der Figur gesteuert.



Das Programm ist unter dem Namen *17stern.mblock* bei den Downloads zu finden.

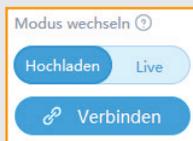
HEUTE IM ADVENTSKALENDER

- 1 Verbindungskabelpaar

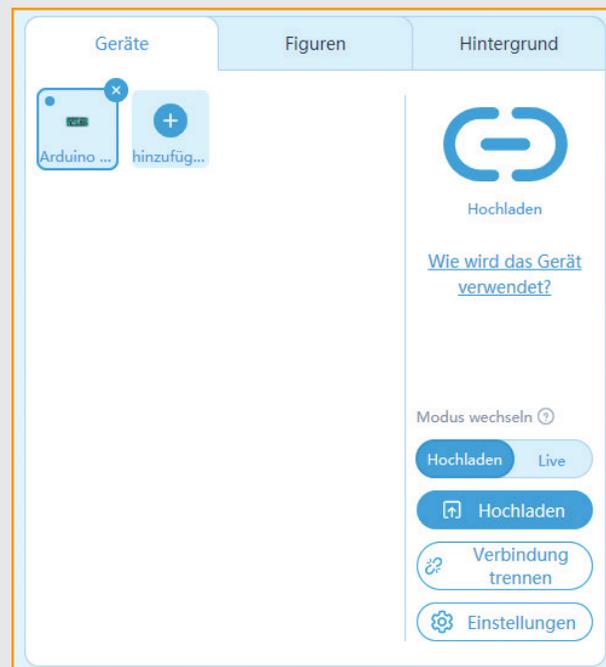
Das Nano-Board kann auch ohne PC genutzt werden und ein gespeichertes Programm abarbeiten. Dazu benötigt es eine externe Stromversorgung. Das kann ein USB-Handyladegerät, eine Powerbank oder eine Batterie sein.

Im Gegensatz zum interaktiven Live-Modus erzeugt mBlock 5 im Arduino-Modus, auch als Hochladen-Modus bezeichnet, ein Programm, das einmal auf das Nano-Board übertragen wird und dann dort ohne Verbindung zum PC läuft. Dieser Modus empfiehlt sich besonders bei zeitkritischen Anwendungen, da es im Live-Modus durch die Echtzeitkommunikation mit dem PC zu unberechenbaren Verzögerungen kommen kann.

Schalte auf der Registerkarte *Geräte* ganz unten von *Live* auf *Hochladen* um und klicke dann auf *Verbinden*.



Wähle hier wieder die COM-Schnittstelle aus. Jetzt erscheinen auf der Registerkarte *Geräte* neue Schaltflächen. Auch auf den Blockpaletten werden einige Blöcke deaktiviert und dafür andere freigeschaltet.



Firmware für mBlock 5 neu installieren

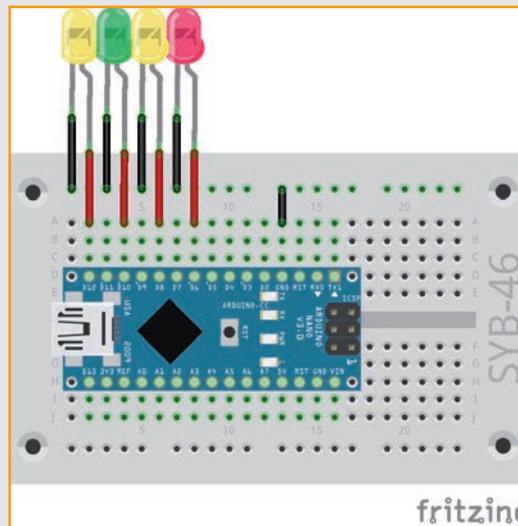
Wenn du zwischendurch ein Programm im Arduino-Modus von mBlock 5 oder ein anderes Arduino-Programm verwendet hast, muss, bevor du mBlock 5 wieder im Live-Modus benutzen kannst, die Firmware erneut installiert werden. Sobald der Nano im Live-Modus verbunden wurde, erscheint unten ein orangefarbener Button *Aktualisieren*. Klicke darauf und dann auf *Firmware aktualisieren*. Klickst du im nächsten Fenster auf *Aktualisierungen*, wird die Firmware heruntergeladen und installiert. Verbinde anschließend den Nano neu.

AUFGABE

Vier LEDs auf dem Hausdach sollen gleichzeitig blinken.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 2 LEDs gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 1 LED rot mit Vorwiderstand
- 4 Verbindungskabelpaare
- 1 Drahtbrücke



DAS PROGRAMM IM ARDUINO-MODUS

Das Programm ähnelt einem Programmblock aus dem Programm des 12. Tages. Wenn du es später laufen lässt, wirst du merken, dass der Arduino-Modus deutlich schneller ist als der interaktive Modus von mBlock 5. Es sind keine Verzögerungen zwischen den LEDs zu erkennen.



SO FUNKTIONIERT DAS PROGRAMM

Jedes Programm im Arduino-Modus beginnt mit dem Block *wenn Arduino startet*. Blöcke, die Objekte auf der Bühne, Tasten der PC-Tastatur oder Mauseingaben verwenden, können im Arduino-Modus nicht genutzt werden. Ebenso wenig können Variablen über Schie-

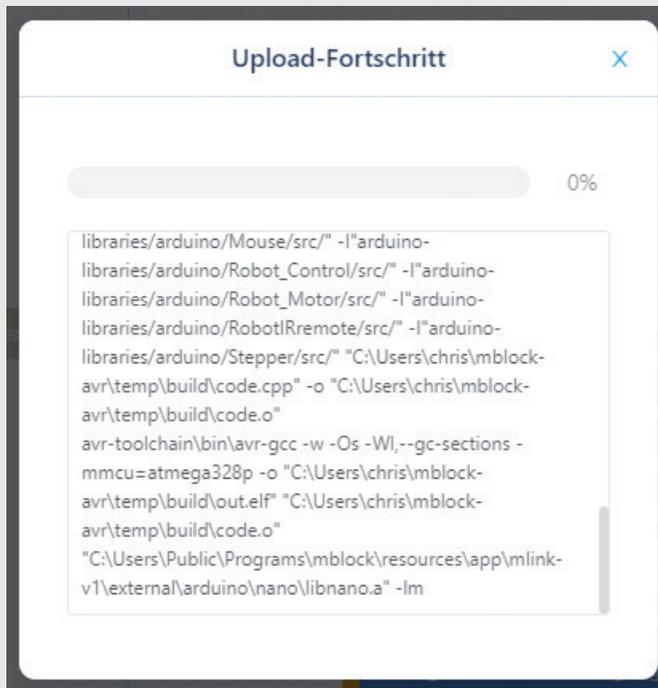
bereger verändert werden, während das Programm läuft.

Die weiteren Blöcke in diesem Programm sind bereits aus früheren Programmen bekannt.

PROGRAMM AUF DEN NANO ÜBERTRAGEN

Klicke unten auf der Registerkarte *Geräte* auf *Hochladen*, um das Programm auf den Nano zu übertragen. Der *Upload-Fortschritt* ist in einem Fenster zu sehen.

Danach beginnen die vier LEDs zu blinken. Im Arduino-Modus blinken sie wirklich gleichzeitig, da die Verzögerungen zwischen den einzelnen Blöcken *digitalen Pin von Ausgang ... als hoch setzen* so gering sind, dass sie dem Auge nicht auffallen.



Jedes Mal, wenn du die Taste auf dem Nano drückst oder die Stromversorgung am USB-Anschluss trennst und wieder neu ansteckst, startet das Programm erneut.



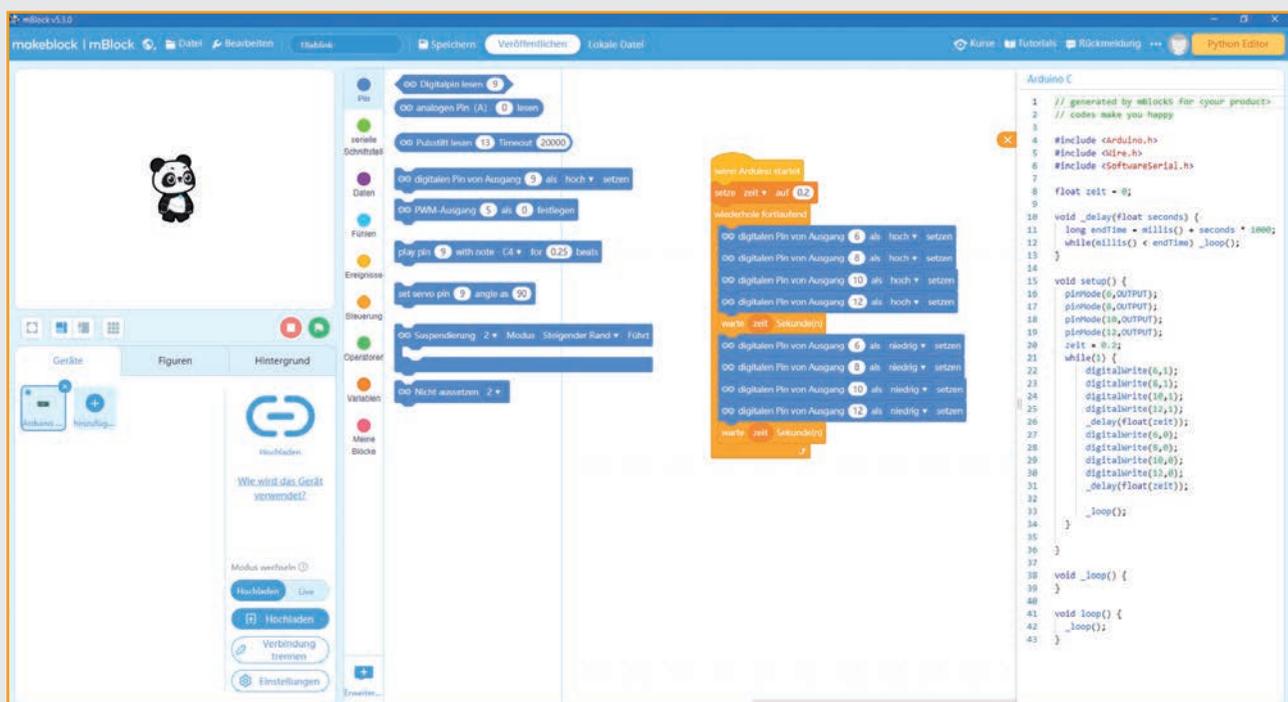
Möchtest du wissen, wie ein echtes Arduino-Programm aussieht, klicke auf das orangefarbene Symbol `</>` rechts oben.

Der rechte Teil des mBlock-Bildschirms zeigt dann den automatisch generierten Arduino-Programmcode. Er wird, ebenfalls automatisch, kompiliert und auf das Nano-Board übertragen.

Klicke in der Meldung *Starte Upload* auf *Schließen*, um die Übertragung mitverfolgen zu können.



Das Programm ist unter dem Namen `18ablink.mblock` bei den Downloads zu finden.



Tag 19

HEUTE IM ADVENTSKALENDER

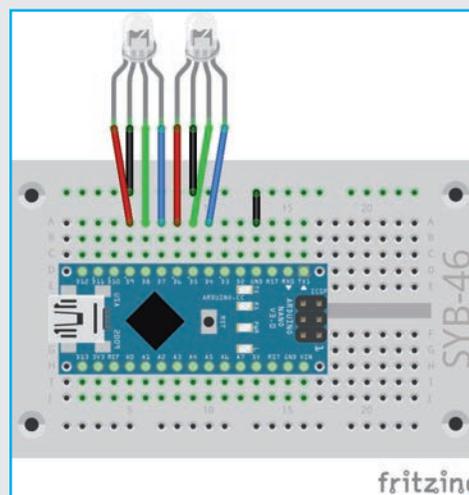
- 1 RGB-LED mit Vorwiderständen

AUFGABE

Zwei RGB-LEDs auf dem Hausdach sollen in zufälligen Farben blinken.

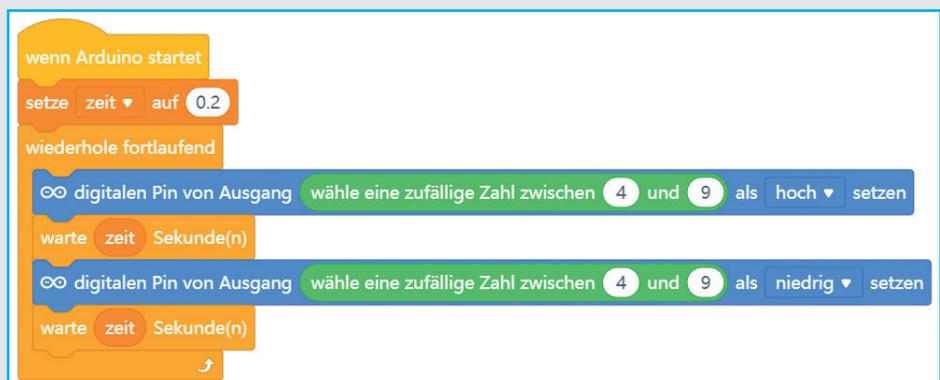
Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 2 RGB-LEDs mit Vorwiderständen
- 4 Verbindungskabelpaare
- 1 Drahtbrücke



DAS PROGRAMM

Das Programm zeigt in einer Endloschleife auf den RGB-LEDs zufällige Farben. Es läuft im Arduino-Modus von mBlock 5. Das Haus kann ohne Verbindung mit dem PC betrieben werden.



SO FUNKTIONIERT DAS PROGRAMM

In jedem Schleifendurchlauf wird ein zufällig gewählter Pin der sechs Pins, an denen die beiden RGB-LEDs angeschlossen sind, eingeschaltet, und nach einer eingestellten Zeit wird wieder ein zufällig gewählter Pin ausgeschaltet. Bei dieser Methode kann es passieren, dass sich in einem Schleifendurchlauf die angezeigten Farben nicht ändern oder auch einmal gar keine Farbe

leuchtet. Das Blinkmuster erscheint unregelmäßig und zufällig.



Das Programm ist unter dem Namen 19a_rgb2led.mblock bei den Downloads zu finden.

Tag 20

HEUTE IM ADVENTSKALENDER

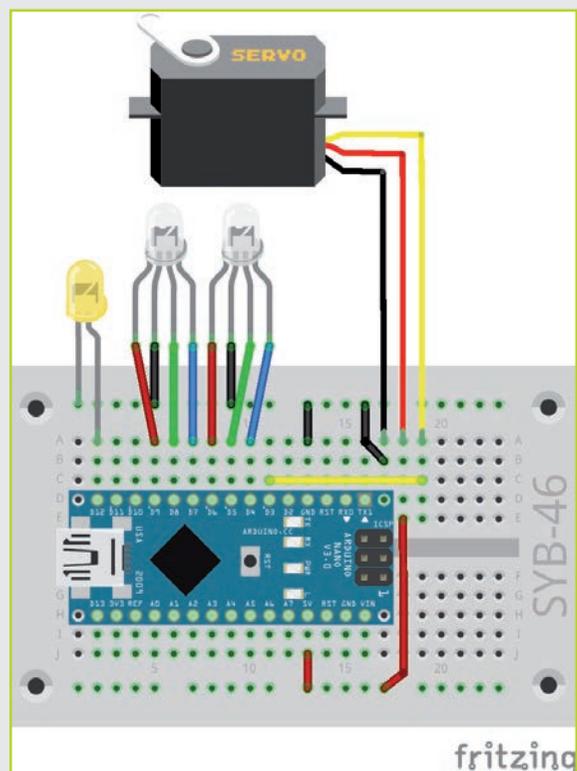
- 1 Verbindungskabelpaar

AUFGABE

Wenn du auf das grüne Fähnchen klickst, geht im Haus das Licht an, nach kurzer Zeit dreht sich der Stern, und die LEDs blinken bunt. Einige Sekunden später gehen alle LEDs aus, und der Stern dreht sich wieder zurück.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 Servo
- 1 Pinleiste 3 polig
- 2 RGB-LEDs mit Vorwiderständen
- 1 LED gelb mit Vorwiderstand
- 4 Verbindungskabelpaare
- 5 Drahtbrücken



DAS PROGRAMM

Das Programm besteht im Wesentlichen aus bekannten Blöcken, die bei Klick auf das grüne Fähnchen nacheinander abgearbeitet werden.

Das Programm läuft wieder im Live-Modus von mBlock 5.

```
wenn geklickt wird
  setze servo auf 3
  setze led auf 12
  digitalen Pin von Ausgang led als hoch setzen
  setze winkel auf 0
  wiederhole 6
    sende Stern drehen an alle
    set servo pin servo angle as winkel
    warte zeit Sekunde(n)
    ändere winkel um 30
  wiederhole w
    digitalen Pin von Ausgang wähle eine zufällige Zahl zwischen 4 und 6 als hoch setzen
    digitalen Pin von Ausgang wähle eine zufällige Zahl zwischen 7 und 9 als hoch setzen
    warte blinkzeit / 10 Sekunde(n)
    digitalen Pin von Ausgang wähle eine zufällige Zahl zwischen 4 und 6 als niedrig setzen
    digitalen Pin von Ausgang wähle eine zufällige Zahl zwischen 7 und 9 als niedrig setzen
    warte blinkzeit / 10 Sekunde(n)
  setze n auf 4
  wiederhole 6
    digitalen Pin von Ausgang n als niedrig setzen
    ändere n um 1
  warte zeit Sekunde(n)
  setze winkel auf 90
  sende Stern drehen an alle
  set servo pin servo angle as winkel
  digitalen Pin von Ausgang led als niedrig setzen
```

Firmware für mBlock 5 neu installieren

Wenn du zwischendurch ein Programm im Arduino-Modus von mBlock 5 oder ein anderes Arduino-Programm verwendet hast, muss, bevor du mBlock 5 wieder im Live-Modus benutzen kannst, die Firmware erneut installiert werden. Sobald der Nano im Live-Modus verbunden wurde, erscheint unten ein orangefarbener Button *Aktualisieren*. Klicke darauf und dann auf *Firmware aktualisieren*. Klickst du im nächsten Fenster auf *Aktualisierungen*, wird die Firmware heruntergeladen und installiert. Verbinde anschließend den Nano neu.

SO FUNKTIONIERT DAS PROGRAMM

Das Programm verwendet sieben Variablen:



blinkzeit – Legt in Zehntelsekunden fest, wie lange die LEDs beim Blinken ein- oder ausgeschaltet sein sollen.

led – Pinnummer der LED im Haus.

n – Zähler für eine Schleife.

servo – Pinnummer des Servos.

w – Anzahl der Wiederholungen beim Blinken.

winkel – Drehwinkel für Servo und Stern.

zeit – Wartezeit zwischen den Bewegungen des Sterns in Sekunden.

Die Variablen *blinkzeit*, *w* und *zeit* kannst du vor dem Start des Programms mit Schiebereglern einstellen.



Nach Klick auf das grüne Fähnchen wird als Erstes die LED im Haus eingeschaltet. Anschließend werden das Servo und damit der Stern auf dem Haus in sechs Schritten gedreht.

Danach beginnt eine Schleife, die die RGB-LEDs blinken lässt. Diese Schleife wird wiederholt. Möchtest du mehr oder weniger Wiederholungen, ändere den Wert der Variablen *w* mit dem Schieberegler.

In jedem Durchlauf der Schleife wird eine zufällig gewählte Farbe der ersten und eine zufällig gewählte Farbe der zweiten RGB-LED eingeschaltet. Nach der eingestellten Blinkzeit in Zehntelsekunden wird wieder eine zufällig gewählte Farbe der ersten und eine zufällig gewählte Farbe der zweiten RGB-LED ausgeschaltet. Auf diese Weise können verschiedene Mischfarben entstehen. Es ist aber auch möglich, dass sich die Farbe einer LED in einem Schleifendurchlauf nicht verändert oder eine LED ganz ausgeht.

Nach der festgelegten Zahl von Wiederholungen schaltet eine weitere Schleife alle sechs Farben der beiden RGB-LEDs aus. Danach dreht sich das Servo wieder in eine Grundstellung.

Der Stern auf der Bühne dreht sich gleichzeitig mit dem Servo. Das Hauptprogramm sendet bei jeder Drehung die Nachricht *Stern drehen*. Diese wird von der Figur empfangen. Ein Block *zeige in Richtung winkel* dreht den Stern auf die Position, die durch die Variable *winkel* in diesem Moment vorgegeben ist.



Um das Ganze wieder zu starten, klicke auf das grüne Fähnchen.



Das Programm ist unter dem Namen *20haus01*. mBlock bei den Downloads zu finden.

Tag 21

HEUTE IM ADVENTSKALENDER

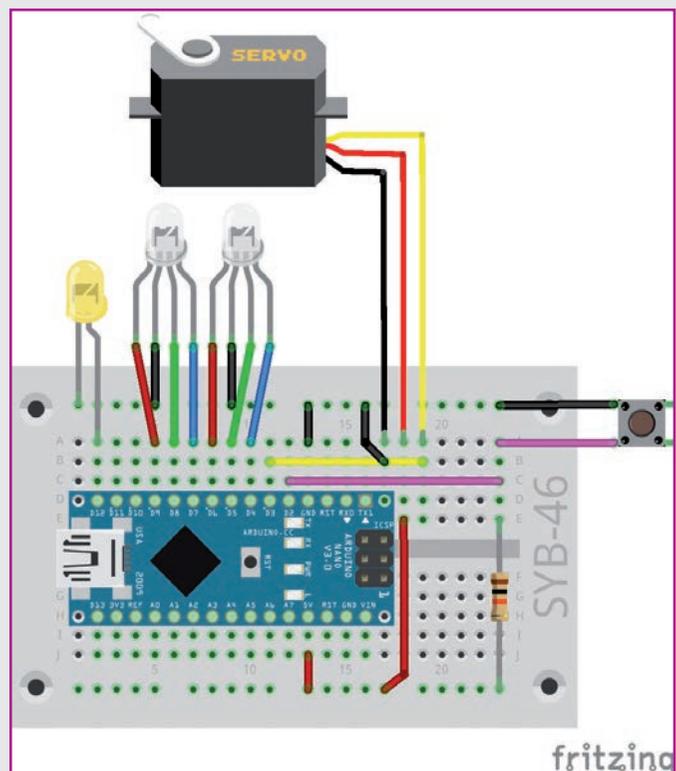
- 1 Verbindungskabelpaar

AUFGABE

Die Aufgabe von heute ist der von gestern sehr ähnlich. Allerdings sollen das Licht im Haus, der Stern und das Blinken der RGB-LEDs auf dem Dach nicht durch einen Klick auf das grüne Fähnchen ausgelöst werden, sondern dadurch, dass man den Klingeltaster an der Haustür drückt.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 Servo
- 1 Pinleiste 3 polig
- 2 RGB-LEDs mit Vorwiderständen
- 1 LED gelb mit Vorwiderstand
- 1 Taster
- 1 Widerstand 10 kOhm (braun, schwarz, orange)
- 5 Verbindungskabelpaare
- 6 Drahtbrücken

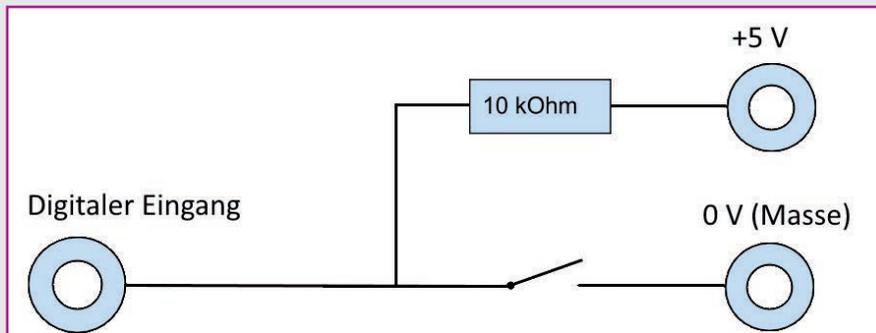


TASTER MIT PULL-UP-WIDERSTAND

Vielleicht ist dir bei den letzten Experimenten mit dem Taster aufgefallen, dass manchmal etwas im Programm passiert, ohne dass der Taster gedrückt wurde. Der Taster scheint sich selbst auszulösen.

Im Experiment heute benutzen wir statt des bisher verwendeten Pull-down-Widerstands einen Pull-up-Widerstand. Dieser zieht den Status des Pins an einem nicht

gedrückten Taster nach oben auf *HIGH*. Am digitalen Eingangspin kommt also immer ein *HIGH*-Signal an. Der Taster muss beim Drücken mit der Masseleitung verbunden werden und liefert dann ein *LOW*-Signal. Das Programm muss also umgekehrt fragen, ob der Pin *LOW* und nicht *HIGH* ist. Das bezeichnet man als inverse Logik.



Taster mit Pull-up-Widerstand. Ist der Taster gedrückt, wird der digitale Eingang mit Masse verbunden.

DAS PROGRAMM

Das Programm ist ähnlich aufgebaut wie das von gestern. Der wichtigste Unterschied ist eine Endlosschleife, die den Zustand des Tasters abfragt.

```

wenn geklickt wird
  setze taste auf 2
  setze servo auf 3
  setze led auf 12
  wiederhole fortlaufend
    wenn nicht Digitalpin lesen taste dann
      Digitalpin von Ausgang led als hoch setzen
      setze winkel auf 0
      wiederhole 6
        sende Stern drehen an alle
        set servo pin servo angle as winkel
        warte zeit Sekunde(n)
        ändere winkel um 30
      wiederhole w
        Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 4 und 6 als hoch setzen
        Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 7 und 9 als hoch setzen
        warte blinkzeit / 10 Sekunde(n)
        Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 4 und 6 als niedrig setzen
        Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 7 und 9 als niedrig setzen
        warte blinkzeit / 10 Sekunde(n)
      setze n auf 4
      wiederhole 6
        Digitalpin von Ausgang n als niedrig setzen
        ändere n um 1
      warte zeit Sekunde(n)
      setze winkel auf 90
      sende Stern drehen an alle
      set servo pin servo angle as winkel
      Digitalpin von Ausgang led als niedrig setzen
      warte 0,1 Sekunde(n)
  
```

SO FUNKTIONIERT DAS PROGRAMM



Bei Klick auf das grüne Fähnchen werden nur die Variablen für die Pins von Taster, Servo und LED initialisiert. Dazu wird im Programm eine neue Variable *taste* angelegt die den Pin D2 bezeichnet.

Jetzt beginnt eine Endlosschleife, die den Taster jede Zehntelsekunde abfragt. Wegen des Pull-up-Widerstands wird hier inverse Logik benötigt. Die *wenn ... dann ...*-Abfrage muss prüfen, ob am Pin, der durch die Variable *taste* festgelegt ist, ein *LOW*-Signal liegt. Der Block *Digitalpin lesen ...* liefert wahr, wenn der Pin *HIGH* ist. Dieser Block ist in einen Block *nicht ...* eingebaut, der den Logikwert genau umkehrt. Die *wenn ... dann ...*-Abfrage ist also dann wahr, wenn der Taster gedrückt ist und damit der Pin ein *LOW*-Signal hat.

Wurde der Taster gedrückt, beginnt die bekannte Szene: Das Licht im Haus geht an, der Servo dreht den Stern, und die RGB-LEDs auf dem Dach blinken bunt. Nach einer bestimmten Zeit gehen die RGB-LEDs aus, der Stern dreht sich wieder zurück, und das Licht im Haus erlischt.

Auch hier sendet das Hauptprogramm jedes Mal, wenn der Stern auf der Bühne gedreht werden soll, eine Nachricht an die Figur.

Die Endlosschleife läuft weiter und wartet wieder darauf, dass du auf den Taster drückst.



Das Programm ist unter dem Namen `21haus02.mblock` bei den Downloads zu finden.

Tag 22

HEUTE IM ADVENTSKALENDER

- 1 Verbindungskabelpaar

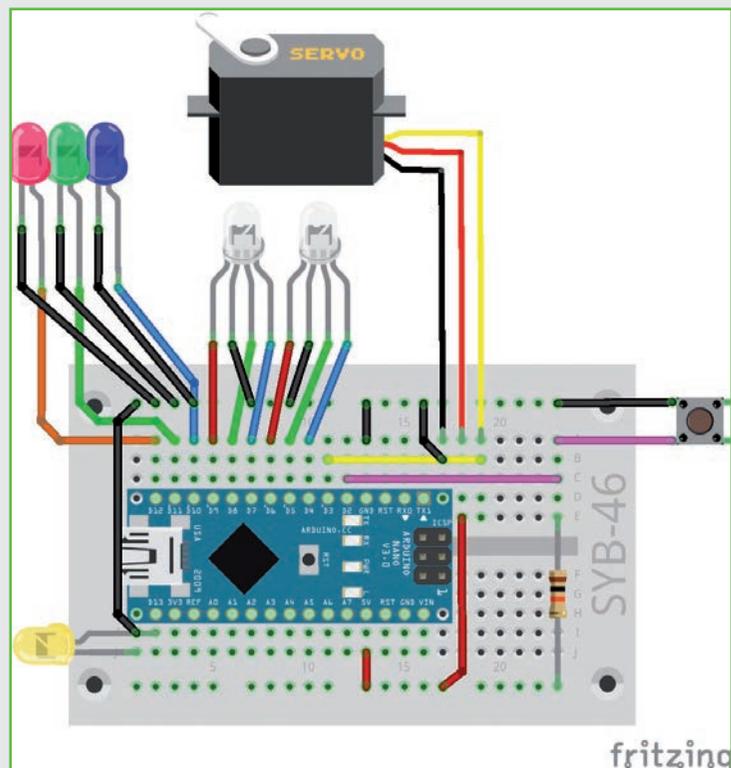
AUFGABE

Das Licht im Haus, die RGB-Blinklichter auf dem Dach, der Stern und ein zusätzliches Lauflicht aus drei LEDs auf dem Dach sind über ein Steuerpult auf dem Bildschirm einstellbar. Wenn man auf den Klingeltaster am Haus drückt, läuft die Szene ab.

Die gelbe LED ganz links unten ist an Pin D13 angeschlossen. Er befindet sich als einziger digitaler Pin auf der gegenüberliegenden Seite auf dem Nano-Board, eben dort, wo die mit „A“ bezeichneten analogen Pins liegen. Immer wenn diese LED leuchtet, leuchtet auch die eingebaute LED auf dem Nano, die ebenfalls mit Pin D13 verbunden ist.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 Servo
- 1 Pinleiste 3 polig
- 2 RGB-LEDs mit Vorwiderständen
- 1 LED rot mit Vorwiderstand
- 1 LED gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 1 LED blau mit Vorwiderstand
- 1 Taster
- 1 Widerstand 10 kOhm (braun, schwarz, orange)
- 8 Verbindungskabelpaare
- 7 Drahtbrücken



STEUERPULT FÜR DAS HAUS IM VOLLBILDMODUS

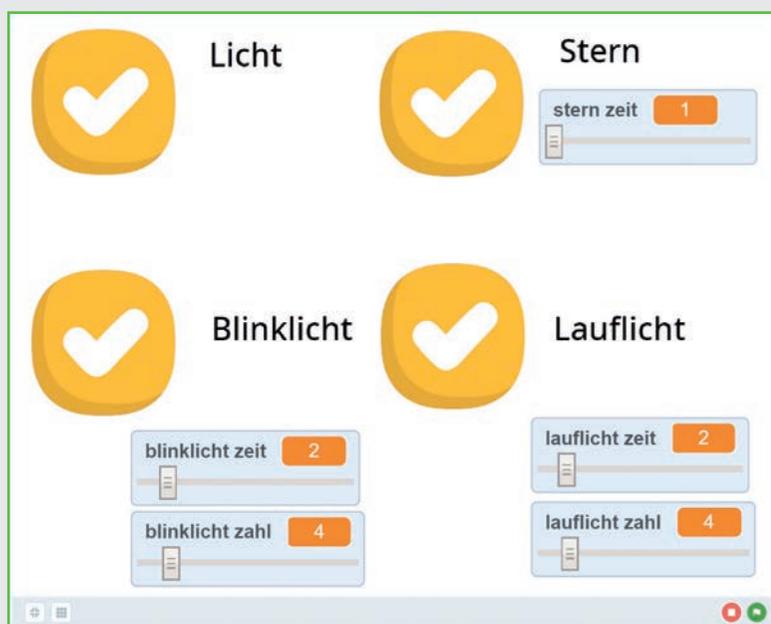
Vier Buttons auf der Bühne schalten die einzelnen Szenenelemente Licht, Stern, Blinklicht und Lauflicht ein oder aus, die nacheinander ablaufen, wenn der Taster gedrückt wird. Ausgeschaltete Szenenelemente werden einfach übersprungen.



Ein Klick auf das blaue Vollbildsymbol links unterhalb der Bühne schaltet die Bühne in den Präsentationsmodus, damit man die

Bedienelemente in voller Größe sehen kann. Ein weiterer Vorteil dieses Modus ist, dass Figuren und auch die Schieberegler der Variablen nicht versehentlich auf der Bühne verschoben werden können.

Schalte die Bühne in den Präsentationsmodus, um die Figuren und Variablen nicht versehentlich zu verschieben. Mit dem Symbol unten links kommst du wieder zurück.



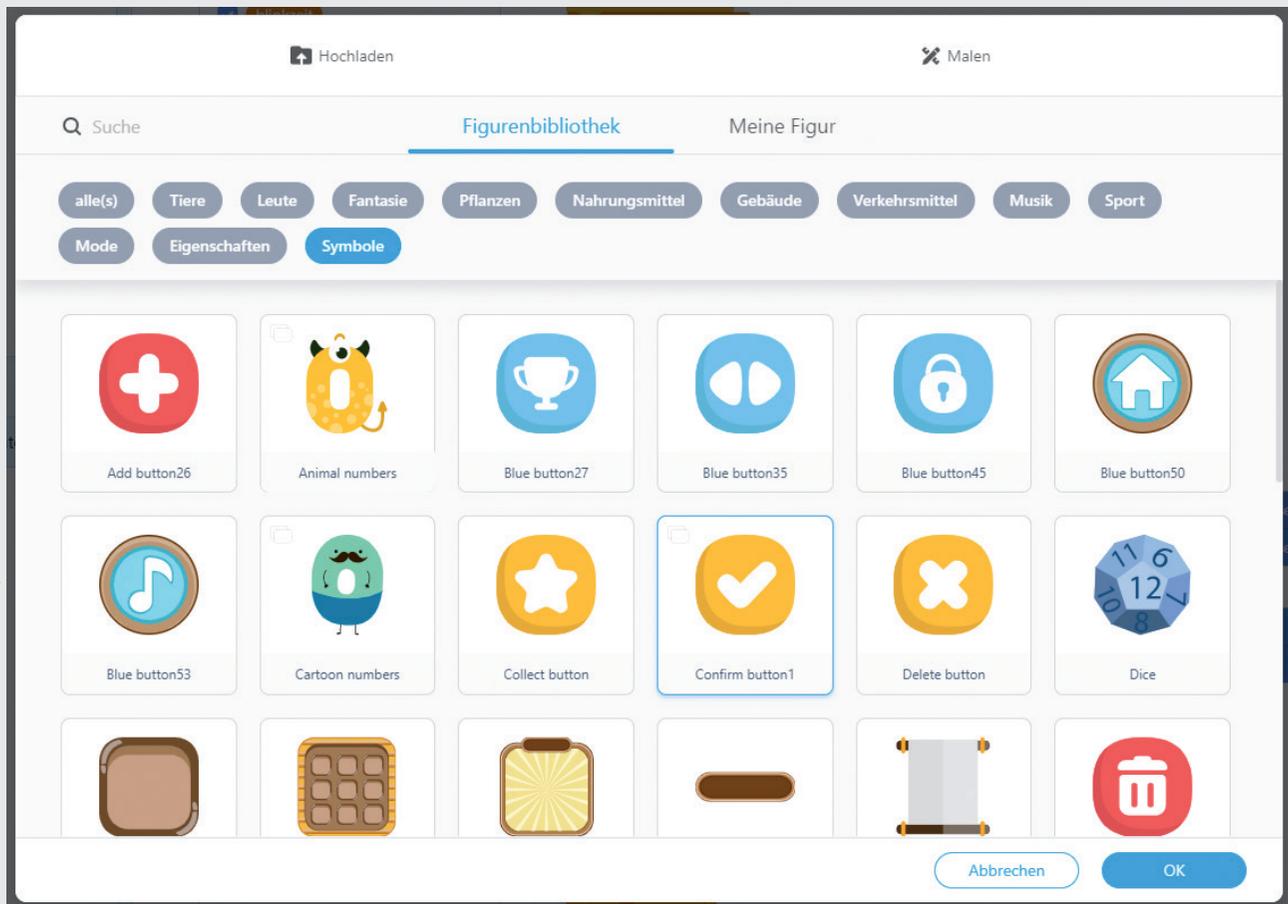
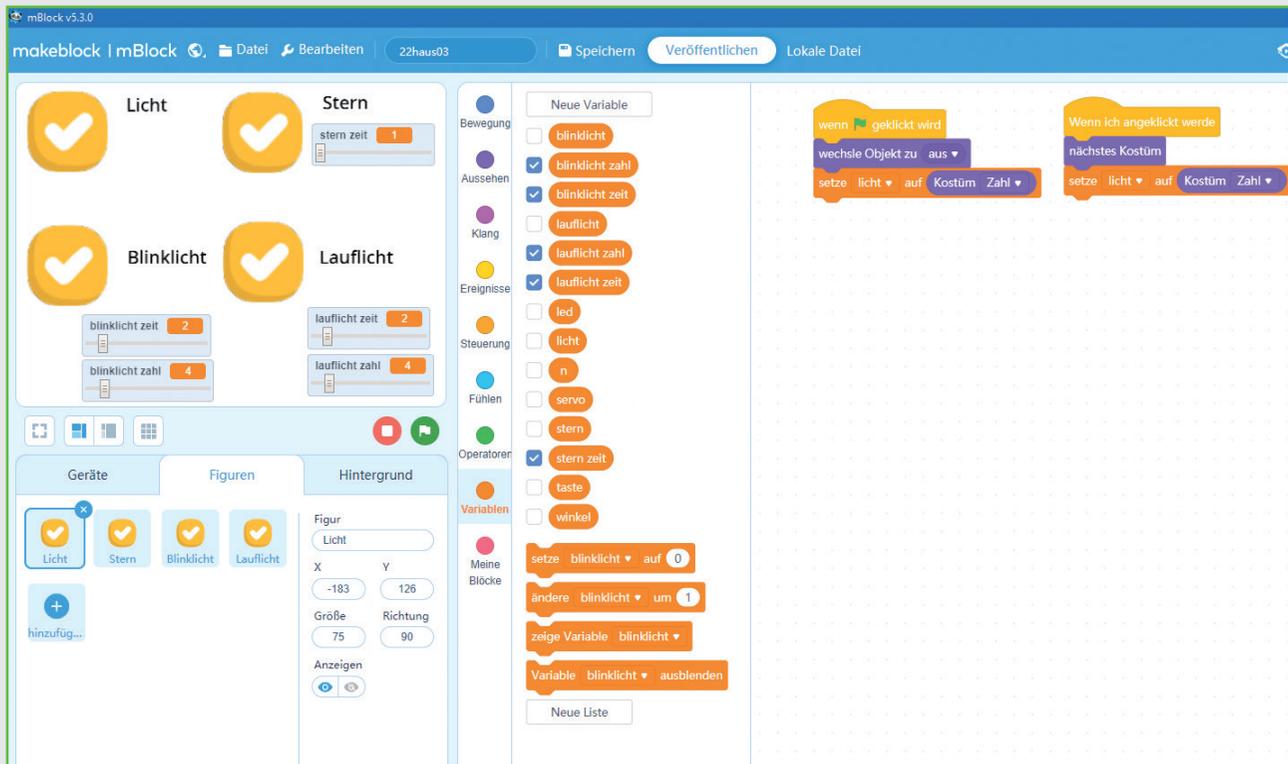
DIE FIGUREN

Das Programm besteht aus vier Figuren. Alle vier sind Buttons, die als Schalter funktionieren. Jede Figur hat ein gelbes und ein rotes Kostüm, die den ein- und den ausgeschalteten Zustand darstellen, sowie eigene Programmblöcke. Zusätzlich sind auf der Bühne über Schieberegler Variablen einstellbar.

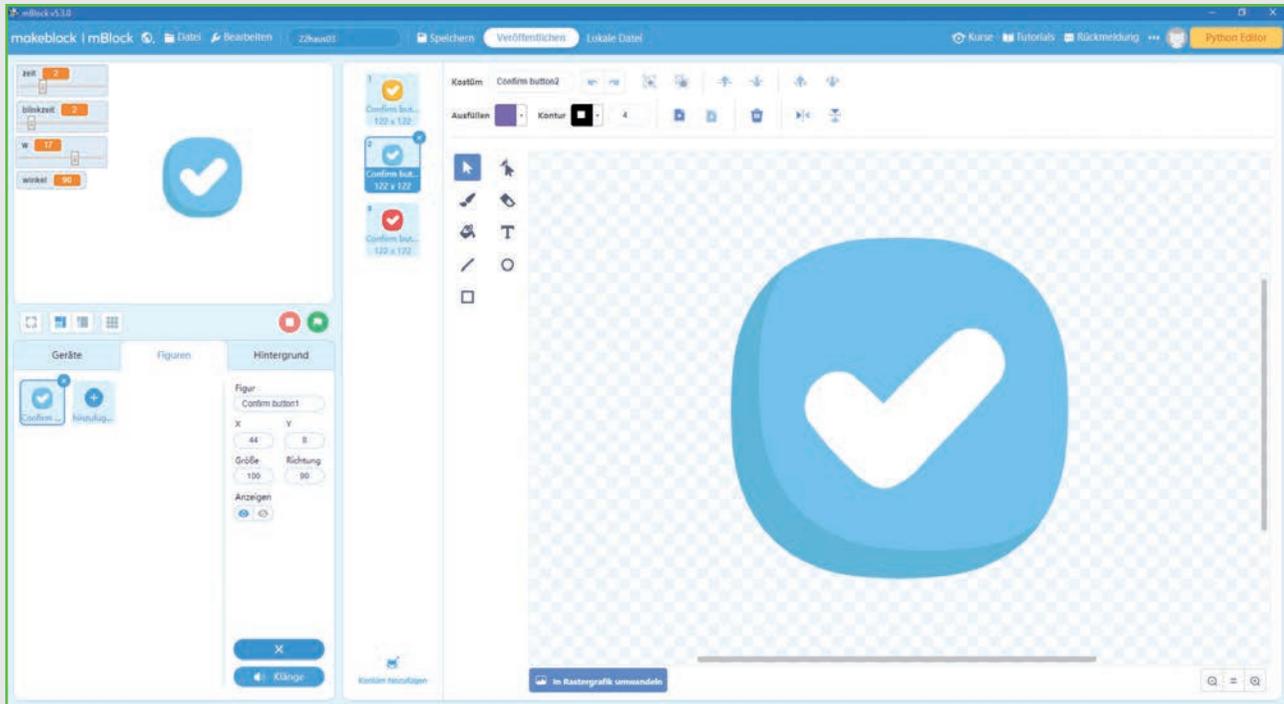
Wenn du das Programm anhand der Abbildungen selber zusammenbaust, stelle erst eine Figur mit ihren Kostümen und Programmblöcken komplett fertig. Anschlie-

ßend kannst du die fertige Figur duplizieren. Damit werden auch die Kostüme, Programmblöcke und Variablen dupliziert, und du brauchst sie nur noch entsprechend zu ändern und musst nicht alles neu bauen.

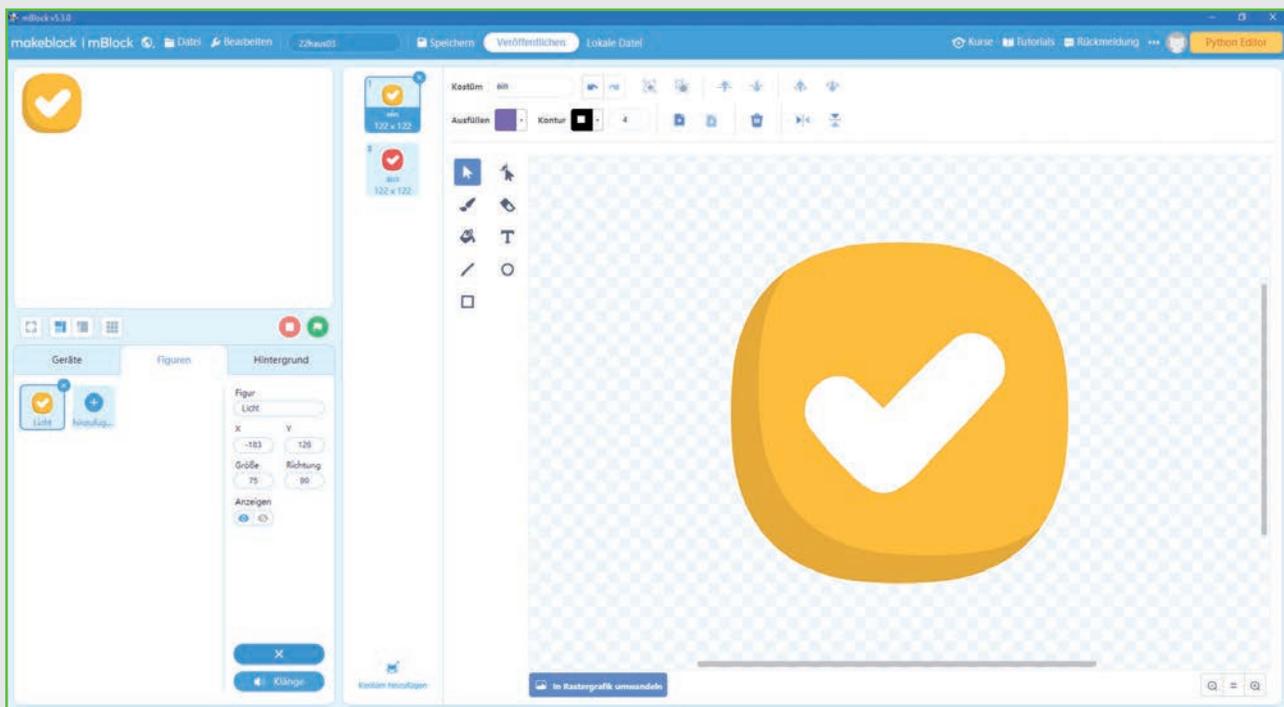
Die für die Figuren verwendeten Symbole sind in der Figurenbibliothek von mBlock 5 enthalten. Klicke dort oben auf *Symbole* und wähle dann den *Confirm button1*.



Das Symbol hat drei Kostüme. Lösche, nachdem du die Figur in das Programm geladen hast, das blaue Kostüm, das wir im Programm nicht brauchen.



Benenne das gelbe Kostüm in *ein* um, das rote in *aus*. Die Namen stehen ganz oben im Textfeld *Kostüm*, wenn das jeweilige Kostüm ausgewählt ist. Nenne die erste Figur *Licht*, da sie für das Licht im Haus zuständig ist. Der Name der Figur steht auf der Registerkarte *Figuren* rechts oben im Feld *Figuren*.



Die Figur *Licht* braucht zwei Programmblöcke:

Wenn du das Programm mit einem Klick auf das grüne Fähnchen startest, wird das rote Kostüm *aus* aktiviert. Die Variable *licht* erhält die Nummer des aktuellen Kos-

tüms. Dazu wird der Block *Kostüm Zahl* von der Blockpalette *Aussehen* verwendet, der immer die aktuelle Kostümnummer enthält.

Klickst du die Figur an, wird auf das nächste Kostüm gewechselt. Da die Figur nur zwei Kostüme hat, ändern sich diese im Wechsel. Wieder erhält die Variable *licht* die Nummer des aktuellen Kostüms.

Nachdem die Figur und die Programmblöcke fertiggestellt sind, dupliziere die Figur dreimal. Die weiteren Figuren bekommen die Namen *Stern*, *Blinklicht* und *Lauflicht*. Die Kostümnamen bleiben unverändert, nur die Variablen, die von den Figuren geschaltet werden, müssen entsprechend angepasst werden.

DIE VARIABLEN

Dieses Programm verwendet deutlich mehr Variablen als die bisherigen Programme. Fünf Variablen lassen sich über Schieberegler auf der Bühne einstellen.

blinklicht – Blinklicht ein- oder ausgeschaltet. Die Variable wird über die entsprechende Figur geschaltet.

blinklicht zahl – Gibt an, wie oft das Blinklicht komplett durchläuft. Wird über einen Schieberegler auf der Bühne gesteuert.

blinklicht zeit – Zeit in Zehntelsekunden, die eine LED im Blinklicht ein- und ausgeschaltet ist. Wird über einen Schieberegler auf der Bühne gesteuert.

lauflicht – Lauflicht ein- oder ausgeschaltet. Die Variable wird über die entsprechende Figur geschaltet.

lauflicht zahl – Gibt an, wie oft das Lauflicht komplett durchläuft. Wird über einen Schieberegler auf der Bühne gesteuert.

lauflicht zeit – Zeit in Zehntelsekunden, die eine LED im Lauflicht ein- und ausgeschaltet ist. Wird über einen Schieberegler auf der Bühne gesteuert.

led – Pin D13 für die LED im Haus.

licht – Licht im Haus ein- oder ausgeschaltet. Die Variable wird über die entsprechende Figur geschaltet.

n – Zähler für aufeinanderfolgende Pins bei Lauflicht und Blinklicht.

servo – Pin D3 für den Servo.

stern – Stern, der sich über dem Servo dreht, ein- oder ausgeschaltet. Die Variable wird über die entsprechende Figur geschaltet.

stern zeit – Wartezeit zwischen den Bewegungen des Sterns in Sekunden.

taste – Pin D2 für den Taster.

winkel – Drehwinkel für den Servo.

DAS PROGRAMM

Das Programm ist ähnlich aufgebaut wie das von gestern. Es besteht diesmal aus mehreren *wenn ... dann ...*-Abfragen, die die über die Figuren gesteuerten Variablen abfragen und, wenn die jeweilige Figur das gelbe Kostüm zeigt, den entsprechenden Teil der gesamten Szene ablaufen lassen, nachdem der Taster gedrückt wurde.

```

wenn geklickt wird
setze taste auf 2
setze servo auf 3
setze led auf 13
wiederhole fortlaufend
wenn nicht Digitalpin lesen taste , dann
wenn licht = 1 , dann
Digitalpin von Ausgang led als hoch setzen
wenn stern = 1 , dann
setze winkel auf 0
wiederhole 6
set servo pin servo angle as winkel
warte stern zeit Sekunde(n)
ändere winkel um 30
wenn blinklicht = 1 , dann
wiederhole blinklicht zahl
Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 4 und 6 als hoch setzen
Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 7 und 9 als hoch setzen
warte blinklicht zeit / 10 Sekunde(n)
Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 4 und 6 als niedrig setzen
Digitalpin von Ausgang wähle eine zufällige Zahl zwischen 7 und 9 als niedrig setzen
warte blinklicht zeit / 10 Sekunde(n)
setze n auf 4
wiederhole 6
Digitalpin von Ausgang n als niedrig setzen
ändere n um 1
wenn laufflicht = 1 , dann
wiederhole laufflicht zahl
setze n auf 10
wiederhole 3
Digitalpin von Ausgang n als hoch setzen
warte laufflicht zeit / 10 Sekunde(n)
Digitalpin von Ausgang n als niedrig setzen
ändere n um 1
wenn stern = 1 , dann
setze winkel auf 90
set servo pin servo angle as winkel
Digitalpin von Ausgang led als niedrig setzen
warte 0.1 Sekunde(n)

```

SO FUNKTIONIERT DAS PROGRAMM

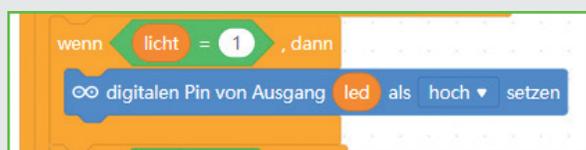


Bei Klick auf das grüne Fähnchen werden die Variablen für die Pinnummern gesetzt. Die anderen Variablen kannst du auf der Bühne mit den Schiebereglern frei einstellen. Alle vier Schalter zeigen das rote Kostüm. Das bedeutet, die jeweiligen Effekte werden bei Drücken des Tasters nicht ausgelöst. Klicke einzelne oder alle Schalter an, erscheinen sie in Gelb. Wenn du dann den Taster drückst, werden die ausgewählten Effekte ausgelöst.

Die Figur *Stern* schaltet die Drehung des Servos ein oder aus. Der Servo dreht sich in sechs Schritten von je 30 Grad. Die Wartezeit zwischen den Drehungen wird über den Schieberegler *stern zeit* eingestellt.

Am Ende der gesamten Szene dreht den Servo wieder in die Grundstellung zurück.

DAS LICHT IM HAUS



Wenn die Figur *Licht* eingeschaltet ist, wird einfach nur die einzelne LED im Haus eingeschaltet.



BLINKENDE RGB-LEDS AUF DEM DACH

Die Figur *Blinklicht* schaltet die Blinkfolge der beiden RGB-LEDs ein oder aus. Die Umschaltung funktioniert genau so wie bei der ersten Figur. Die Blinksequenz wird wie im Programm von gestern gesteuert. Am Ende werden alle sechs Farben der beiden RGB-LEDs an den Pins D4 bis D9 ausgeschaltet.

SERVO MIT STERN



Wie oft diese Schleife wiederholt werden soll, wird über den Schieberegler *blinklicht zahl* eingestellt.



LAUFLICHT AUF DEM DACH



Die Figur *Laufflicht* schaltet das Lauflicht aus drei LEDs an den Pins D10 bis D12 ein oder aus. Die Umschaltung funktioniert analog zu den anderen Figuren. Im Lauflicht wird nacheinander je eine LED ein- und nach einer kurzen Wartezeit, die mit dem Schieberegler *laufflicht zeit* eingestellt wird, wieder ausgeschaltet.

Wie oft diese Schleife wiederholt wird, kann über den Schieberegler *laufflicht zahl* eingestellt werden.



Das Programm ist unter dem Namen 22haus03.mblock bei den Downloads zu finden.

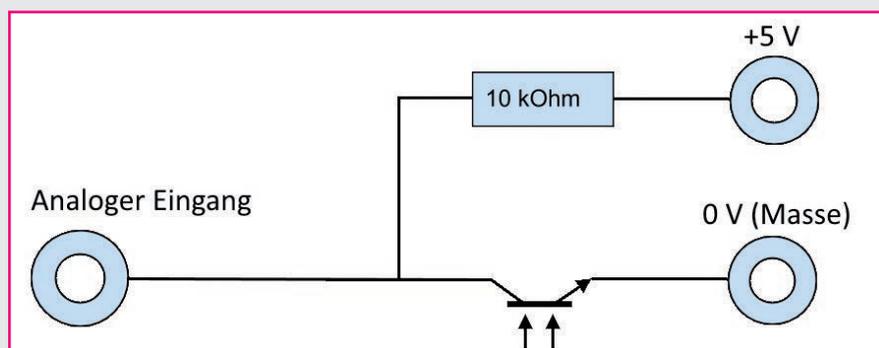
Tag 23

HEUTE IM ADVENTSKALENDER

- 1 Fototransistor PT331C
- 1 Widerstand 10 kOhm (braun, schwarz, orange)

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. Die Schaltung ist so aufgebaut, dass der Wert am analogen Eingang umso geringer wird, je heller das Licht auf den Fototransistor fällt. Anders als bei LEDs wird bei einem Fototransistor der lange Anschluss mit Masse verbunden, nicht der kurze.



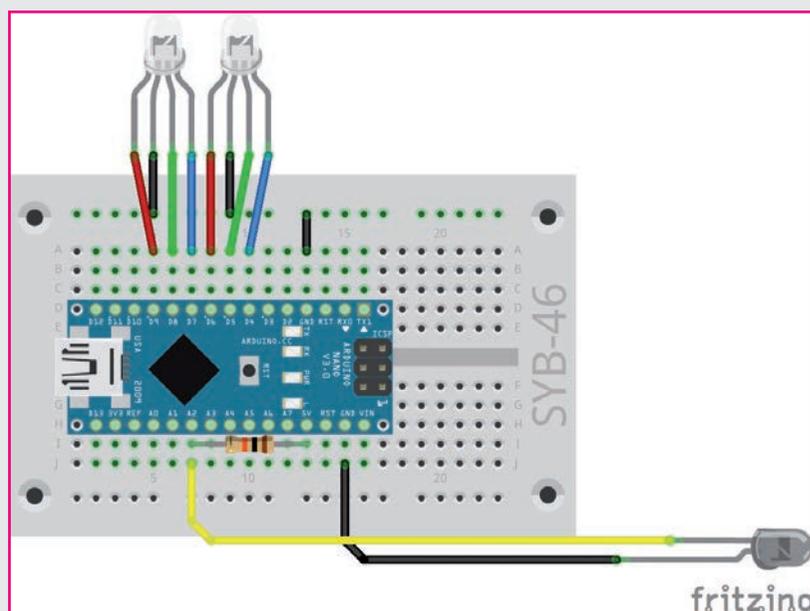
Schaltschema für einen Fototransistor.

AUFGABE

Die beiden RGB-LEDs sollen bunt blinken, wenn es dunkel genug ist.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 2 RGB-LEDs mit Vorwiderständen
- 1 Fototransistor
- 1 Widerstand 10 kOhm (braun, schwarz, orange)
- 5 Verbindungskabelpaare
- 1 Drahtbrücke



WEIHNACHTLICHE LICHTEFFEKTE IN DER DUNKELHEIT

Der Blinkeffekt startet automatisch, wenn es dunkel wird. Dabei lässt sich der Helligkeitswert einstellen, ab dem es dunkel genug ist.

```
wenn  geklickt wird
  setze n auf 4
  wiederhole 6
    digitalen Pin von Ausgang n als niedrig setzen
    ändere n um 1
  wiederhole fortlaufend
    setze helligkeit auf analogen Pin (A) 2 lesen
    wenn helligkeit > schwellwert, dann
      setze n auf wähle eine zufällige Zahl zwischen 4 und 9
      digitalen Pin von Ausgang n als hoch setzen
      warte blinkzeit / 10 Sekunde(n)
      digitalen Pin von Ausgang n als niedrig setzen
      warte blinkzeit / 10 Sekunde(n)
```

SO FUNKTIONIERT DAS PROGRAMM



Bei Klick auf das grüne Fähnchen werden alle sechs Farben der beiden RGB-LEDs ausgeschaltet. Danach startet eine Endlosschleife.

Der Fototransistor liefert je nach Helligkeit einen analogen Wert zwischen 0 und 1023, der umso höher ist, je weniger Licht auf den Fototransistor fällt. Der Block *analogen Pin (A) ... lesen* liest den analogen Wert an Pin A2 und zeigt ihn in der Variablen *helligkeit* auf der Bühne an.

Ist dieser Wert größer als der eingestellte *schwellwert*, ist es dem Programm dunkel genug, um die LEDs blin-

ken zu lassen. Diesen Schwellenwert kannst du entsprechend der Umgebung, in der du die Schaltung verwendest, mit einem Schieberegler ändern.

Diesmal verwenden wir zur Abwechslung ein etwas anderes Blinkmuster. Einer der sechs Pins der beiden RGB-LEDs wird zufällig ausgewählt und blinkt kurz auf. Im nächsten Durchlauf wird wieder ein Pin zufällig gewählt. Die Blinkzeit kann in Zehntelsekunden ebenfalls über einen Schieberegler eingestellt werden.



Das Programm ist unter dem Namen `23blink.mblock` bei den Downloads zu finden.

Tag 24

HEUTE IM ADVENTSKALENDER

- 1 Verbindungskabelpaar
- 1 Downloadcode

BUNTE LICHTEFFEKTE UND WEIHNACHTSLIEDER

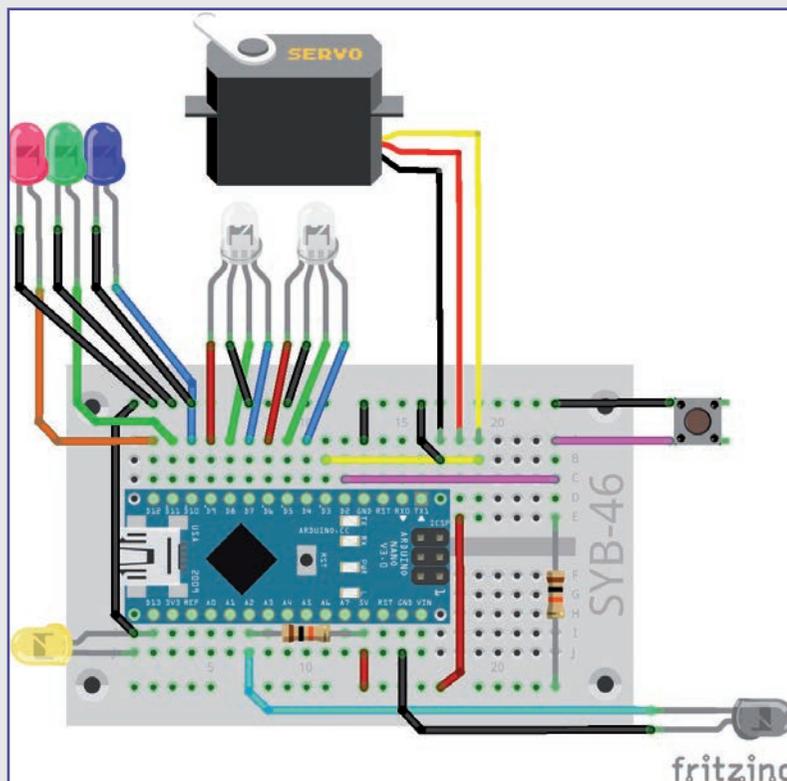
Das Programm des 24. Tages schaltet beim Drücken des Tasters das Licht im Haus an, dreht den Stern und lässt das Lauflicht laufen. Zusätzlich ertönt aus dem PC-Lautsprecher ein Weihnachtslied. Bei Dunkelheit blinken außerdem die LEDs auf dem Dach.

 Über den Downloadcode im Adventskalender heute kannst du dir noch weitere Weihnachtslieder im MP3-Format herunterladen.

Die Schaltung bringt alle aus den letzten Tagen bekannten Elemente zusammen.

Bauteile

- 1 Nano-Board
- 1 Steckbrett
- 1 Servo
- 1 Pinleiste 3 polig
- 2 RGB-LEDs mit Vorwiderständen
- 1 LED rot mit Vorwiderstand
- 1 LED gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 1 LED blau mit Vorwiderstand
- 1 Taster
- 1 Fototransistor
- 2 Widerstände 10 kOhm (braun, schwarz, orange)
- 9 Verbindungskabelpaare
- 7 Drahtbrücken (unterschiedliche Längen)



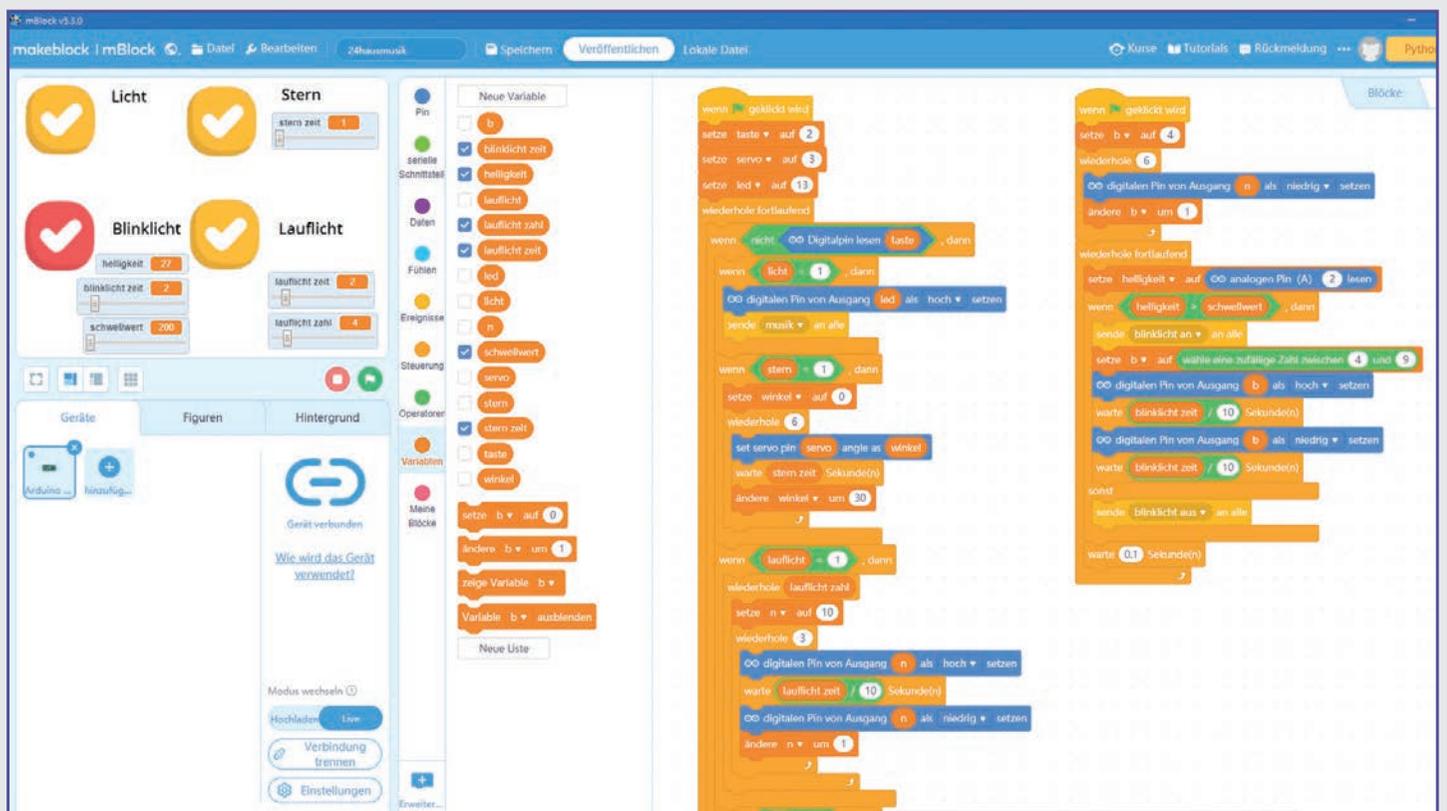
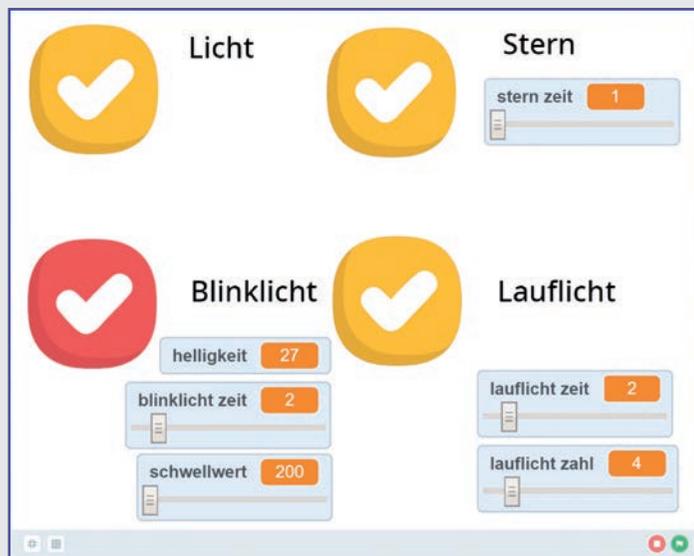
DAS PROGRAMM

Das Programm zeigt wieder ein Steuerpult auf der Bühne. Am besten verwendest du auch hier den Präsentationsmodus.

Die Figuren *Licht*, *Stern* und *Laufflicht* funktionieren wie im Programm des 22. Tages.

Die Figur *Blinklicht* kann nicht angeklickt werden. Das Blinklicht wird wieder über den Fototransistor gesteu-

ert. Die Variable *schwellewert* legt den Schwellenwert fest, ab dem es dunkel genug ist, um den Blinkeffekt zu zeigen, und kann über einen Schieberegler eingestellt werden. Die Variable *helligkeit* ist nicht einstellbar. Sie zeigt, während das Programm läuft, auf der Bühne die aktuell gemessene Lichtstärke auf dem Fototransistor an. Damit kannst du den Wert *Helligkeit* dem Umgebungslicht entsprechend einstellen.



DIE VARIABLEN

Dieses Programm nutzt ähnliche Variablen wie das Programm des 22. Tages. Wieder lassen sich fünf Variablen über Schieberegler auf der Bühne einstellen.

b – Zähler für aufeinanderfolgende Pins beim Blinklicht.

blinklicht zeit – Zeit in Zehntelsekunden, die eine LED im Blinklicht ein- und ausgeschaltet ist. Wird über einen Schieberegler auf der Bühne gesteuert.

helligkeit – Aktuelle Helligkeit, vom Fototransistor gemessen.

lauflicht – Lauflicht ein- oder ausgeschaltet. Die Variable wird über die entsprechende Figur geschaltet.

lauflicht zahl – Gibt an, wie oft das Lauflicht komplett durchläuft. Wird über einen Schieberegler auf der Bühne gesteuert.

lauflicht zeit – Zeit in Zehntelsekunden, die eine LED im Lauflicht ein- und ausgeschaltet ist. Wird über einen Schieberegler auf der Bühne gesteuert.

led – Pin D13 für die LED im Haus.

licht – Licht im Haus ein- oder ausgeschaltet. Die Variable wird über die entsprechende Figur geschaltet.

n – Zähler für aufeinanderfolgende Pins beim Lauflicht.

schwellwert – Schwellenwert für die Helligkeit zum Einschalten des Blinklichts.

servo – Pin D3 für den Servo.

stern – Stern, der sich über dem Servo dreht, ein- oder ausgeschaltet. Die Variable wird über die entsprechende Figur geschaltet.

stern zeit – Wartezeit zwischen den Bewegungen des Sterns in Sekunden.

taste – Pin D2 für den Taster.

winkel – Drehwinkel für den Servo.

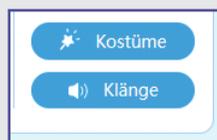
SO FUNKTIONIERT DAS PROGRAMM

Die Programmblöcke für das Licht im Haus, das Lauflicht und den Servo zum Drehen des Sterns wurden aus dem Programm vom 22. Tag übernommen.

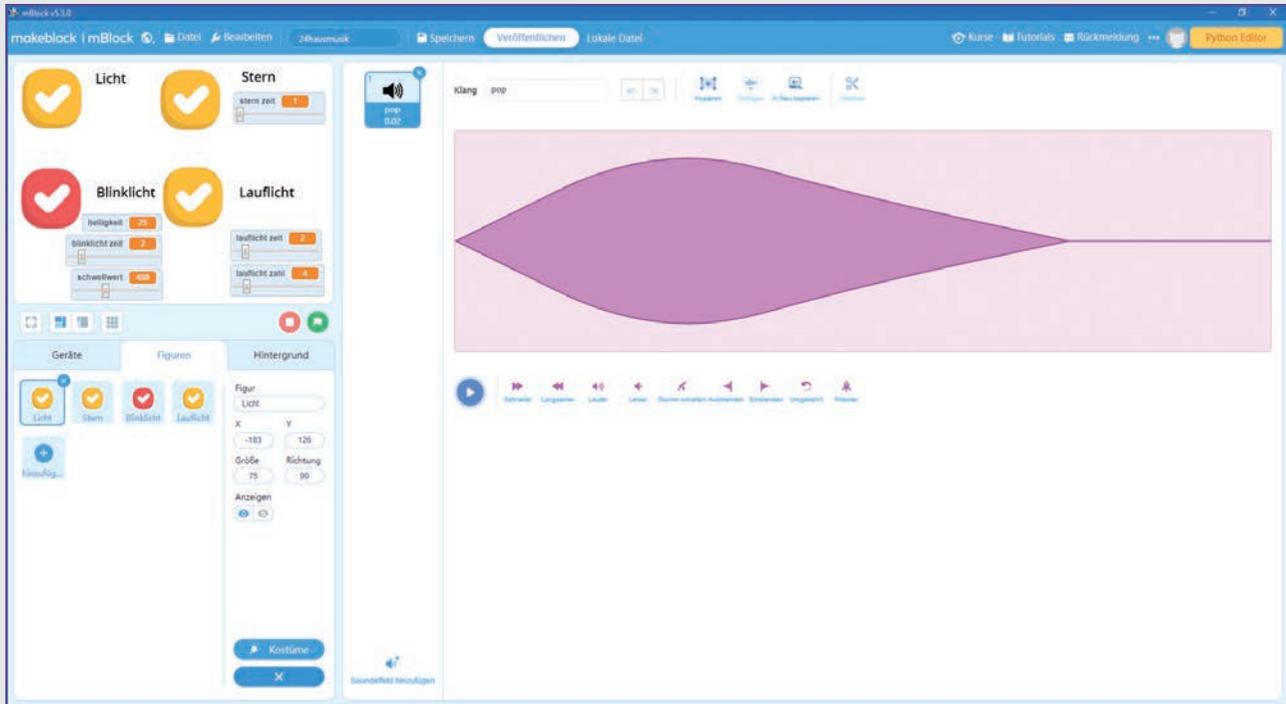
```
wenn geklickt wird
  setze taste auf 2
  setze servo auf 3
  setze led auf 13
  wiederhole fortlaufend
    wenn nicht Digitalpin lesen taste, dann
      wenn licht = 1, dann
        Digitalpin von Ausgang led als hoch setzen
        sende musik an alle
      wenn stern = 1, dann
        setze winkel auf 0
        wiederhole 6
          set servo pin servo angle as winkel
          warte stern zeit Sekunde(n)
          ändere winkel um 30
        wenn laufflicht = 1, dann
          wiederhole laufflicht zahl
            setze n auf 10
            wiederhole 3
              Digitalpin von Ausgang n als hoch setzen
              warte laufflicht zeit / 10 Sekunde(n)
              Digitalpin von Ausgang n als niedrig setzen
              ändere n um 1
            wenn stern = 1, dann
              setze winkel auf 90
              set servo pin servo angle as winkel
              Digitalpin von Ausgang led als niedrig setzen
          warte 0.1 Sekunde(n)
```

Das Programm soll zusätzlich ein Weihnachtslied aus dem PC-Lautsprecher erklingen lassen. mBlock 5 verwendet dazu Blöcke von der Blockpalette *Klang*, die nur bei Figuren zur Verfügung stehen. Das Weihnachtslied soll immer dann abgespielt werden, wenn das Licht im Haus eingeschaltet ist. Deshalb verwenden wir auch die Figur *Licht*, um das Lied abzuspielen. Das Hauptprogramm des Geräts sendet, nachdem die LED eingeschaltet wurde, eine Nachricht *musik*, die die Figur dann auswertet.

Um das Weihnachtslied in das Programm einzubauen, wähle auf der Registerkarte *Figuren* die Figur *Licht* und klicke unten auf das Symbol *Klänge*.



Im Bereich *Klänge* ist der Figur bereits ein vordefinierter Klang *pop* zugeordnet. Lösche diesen Klang mit dem blauen x-Symbol.

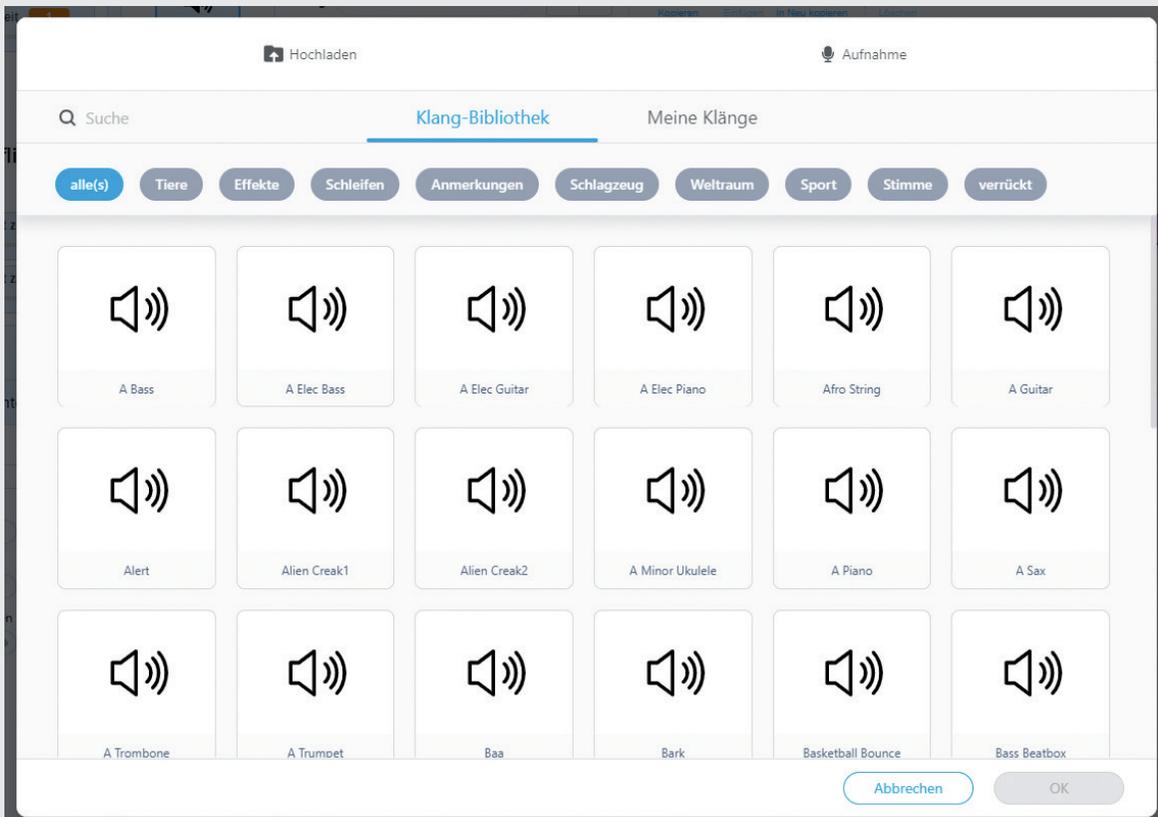


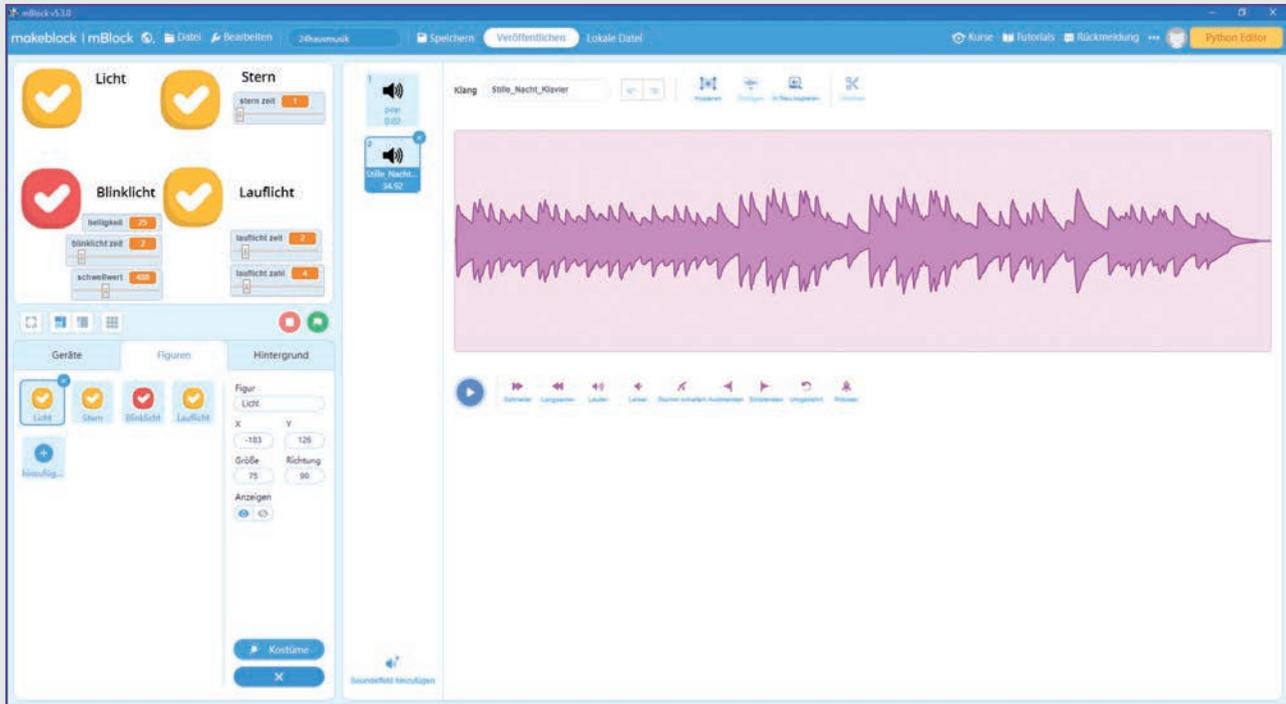
Klicke unten auf *Soundeffekt hinzufügen*.



Jetzt erscheint die in mBlock 5 mitgelieferte Klangbibliothek mit einer vielfältigen Auswahl an Klängen. Klicke oben auf *Hochladen*, um eine eigene MP3- oder WAV-Datei hochzuladen.

Lade das Lied *Stille Nacht* aus den Weihnachtsliedern, die du über den Downloadcode im heutigen Türchen des Adventskalenders bekommst, in das Programm.





Jetzt kannst du das Lied anhören oder bearbeiten, zum Beispiel kannst du ein Stück abschneiden oder das Lied schneller oder langsamer, lauter oder leiser abspielen lassen.

Das Programm von heute kombiniert das Programm für den Fototransistor und das Blinklicht vom 23. Tag mit Steuerpult, Servosteuerung und Lauflicht aus früheren Tagen.



Verlasse den Bereich *Klänge* mit einem Klick auf das x-Symbol unten.

Eine Endlosschleife wartet darauf, dass der vom Fototransistor am analogen Eingang A2 gelieferte Wert *helligkeit* größer ist als der eingestellte Schwellenwert. Dann startet das Blinklicht auf den RGB-LEDs.

Die Figur *Licht* bekommt jetzt den weiteren Programmblock *Wenn ich musik empfange*. Er ruft einen Block *spiele Sound ...* von der Blockpalette *Klang* auf. Wähle hier im Listenfeld das hochgeladene Weihnachtslied aus.



```

wenn  geklickt wird
setze b  auf 4
wiederhole 6
   digitalen Pin von Ausgang n als niedrig  setzen
  ändere b  um 1
wiederhole fortlaufend
  setze helligkeit  auf  analogen Pin (A) 2 lesen
  wenn  helligkeit > schwellwert , dann
    sende blinklicht an  an alle
    setze b  auf wähle eine zufällige Zahl zwischen 4 und 9
     digitalen Pin von Ausgang b als hoch  setzen
    warte blinklicht zeit / 10 Sekunde(n)
     digitalen Pin von Ausgang b als niedrig  setzen
    warte blinklicht zeit / 10 Sekunde(n)
  sonst
    sende blinklicht aus  an alle
  warte 0.1 Sekunde(n)

```

Die Figur *Blinklicht* soll über ihr Kostüm auf der Bühne anzeigen, ob das Blinklicht gerade läuft oder nicht. Die *wenn ... dann ... sonst*-Abfrage sendet dazu eine Nachricht *blinklicht an* oder *blinklicht aus*.

Die Figur *Blinklicht* wertet diese Nachricht aus und wechselt zu dem entsprechenden Kostüm. Das gelbe Kostüm *ein* bedeutet, das Blinklicht ist eingeschaltet, das rote Kostüm *aus* zeigt an, dass das Blinklicht ausgeschaltet ist.

```

wenn ich blinklicht an  empfange
  wechse Objekt zu ein 
wenn ich blinklicht aus  empfange
  wechse Objekt zu aus 

```

Starte das Programm mit einem Klick auf das grüne Fähnchen. Schalte dann die gewünschten Figuren ein und drücke den Taster. Wenn es dunkel genug ist, blinken zusätzlich noch die RGB-LEDs.

Der Downloadcode heute im Adventskalender liefert einige weitere Weihnachtslieder im MP3-Format. Damit kannst du das im Programm gespeicherte Lied nach

deinem persönlichen Geschmack austauschen oder das Programm um weitere Lieder ergänzen.



Das Programm ist unter dem Namen 24hausmusik.mblock bei den Downloads zu finden.

Frohe Weihnachten!

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 und somit heute urheberrechtsfrei genutzt werden kann.

