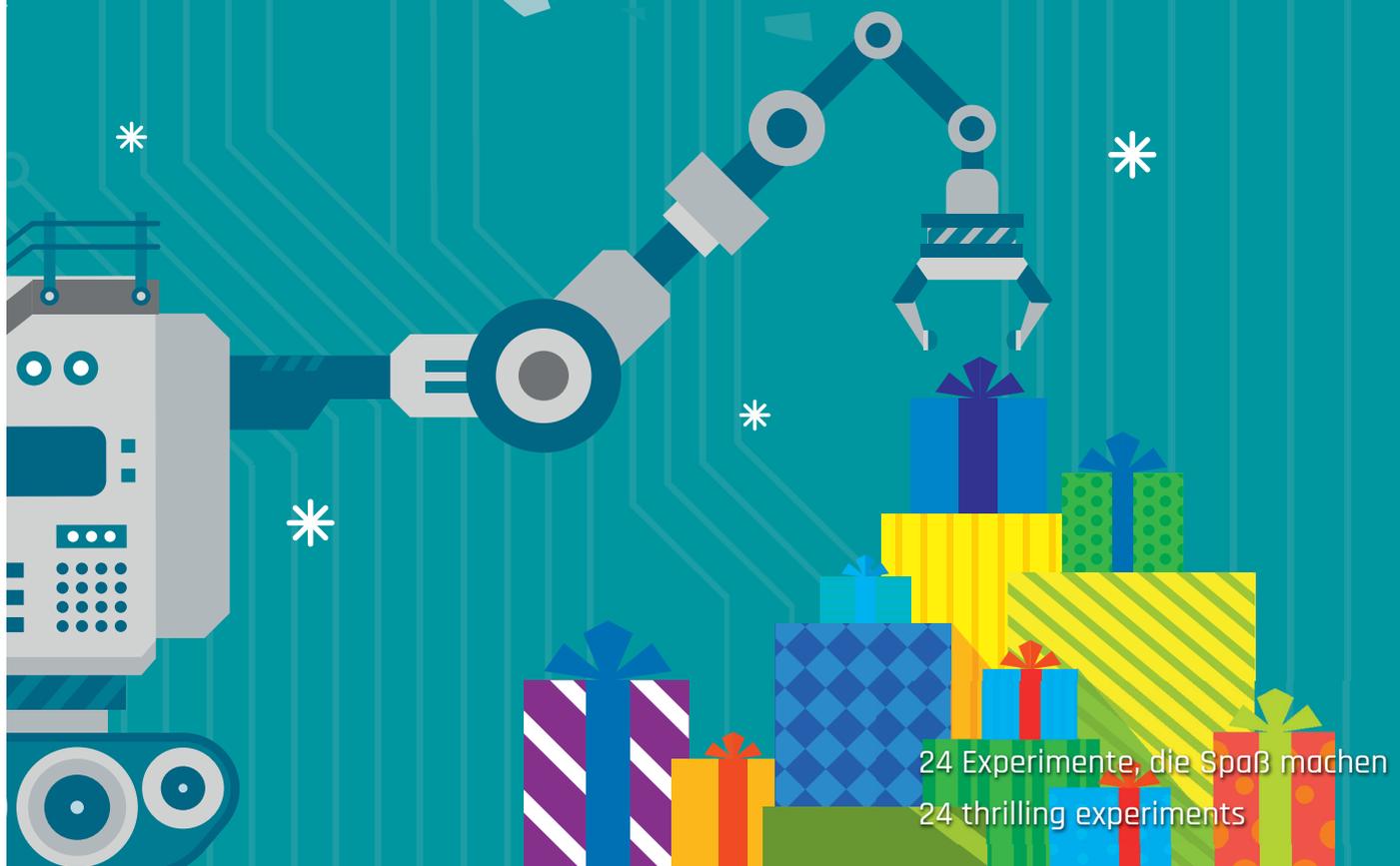




## ADVENTSKALENDER FÜR ARDUINO®

## ADVENT CALENDAR FOR ARDUINO®



## Alle Versuche im Überblick

<b>Arduino®-Adventskalender 2019</b> .....	<b>3</b>	<b>12. Tag</b> .....	<b>17</b>
<b>1. Tag</b> .....	<b>3</b>	Heute im Adventskalender .....	17
Heute im Adventskalender .....	3	Lauflicht in zwei Richtungen mit einstellbarer Geschwindigkeit .....	17
Nano-Board – Arduino-kompatible Platine .....	3	Das Programm .....	17
Nano vorbereiten .....	3	So funktioniert das Programm .....	17
Softwareinstallation in Kürze .....	4	<b>13. Tag</b> .....	<b>17</b>
mBlock 3 .....	4	Heute im Adventskalender .....	17
LED blinkt .....	5	Pinbelegung eines HD44780-kompatiblen LCD-Moduls .....	18
So funktioniert das Programm .....	5	LCD-Modul mit mBlock 3 steuern .....	18
Programm auf den Nano übertragen .....	6	Das Programm .....	19
<b>2. Tag</b> .....	<b>6</b>	So funktioniert das Programm .....	19
Heute im Adventskalender .....	6	<b>14. Tag</b> .....	<b>20</b>
Steckbrett .....	6	Heute im Adventskalender .....	20
LEDs .....	6	Laufschrift auf dem LCD-Modul .....	20
Wechselblinklicht .....	6	Das Programm .....	20
Das Programm .....	7	So funktioniert das Programm .....	20
So funktioniert das Programm .....	7	<b>15. Tag</b> .....	<b>20</b>
<b>3. Tag</b> .....	<b>7</b>	Heute im Adventskalender .....	20
Heute im Adventskalender .....	7	Batteriekasten .....	20
Verbindungskabel .....	7	Stoppuhr .....	20
LEDs blinken mit einstellbarer Geschwindigkeit .....	7	Das Programm .....	21
Das Programm .....	7	So funktioniert das Programm .....	21
So funktioniert das Programm .....	8	<b>16. Tag</b> .....	<b>22</b>
<b>4. Tag</b> .....	<b>8</b>	Heute im Adventskalender .....	22
Heute im Adventskalender .....	8	Weihnachtsbotschaft mit Schreibmaschineneffekt .....	22
Ampelschaltung .....	8	Das Programm .....	22
Das Programm .....	8	So funktioniert das Programm .....	22
So funktioniert das Programm .....	8	<b>17. Tag</b> .....	<b>22</b>
<b>5. Tag</b> .....	<b>9</b>	Heute im Adventskalender .....	22
Heute im Adventskalender .....	9	Analoge Eingänge auslesen .....	22
Schaltdraht .....	9	Das Programm .....	22
LEDs dimmen .....	9	So funktioniert das Programm .....	23
Das Programm .....	9	<b>18. Tag</b> .....	<b>23</b>
So funktioniert das Programm .....	9	Heute im Adventskalender .....	23
<b>6. Tag</b> .....	<b>10</b>	Sensorkontakt aus Knete .....	23
Heute im Adventskalender .....	10	Das Programm .....	24
Potentiometer .....	10	So funktioniert das Programm .....	24
LED mit Potentiometer dimmen .....	10	<b>19. Tag</b> .....	<b>24</b>
Das Programm .....	10	Heute im Adventskalender .....	24
So funktioniert das Programm .....	10	Zahlen raten .....	24
<b>7. Tag</b> .....	<b>11</b>	Das Programm .....	24
Heute im Adventskalender .....	11	So funktioniert das Programm .....	25
Lauflicht mit Potentiometer steuern .....	11	<b>20. Tag</b> .....	<b>26</b>
Das Programm .....	11	Heute im Adventskalender .....	26
So funktioniert das Programm .....	11	Pong-Spiel auf dem PC mit Tasten steuern .....	26
<b>8. Tag</b> .....	<b>11</b>	Das Programm .....	26
Heute im Adventskalender .....	11	So funktioniert das Spiel .....	26
LEDs blinken zufällig .....	11	Pong mit Tastern am Nano steuern .....	28
Das Programm .....	11	<b>21. Tag</b> .....	<b>28</b>
So funktioniert das Programm .....	12	Heute im Adventskalender .....	28
<b>9. Tag</b> .....	<b>12</b>	Das Programm .....	28
Heute im Adventskalender .....	12	So funktioniert das Programm .....	29
Analoge Pegelanzeige mit LEDs .....	12	<b>22. Tag</b> .....	<b>32</b>
Das Programm .....	12	Heute im Adventskalender .....	32
So funktioniert das Programm .....	12	Reaktionsspiel .....	32
<b>10. Tag</b> .....	<b>13</b>	Das Programm .....	32
Heute im Adventskalender .....	13	So funktioniert das Programm .....	33
Nano-Board über mBlock 3 interaktiv steuern .....	13	<b>23. Tag</b> .....	<b>33</b>
Echtzeitverbindung mit dem Nano einrichten .....	13	Heute im Adventskalender .....	33
Das Programm .....	14	Fototransistor .....	33
So funktioniert das Programm .....	14	Weihnachtliche Lichteffekte in der Dunkelheit .....	33
Das Bühnenbild .....	14	Das Programm .....	33
<b>11. Tag</b> .....	<b>15</b>	So funktioniert das Programm .....	33
Heute im Adventskalender .....	15	<b>24. Tag</b> .....	<b>34</b>
Widerstände und ihre Farbcodes .....	15	Heute im Adventskalender .....	34
Taster .....	16	Weihnachtsjukebox .....	34
Ampel mit Fußgängerblinklicht mit Taster schalten .....	16	Das Programm .....	34
Das Programm .....	16	So funktioniert das Programm .....	34
So funktioniert das Programm .....	16		

## Arduino®-Adventskalender 2019

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

### Der Name Arduino

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

### Vorsichtsmaßnahmen

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

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

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

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

## 1. Tag

### Heute im Adventskalender

- Nano-Board (Arduino-kompatible Platine)

### Nano-Board - Arduino-kompatible Platine

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

Die Experimente in diesem Adventskalender werden mit mBlock 3 programmiert. Diese Programmiersprache basiert auf Scratch, einer der am leichtesten erlernbaren Programmiersprachen überhaupt. Die verwendeten Programme gibt es hier zum Download: [bit.ly/c-adventskalender-arduino-19](https://bit.ly/c-adventskalender-arduino-19). Entpacken Sie das Zip-Archiv in ein Verzeichnis auf Ihrer Festplatte.

### Nano vorbereiten

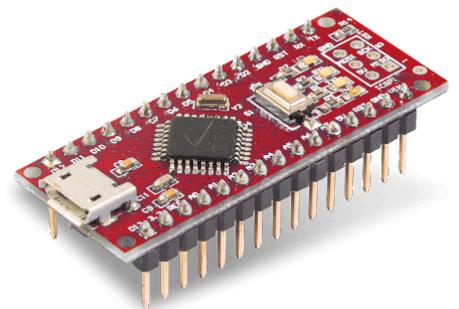
Um den Nano in Betrieb zu nehmen, braucht man:

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

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

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

1. Tag



Das Arduino-kompatible Nano-Board.



Installation des Gerätetreibers für das Nano-Board.

### Softwareinstallation in Kürze

Die Treiberinstallation in vier Schritten:

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



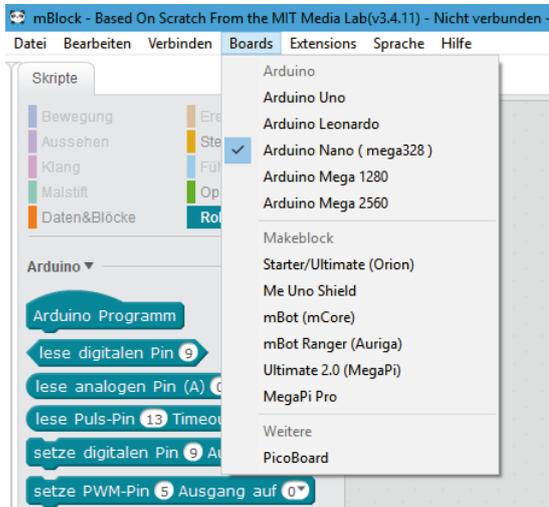
Firewallzugriff zulassen.

### mBlock 3

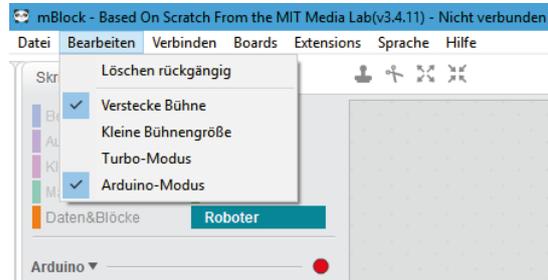
Für die Projekte im Adventskalender verwenden wir die einfach zu erlernende Programmiersprache **mBlock 3**. Laden Sie sich diese bei [www.mblock.cc/](http://www.mblock.cc/) **mblock-software** herunter und installieren Sie sie.

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

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



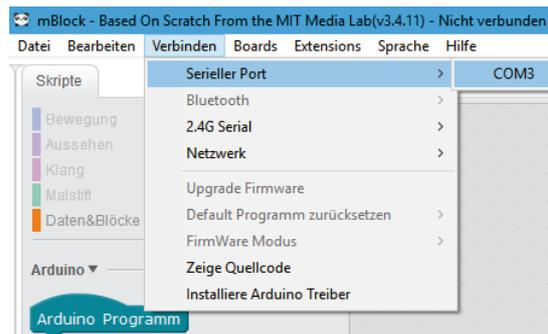
Nano-Board auswählen.



Arduino-Modus einschalten.

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

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



Schnittstelle für die Verbindung auswählen.

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

## LED blinkt

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

### Die Programme zum Adventskalender

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

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

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

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

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



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

### Die LED am Pin 13

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

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

### Den Nano ausschalten

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

### So funktioniert das Programm

Die Schleife sorgt dafür, dass die darin enthaltenen Blöcke zehnmal hintereinander ausgeführt werden.

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

### Dezimalpunkt statt Komma

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

```

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

```

```

avrduide: input file C:\Users\chris\AppData\Local\Temp\build1243307170804926706.tmp/project_01nano4
avrduide: reading on-chip flash data:
Reading | ##### | 100% 0.71s
avrduide: verifying ...
avrduide: 3116 bytes of flash verified
avrduide done. Thank you.

```

send encode mode      recv encode mode  
 Binär-Modus    Zeichen-Modus       Binär-Modus    Zeichen-Modus

Senden

Programm auf das Nano-Board übertragen.



Meldung beim Start der Übertragung.

## Programm auf den Nano übertragen

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

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

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

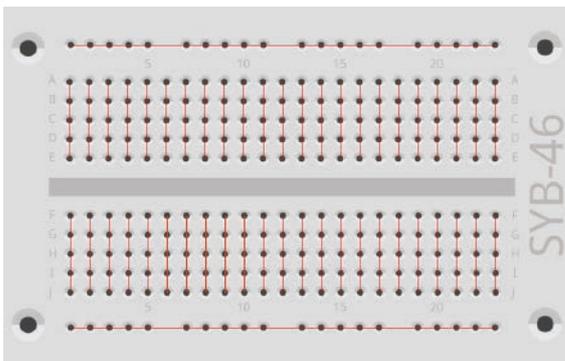
Das Programm startet nach der Übertragung auf den Nano automatisch. Außerdem können Sie es durch einen Druck auf die Taste auf dem Nano jederzeit wieder starten.



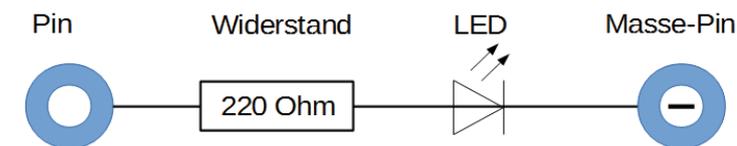
## 2. Tag

### Heute im Adventskalender

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



Die Verbindungen auf dem Steckbrett.



Schaltplan einer LED mit Vorwiderstand.

### Steckbrett

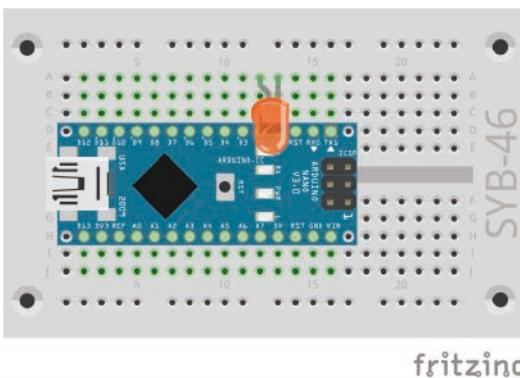
Für den schnellen Aufbau elektronischer Schaltungen, ohne löten zu müssen, ist ein Steckbrett im Adventskalender. Hier können elektronische Bauteile direkt in ein Lochraster gesteckt werden.

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

### LEDs

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

Die LEDs im Adventskalender haben den Vorwiderstand bereits eingebaut und können daher direkt an die Pins angeschlossen werden.



LED-Wechselblinklicht am Nano.

### LED in welcher Richtung anschließen?

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

### Wechselblinklicht

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

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

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

#### Die Pins auf dem Nano

Alle Pins mit **D...** sind digitale Ein- oder Ausgänge, die die Werte **Wahr** oder **Falsch** (Ein oder Aus) annehmen können. Die Pins mit **A...** sind analoge Eingänge. **GND**-Pins sind Masseleitungen. Arduino-kompatible Platinen arbeiten mit unterschiedlichen Spannungen und haben dazu standardmäßig zwei verschiedene Plus-Pins. Am Pin **3V3** liegen +3,3 V Spannung an. Am Pin **5V** liegen +5 V Spannung an. Der Nano im Adventskalender benötigt für ein logisches **Wahr**-Signal +5 V, manche anderen Platinen nur +3,3 V.

#### Das Programm

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

#### So funktioniert das Programm

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

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

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



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

## 3. Tag

#### Heute im Adventskalender

- 1x LED grün mit Vorwiderstand
- 1x Verbindungskabel

#### Verbindungskabel

Heute ist ein Verbindungskabel im Adventskalender. Diese Leitungen werden verwendet, um Kontaktleisten auf dem Steckbrett miteinander zu verbinden. Die Verbindungskabel haben an beiden Enden kleine Drahtstecker, mit denen sie sich leicht in das Steckbrett stecken lassen.

#### LEDs blinken mit einstellbarer Geschwindigkeit

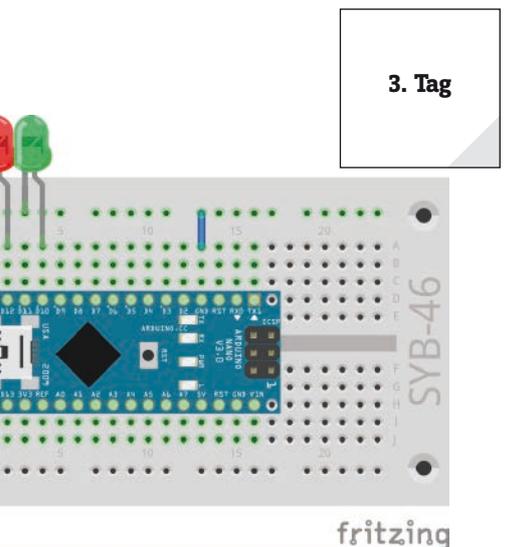
Das Experiment des dritten Tages lässt wieder zwei LEDs abwechselnd blinken. Allerdings können Sie die Geschwindigkeit mit Hilfe einer Variablen im Programm einstellen.

**Bauteile:** 1x Nano-Board, 1x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x Verbindungskabel

Die Schaltung von heute zeigt den typischen Schaltungsaufbau auf dem Steckbrett. Eine der horizontalen Kontaktleisten wird als Masseleitung verwendet, die mit dem GND-Pin auf dem Nano über ein Verbindungskabel verbunden ist. Achten Sie beim Aufbau der Schaltung darauf, dass die Kathoden (kurzer Draht) der LEDs in der Masseleiste stecken. Die Anoden (langer Draht) sind in dieser Schaltung mit den Pins **D10** und **D12** auf dem Nano verbunden.

#### Das Programm

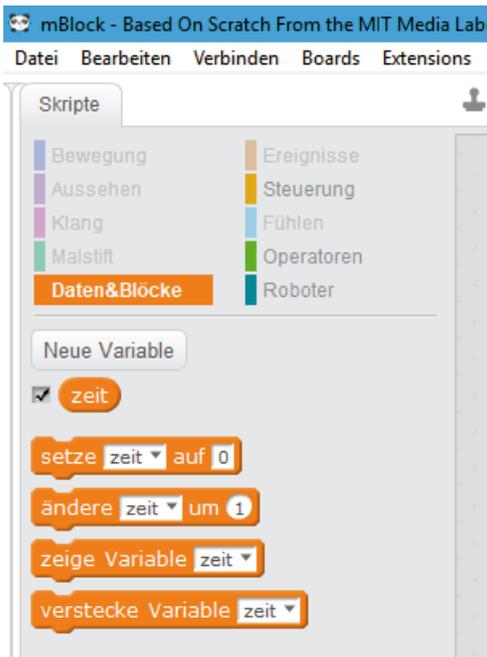
Das Programm `03mb1ock` funktioniert ähnlich wie das von gestern und lässt die diesmal am Pin **D10** und am Pin **D12** angeschlossenen LEDs abwechselnd blinken. Die Blinkfrequenz lässt sich über eine Variable **zeit** festlegen.



LED-Wechselblinklicht am Nano.



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



Blockpalette Daten & Blöcke mit einer Variablen.

### So funktioniert das Programm

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

#### Variablen in mBlock

Variablen sind kleine Speicherplätze, in denen man sich während eines Programms eine Zahl oder irgendetwas anders merken kann. Wenn das Programm beendet wird, werden diese Variablenspeicher automatisch wieder geleert. Variablen müssen in mBlock 3 erst einmal auf der Blockpalette **Daten & Blöcke** mit dem Button **Neue Variable** angelegt werden, bevor man sie benutzen kann. Anschließend können Sie das Symbol der neu angelegten Variablen aus der Blockpalette in ein dafür vorgesehenes Feld eines Blocks im Programm ziehen. Auf der Blockpalette stehen zusätzlich verschiedene Blöcke zum Auslesen und Verändern der Variablen zur Verfügung.

Wurde eine Variable angelegt, erscheint sie als orangefarbenes Symbol auf der Blockpalette.

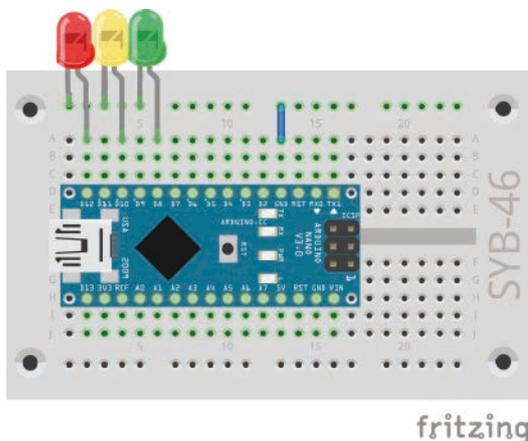
Im Programm wird der eingestellte Wert der Variablen **zeit** für beide Wartezeiten verwendet. Geben Sie im Block **setze zeit auf 0.2** am Anfang des Programms einen anderen Wert ein, blinken die LEDs entsprechend schneller oder langsamer. Danach muss das Programm erneut auf den Nano übertragen werden.



## 4. Tag

### Heute im Adventskalender

- 1x LED gelb mit Vorwiderstand



Ampelschaltung auf dem Steckbrett.

### Ampelschaltung

Eine LED ein- und wieder auszuschalten mag im ersten Moment ganz spannend sein, aber dafür braucht man eigentlich keinen Mikrocontroller. Eine Verkehrsampel mit ihrem typischen Leuchtzyklus von Grün über Gelb nach Rot und dann über eine Lichtkombination Rot-Gelb wieder zu Grün zeigt weitere Programmieretechniken.

**Bauteile:** 1x Nano-Board, 1x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x Verbindungskabel

### Das Programm

Das Programm `04mb1ock` funktioniert ähnlich wie das von gestern. Wieder werden in einer Endlosschleife abwechselnd verschiedene digitale Pins ein- und ausgeschaltet. Der Übersichtlichkeit halber werden diesmal die Pins der LEDs mit Variablen bezeichnet, was auch den Vorteil hat, dass man die Schaltung leichter auf andere Pins umbauen kann.

### So funktioniert das Programm

Zum Start werden die drei Variablen auf die für die LEDs verwendeten Pin-Nummern gesetzt. Danach werden noch vor Beginn der Hauptschleife die LEDs auf einen definierten Zustand gesetzt. Nur die grüne LED soll leuchten.

Bei der Verwendung von Variablen kann mBlock 3 nicht gleich am Anfang erkennen, welche Pins als Ausgänge verwendet werden sollen, und initialisiert sie erst bei ihrer ersten Verwendung. Daher kann man bei Programmstart von keinem eindeutig definierten Zustand der LEDs ausgehen. Um dies zu vermeiden, schaltet das Script am Anfang alle LEDs aus und initialisiert sie damit.

Es startet eine Endlosschleife, die in jedem Durchlauf den typischen Ampelzyklus laufen lässt. Dabei sind die Phasen Gelb und Rot/Gelb nur je 0,6 Sekunden lang, wohingegen die Ampel je 2 Sekunden Rot und Grün leuchtet.

## 5. Tag

### Heute im Adventskalender

- Schaltdraht (isoliert)

5. Tag

### Schaltdraht

Heute ist Schaltdraht im Adventskalender enthalten. Damit stellen Sie kurze Verbindungsbrücken her, mit denen dicht nebeneinander liegende Kontaktreihen auf der Steckplatine verbunden werden. Solche Drahtbrücken liegen flach auf dem Steckbrett und stören beim Aufbau der anderen Bauteile weniger als die langen Verbindungskabel.

Schneiden Sie den Draht mit einem kleinen Seitenschneider je nach Experiment auf die passenden Längen ab. Um die Drähte besser in die Steckplatine stecken zu können, empfiehlt es sich, sie leicht schräg abzuschneiden, sodass eine Art Keil entsteht. Entfernen Sie an beiden Enden auf einer Länge von etwa einem halben Zentimeter die Isolierung.

### LEDs dimmen

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



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

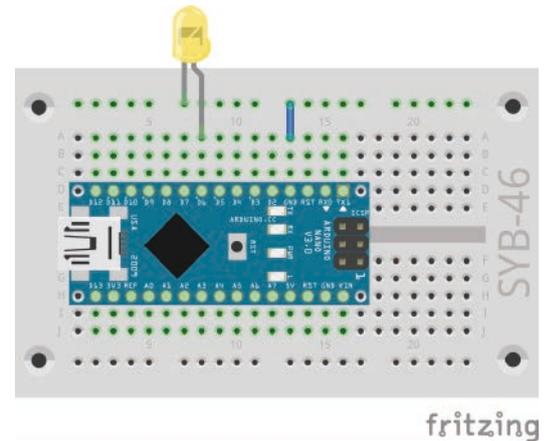
Mit einem Trick ist es dennoch möglich, die Helligkeit einer LED an einem digitalen Pin zu regeln. Lässt man eine LED schnell genug blinken, nimmt das menschliche Auge das nicht mehr als Blinken wahr. Die als Pulsweitenmodulation (PWM) bezeichnete Technik erzeugt ein pulsierendes Signal, das sich in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

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

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

### Pins für PWM-Signale

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



Eine LED wird gedimmt.

### Das Programm

Das Programm `05mblock` dimmt die LED am Pin **D6** zyklisch heller und dunkler. Dabei werden einige neue Programmblöcke verwendet.

### So funktioniert das Programm

Am Anfang werden drei Variablen definiert: **zeit** legt die Geschwindigkeit beim Dimmen fest, **hell** bezeichnet den PWM-Wert für die Helligkeit der LED und **schritt** gibt die Schrittweite beim Dimmen an.

Jetzt beginnt eine Endlosschleife. Als Erstes wird bei jedem Schleifendurchlauf der aktuelle Wert der Variable **hell** als PWM-Wert auf den Pin **D6** ausgegeben. Auf der Blockpalette **Roboter** gibt es einen eigenen Block für PWM-Pins. Anschließend wird der Wert der Variable **hell** um den Wert **schritt** erhöht.

Im nächsten Schritt wird mit einem **falls...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 **falls**-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 Variable **hell** den Wert 0 oder 250 erreicht hat. Trifft dies zu, wird die Variable **schritt** auf einen neuen Wert gesetzt. Da mBlock 3 keine Möglichkeit bietet, das

Vorzeichen einer Variable umzukehren, verwenden wir den Operator '-' und subtrahieren den Wert der Variable von 0, was das gleiche Ergebnis liefert.

Zum Schluss wartet das Programm in jedem Schleifendurchlauf die in der Variablen **zeit** gespeicherten 0,2 Sekunden lang. Anschließend startet die Endlosschleife neu und liefert der LED einen neuen PWM-Wert.

## 6. Tag

## 6. Tag

### Heute im Adventskalender

- 15-kOhm-Potentiometer
- 1x Verbindungskabel

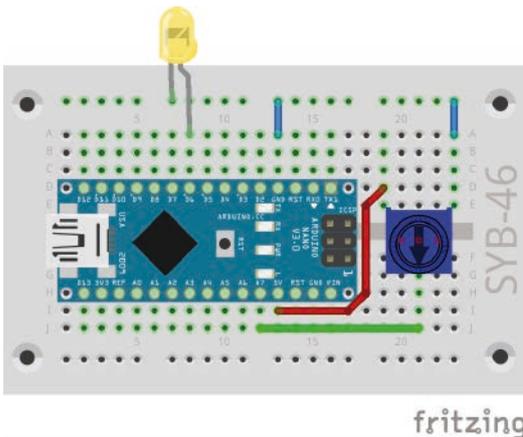
### Potentiometer

Das Potentiometer aus dem Adventskalender von heute ist ein einstellbarer Widerstand, der Werte zwischen 0 Ohm und 15 kOhm annehmen kann, indem man den Knopf dreht.

### LED mit Potentiometer dimmen

Um eine LED wirklich wie eine Wohnzimmerlampe zu dimmen, verwenden wir das Potentiometer aus dem Adventskalender. Mit diesem einstellbaren Widerstand lässt sich ein Spannungsteiler bauen, der eine beliebige Spannung zwischen 0 V und +5 V liefern kann. Diese analoge Spannung kann aber nicht direkt an die LED weitergegeben werden, sondern muss in einen digitalen PWM-Wert umgerechnet werden.

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



LED-Dimmer mit Potentiometer.

Die beiden kurzen Leitungen sind Drahtbrücken. Für die beiden längeren Leitungen können Sie die beiden Verbindungskabel nutzen oder Drahtbrücken zurechtschneiden.

Das Nano-Board verfügt über acht analoge Eingangspins **A0...A7**, die eine anliegende Spannung in einen Zahlenwert umrechnen.

### Das Programm

Das Programm `06mb1ock` dimmt die LED mit dem Potentiometer. Dazu wird es am analogen Eingang **A7** angeschlossen.

Die analogen Eingänge des Nano werten einen analogen Spannungswert aus und liefern digitale Werte zwischen 0 und 1023. Dabei steht 0 für 0 V und 1023 für +5 V Spannung am jeweiligen Pin. Diese Werte müssen auf Werte zwischen 0 und 255 umgerechnet werden, um sie dann wieder als PWM-Wert an eine LED auszugeben.



Das Programm `06mb1ock` dimmt die LED mit einem Potentiometer.

Nicht angeschlossene analoge Eingänge haben keinen definierten Zustand. Die dort ausgelesenen Werte schwanken wegen der minimalen elektrischen Felder auf der Nano-Platine und in der nahen Umgebung. Um einen analogen Wert wirklich auswerten zu können, muss zu jedem Zeitpunkt eine klar definierte Spannung anliegen. Ein variabler Widerstand, ein Potentiometer oder irgendein Sensor sollte immer als Spannungsteiler zwischen 0 V und +5 V angeschlossen sein.

Das mBlock-3-Programm `06mb1ock`, das das Potentiometer auswertet, den Wert umrechnet und an die LED schickt, ist sehr einfach.

### So funktioniert das Programm

In einer Endlosschleife wird der Wert am analogen Pin **A7** ständig ausgelesen, umgerechnet und auf den PWM-Pin **D5** wieder ausgegeben.

Der Block **lese analogen Pin (A)...** von der Blockpalette **Roboter** liest den analogen Spannungswert an einem der Analog-Pins aus und liefert einen Zahlenwert zwischen 0 und 1023.

Der vom analogen Pin gelieferte Wert wird durch 4 geteilt, um einen für PWM verwendbaren Wert zwischen 0 und 255 zu liefern.

Der so errechnete Wert wird dann über den Block **setze PWM-Pin...Ausgang auf...** auf den PWM-Pin ausgegeben und lässt die LED entsprechend hell leuchten.

## 7. Tag

### Heute im Adventskalender

- 1x LED blau mit Vorwiderstand

### Lauflicht mit Potentiometer steuern

Ein Potentiometer an einem analogen Eingangspin des Nano-Boards regelt die Geschwindigkeit des Lauflichts.

**Bauteile:** 1x Nano-Board, 1x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 1x Potentiometer, 2x Verbindungskabel, 2x Drahtbrücke

### Das Programm

Das Programm `07mblock` steuert eine LED-Leiste anhand der Einstellung des Potentiometers.

Das Programm `07mblock` lässt die vier LEDs zyklisch als Lauflicht aufleuchten. Die Pinnummern haben jeweils einen Abstand von 2 voneinander. So können sie einfach in einer Schleife hochgezählt werden.

### So funktioniert das Programm

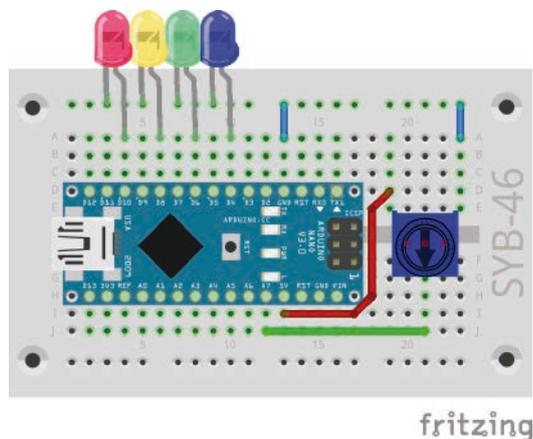
Am Anfang werden die vier für die LEDs verwendeten Pins einmal ausgeschaltet, um sie zu initialisieren, da sie später nur noch über eine Variable angesprochen werden und sonst nicht korrekt als Ausgänge initialisiert würden.

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

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

Die Leuchtzeit der einzelnen LEDs wird über das Potentiometer am analogen Eingang **A7** interaktiv eingestellt. Der analoge Pin **A7** wird in jedem Schleifendurchlauf abgefragt und sein Wert mit 0,001 multipliziert, um brauchbare Werte für das Lauflicht zu erhalten. Der so errechnete Wert legt die Zeit fest, wie lange jede einzelne LED leuchtet.

7. Tag



Lauflicht mit vier LEDs mit Potentiometer steuern.

## 8. Tag

### Heute im Adventskalender

- 3x Verbindungskabel

### LEDs blinken zufällig

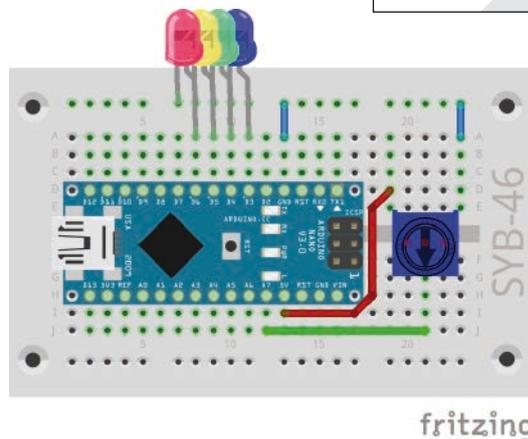
Das Experiment des achten Tages lässt vier LEDs in zufälliger Reihenfolge blinken. Die LEDs stecken diesmal sehr dicht nebeneinander auf dem Steckbrett, da das Programm aufeinander folgende Pinnummern benötigt.

**Bauteile:** 1x Nano-Board, 1x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 1x Potentiometer, 2x Verbindungskabel, 2x Drahtbrücke

### Das Programm

Das Programm `08mblock` lässt vier LEDs mit aufeinander folgenden Pinnummern zufällig abwechselnd blinken.

8. Tag



Vier LEDs blinken zufällig.

### Wie entstehen Zufallszahlen?

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

mBlock 3 verwendet im Arduino-Modus eine Zufallsfunktion, die tatsächlich reproduzierbare Werte erzeugt, was aber in diesem Programm nicht auffällt. In späteren Programmen wird deshalb ein selbst programmierter Zufallsgenerator verwendet.

### 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 die Variable **n** auf eine Zufallszahl zwischen 3 und 6 gesetzt. Diese gibt die Pinnummer der einzuschaltenden LED an. Deshalb benötigt die Schaltung aufeinander folgende Pinnummern.

Die Geschwindigkeit des Farbwechsels wird wie im letzten Programm über das Potentiometer gesteuert, das am analogen Eingang **A7** angeschlossen ist und in jedem Schleifendurchlauf einmal abgefragt wird.

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

## 9. Tag

## 9. Tag

### Heute im Adventskalender

- 1x Steckbrett (SYB-46)

### Analoge Pegelanzeige mit LEDs

Auf einer Pegelanzeige lassen sich analoge Werte auf einen Blick ablesen. Solche Anzeigen aus mehreren LEDs werden zum Beispiel bei Lautstärke- oder Temperaturreglern verwendet. Das Experiment des neunten Tages zeigt den auf dem Potentiometer eingestellten Wert über eine Reihe von vier LEDs an.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 1x Potentiometer, 3x Verbindungskabel, 1x Drahtbrücke

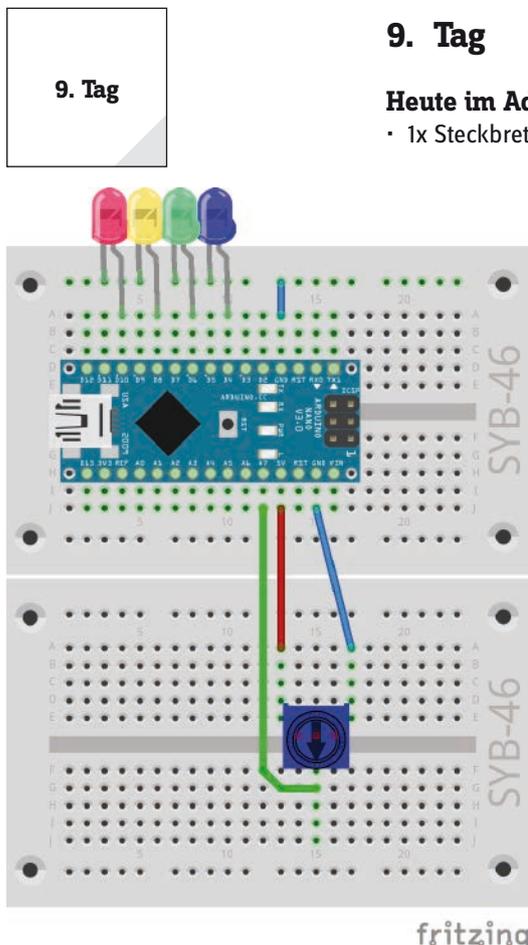
Heute ist ein zweites Steckbrett im Adventskalender. Damit lässt sich das Potentiometer so aufbauen, dass es leichter zugänglich ist.

### Das Programm

Das Programm `09mBlock` liest den eingestellten Wert des Potentiometers am analogen Eingang **A7** aus und zeigt ihn mithilfe einer Schleife auf vier LEDs an.

### So funktioniert das Programm

In der Hauptschleife wird zuerst der analoge Wert des Potentiometers ausgelesen, durch 250 geteilt, um Werte zwischen 0 und 4 zu erhalten, und in der Variablen **a** gespeichert. Der analoge Eingang liefert Werte zwischen 0 und 1023. Würde man den Wert genau durch 256 teilen, würde der Wert 4 nie erreicht. Die Ergebnisse lägen alle zwischen 0 und 3,99. Das Programm teilt den analogen Eingabewert durch 250. Damit kommt ganz am Ende ein Wert von knapp größer als 4 heraus.



Pegelanzeige mit vier LED und Potentiometer.

Der Zähler  $n$  beginnt bei 1. Jetzt startet wieder eine Schleife, die für die vier LEDs viermal durchläuft. Bei jeder LED wird geprüft, ob die Nummer der LED  $n$  größer ist als der in  $a$  gespeicherte Pegelwert. Ist das der Fall, wird die entsprechende LED ausgeschaltet. Dazu wird der Wert  $n$  mit 2 multipliziert und anschließend 2 addiert, um auf die Pin-Nummer der entsprechenden LED zu kommen.

Ist die LED-Nummer nicht größer, also kleiner oder gleich dem Pegelwert, wird die LED eingeschaltet. Auf diese Weise werden immer alle LEDs geprüft, unabhängig davon, ob das Potentiometer hoch- oder heruntergedreht wird.

## 10. Tag

### Heute im Adventskalender

- 2x Verbindungskabel

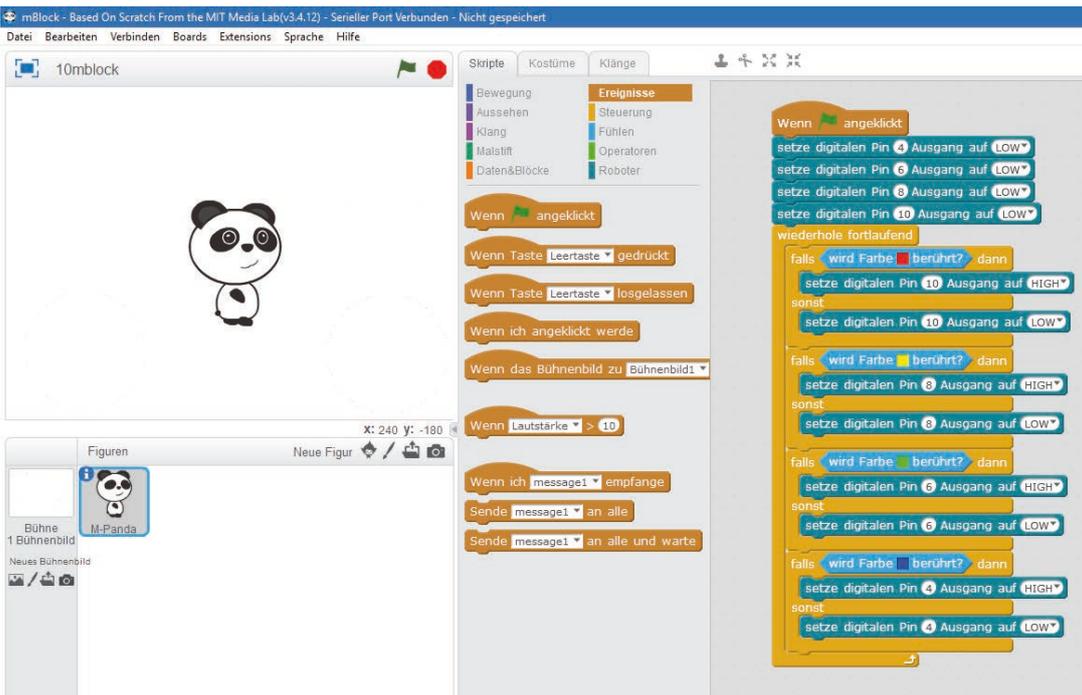
10. Tag

### Nano-Board über mBlock 3 interaktiv steuern

mBlock 3 ermöglicht es nicht nur, Arduino-Sketches zu erzeugen, um sie dann auf den Nano zu laden, sondern bietet auch einen interaktiven Modus, in dem der Nano in Echtzeit mit dem auf dem PC laufenden Programm verbunden ist.

### Echtzeitverbindung mit dem Nano einrichten

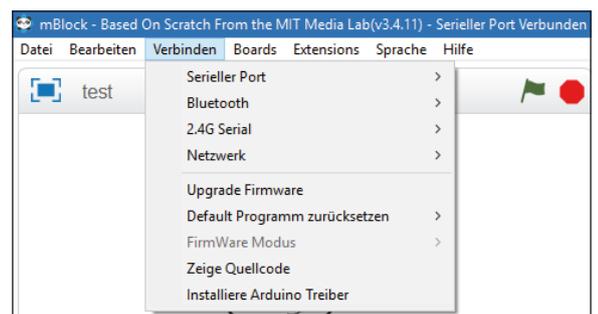
1. Schalten Sie im Menü **Bearbeiten** den **Arduino-Modus** aus. Damit verschwindet rechts das Fenster mit dem Arduino-Sketch. Links oben erscheint dafür die sogenannte Bühne für grafische Programme mit der M-Panda-Symbolfigur.



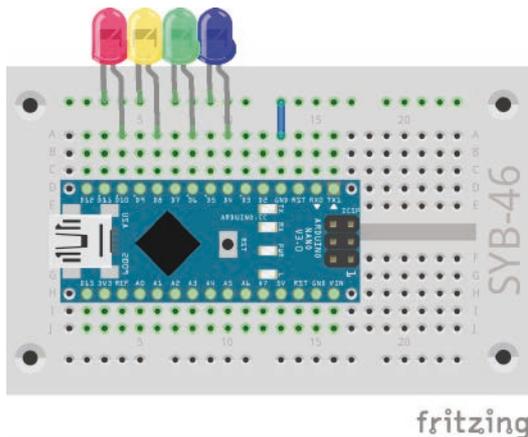
mBlock 3 im Standardmodus.

2. Wählen Sie im Menü **Verbinden / Serieller Port** die verwendete COM-Schnittstelle. Die Titelleiste des Fensters zeigt **Serieller Port verbunden**. Sollte sich die Verbindung nicht aufbauen, wählen Sie im Menü **Verbinden / Installiere Arduino-Treiber**.

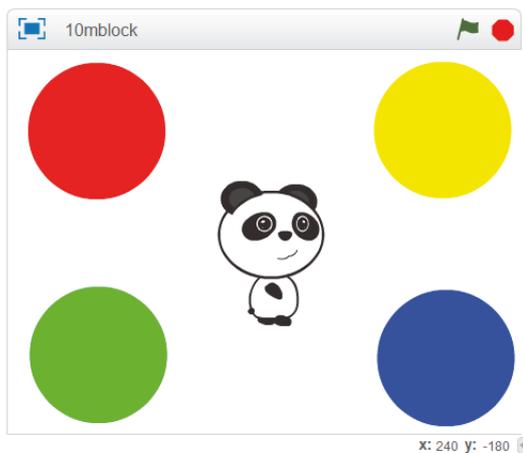
3. Nachdem die Verbindung aufgebaut ist, wählen Sie im Menü **Verbinden / Upgrade Firmware**. Damit wird eine spezielle Firmware zur Echtzeitkommunikation auf dem Nano installiert, was einige Sekunden dauert, während der eine LED auf dem Nano schnell blinkt. Wundern Sie sich nicht, dass die Prozentanzeige auf dem Bildschirm bei 0% stehen bleibt.



mBlock 3 Firmware auf den Nano übertragen.



Vier LEDs zeigen die Position des M-Panda.



Die Bühne mit einem Bühnenbild und der Figur M-Panda.



Der Block **Wenn grüne Fahne angeklickt** ist einer der wichtigen Standardblöcke aus Scratch. Er startet ein Programm, wenn der Benutzer auf die grüne Fahne oben auf der Bühne klickt.

Klicken Sie auf die grüne Fahne, startet das Programm. Alle vier LEDs werden initialisiert und ausgeschaltet.



Jetzt können Sie die Figur M-Panda mit der Maus auf der Bühne verschieben. Berührt sie eine der farbigen Flächen, leuchtet die entsprechende LED.

Eine Endlosschleife fragt nacheinander ab, ob der M-Panda eine der vier Farben berührt. Dazu werden **falls...dann...sonst**-Blöcke verwendet. Wird die Farbe berührt, wird die passende LED eingeschaltet, wird sie nicht berührt, wird die LED ausgeschaltet. Ohne das Ausschalten wären irgendwann alle LEDs eingeschaltet, wenn der M-Panda alle Farbflächen einmal berührt hat.

Um die Farbe im Block **wird Farbe ... berührt** auszuwählen, tippen Sie in das Farbfeld des Blocks. Eine Pipette erscheint, mit der Sie auf der Bühne die gewünschte Farbe auswählen können.



Die Schleife wird endlos wiederholt, bis der Benutzer das Programm mit einem Klick auf das rote Stopp-symbol oben rechts auf der Bühne beendet.



Die Figurenpalette in mBlock 3.

**Bauteile:** 1x Nano-Board, 1x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 1x Drahtbrücke

### Das Programm

Das Programm `10mblock` lässt eine der vier LEDs aufleuchten, wenn die Symbolfigur M-Panda die Fläche in der entsprechenden Farbe berührt.

Die LEDs **RX** und **TX** auf dem Nano blinken ständig, da im interaktiven Modus permanent Daten zwischen dem Nano-Board und dem PC hin- und hergeschickt werden.

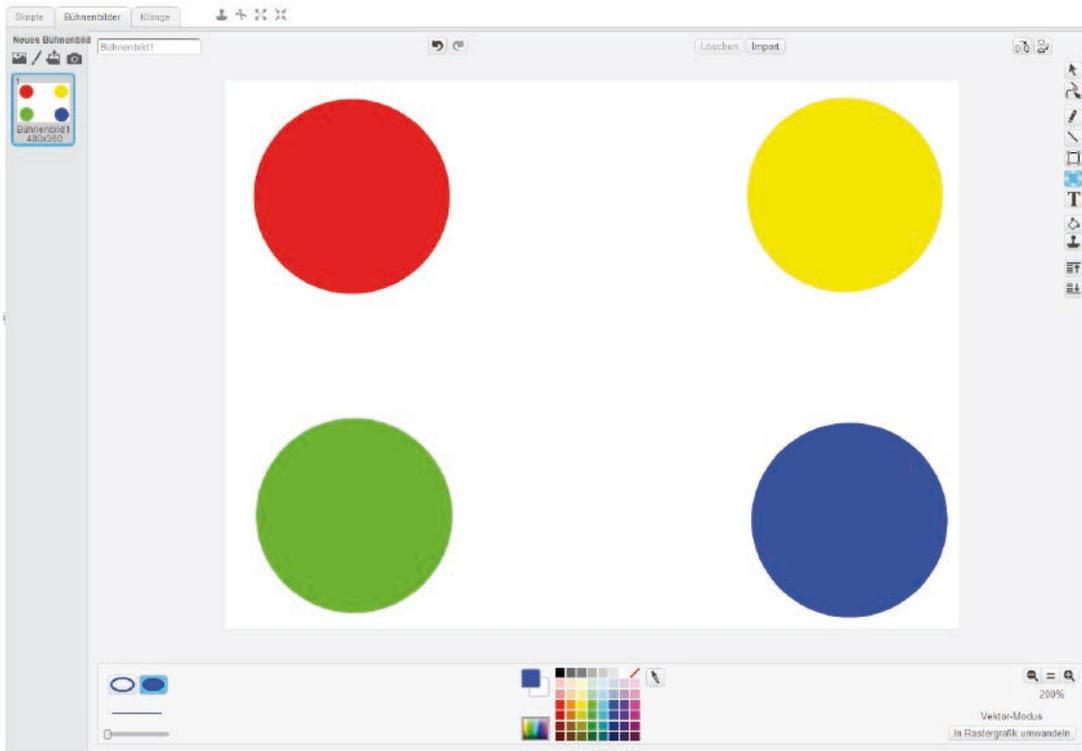
### So funktioniert das Programm

mBlock 3 basiert auf der Programmiersprache Scratch und enthält im interaktiven Modus auch deren kompletten Funktionsumfang. Im Arduino-Modus steht nur ein Teil der Funktionen zur Verfügung, die sich direkt in Arduino-Sketches umsetzen lassen.

### Das Bühnenbild

Jedes Scratch-Programm kann verschiedene Bühnenbilder enthalten. Klicken Sie links auf der Figurenpalette auf das Bühnenbild, um es zu bearbeiten oder ein neues Bühnenbild zu malen.

Schalten Sie oben auf die Registerkarte **Bühnenbilder**, können Sie das Bühnenbild bearbeiten. mBlock 3 enthält ein Grafikprogramm für Vektorgrafik und Rastergrafik, mit dem sich Bühnenbilder und neue Figuren gestalten lassen.



Das Grafikprogramm in mBlock 3

Mit dem Kreiswerkzeug erzeugen Sie bei gedrückter [Umschalt]-Taste Kreise. Schalten Sie unten links auf den Füllmodus, um ausgefüllte Kreise zu zeichnen. Die Farbe wählen Sie auf der Palette unten in der Mitte.

Mit dem Auswahlwerkzeug ganz oben wählen und kopieren Sie den fertigen Kreis mit [Strg]+[C] und [Strg]+[V]. Auf diese Weise erhalten Sie vier gleich große Kreise, die Sie nur noch an die passenden Stellen kopieren müssen.

Zum Schluss füllen Sie die Kreise mit dem Farbeimer-Werkzeug in den gewünschten Farben.

Um das Programm wieder zu bearbeiten, klicken Sie auf **M-Panda**. Schalten Sie dann oben auf die Registerkarte **Skripte**. Hier finden Sie die Programmblöcke.

## 11. Tag

### Heute im Adventskalender

- 1x Taster
- 1x 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. 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  $\Omega$  verwendet.

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

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

11. Tag

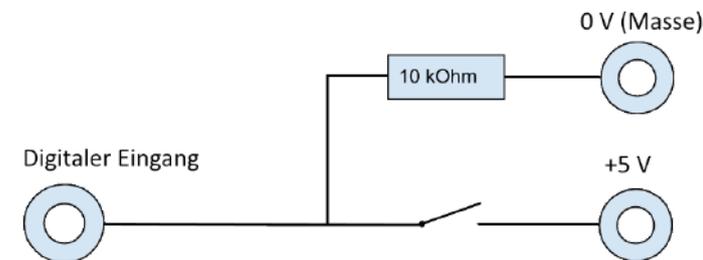
Farbe	Widerstandswert in Ohm			
	1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multiplikator)	4. Ring (Toleranz)
Silber			$10^{-2} = 0,01$	$\pm 10 \%$
Gold			$10^{-1} = 0,1$	$\pm 5 \%$
Schwarz		0	$10^0 = 1$	
Braun	1	1	$10^1 = 10$	$\pm 1 \%$
Rot	2	2	$10^2 = 100$	$\pm 2 \%$
Orange	3	3	$10^3 = 1.000$	
Gelb	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	$10^9 = 1.000.000.000$	

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

### Taster

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

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



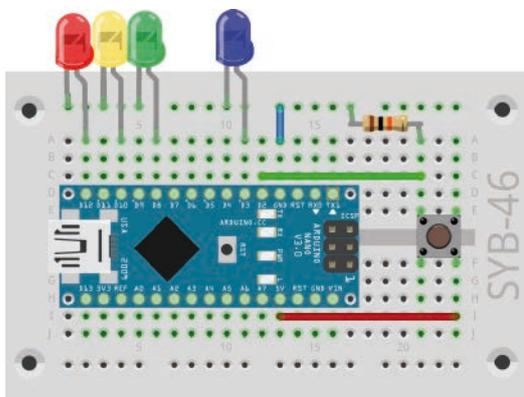
Schaltschema eines Tasters mit Pulldown-Widerstand.

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

### Ampel mit Fußgängerblinklicht mit Taster schalten

Das Projekt des elften Tages stellt eine einfache Ampelschaltung mit Fußgängerblinklicht dar. Während der Rotphase der Verkehrsampel blinkt die blaue LED. Da nur vier LEDs im Adventskalender sind, stellt die Schaltung keine Rot/Grün-Fußgängerampel dar, wie sie in Mitteleuropa üblich sind, sondern ein einfaches Fußgängerblinklicht, wie es in einigen anderen Ländern verwendet wird. Es zeigt den Fußgängern, dass die Verkehrsampel für die Autos rot leuchtet.

**Bauteile:** 1x Nano-Board, 1x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 1x Taster, 1x 10-kOhm-Widerstand (braun - schwarz - orange), 1x Drahtbrücke, 2x Verbindungskabel



fritzing

### Das Programm

Das Programm `11mblock` basiert auf der Ampelschaltung des vierten Tages. Eine Endlosschleife prüft regelmäßig, ob der Taster gedrückt ist. Ist dies der Fall, wird der Ampelzyklus gestartet. Solange der Taster nicht gedrückt ist, bleibt die Verkehrsampel grün und das Fußgängerblinklicht ausgeschaltet.

### So funktioniert das Programm

Am Anfang werden vier Variablen für die Pinnummern der vier LEDs sowie eine weitere für die Pinnummer des Tasters angelegt. Alle vier LEDs werden ausgeschaltet. Der Pin für den Taster braucht nicht eigens initialisiert zu werden.

Danach startet eine Endlosschleife. In jedem Durchlauf wird in einem **falls ... dann ...** Block der digitale Pin 2 abgefragt, an dem der Taster angeschlossen ist. Ist der Taster gedrückt, ergibt die Abfrage den Logikwert **wahr**. In diesem Fall startet der bereits bekannte Ampelzyklus. Während der Rotphase der Verkehrsampel schaltet eine Schleife 20-mal für je 0,05 Sekunden die blaue LED ein und wieder aus.

10-kOhm-Widerstand Fußgängerampel mit einem Taster schalten.

## 12. Tag

### Heute im Adventskalender

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

12. Tag

### Lauflicht in zwei Richtungen mit einstellbarer Geschwindigkeit

Lauflichter sind immer wieder beliebte Effekte, nicht nur für Werbung und Partyräume. Das Experiment des zwölften Tages lässt beim Druck auf einen Taster vier LEDs als Lauflicht leuchten. Der andere Taster lässt das Lauflicht in umgekehrter Richtung laufen. Über ein Potentiometer steuert man die Geschwindigkeit des Lauflichts.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 1x Potentiometer, 2x Taster, 2x 10-kOhm-Widerstand (braun - schwarz - orange), 5x Verbindungskabel, 5x Drahtbrücke

### Das Programm

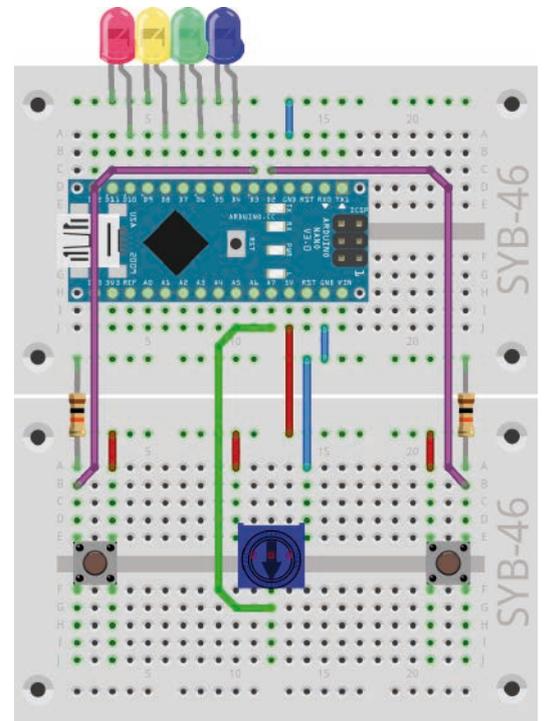
Das Programm `12mblock` funktioniert ähnlich wie das Programm des siebten Tages, fragt nur zusätzlich die beiden Taster ab, um die Richtung des Lauflichts mithilfe einer zusätzlichen Variable auszuwählen.

### So funktioniert das Programm

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

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

In jedem Durchlauf der Schleife leuchtet die LED am Pin `n`. Die Zeit wird durch Abfrage des Potentiometers festgelegt. Danach werden die beiden Taster abgefragt. Ist der Taster am Pin 2 gedrückt, wird die Variable `i` auf 2 gesetzt, damit das Lauflicht nach links läuft. Ist der Taster am Pin 3 gedrückt, wird die Variable `i` auf -2 gesetzt, damit das Lauflicht nach rechts läuft. Die Pinnummer `n` wird auf den neuen Wert gesetzt. Danach wird noch geprüft, ob `n` den Bereich der möglichen Werte für die LED-Pins überschritten hat. Hat `n` durch die Veränderung einen Wert größer als 10, beginnt das Lauflicht rechts wieder bei Pin 4. Ist der neue Wert von `n` dagegen kleiner als 4, beginnt das Lauflicht wieder links bei Pin 10.



Lauflicht mit vier LEDs, Potentiometer und zwei Tastern mit PullDown-Widerständen.

## 13. Tag

### Heute im Adventskalender

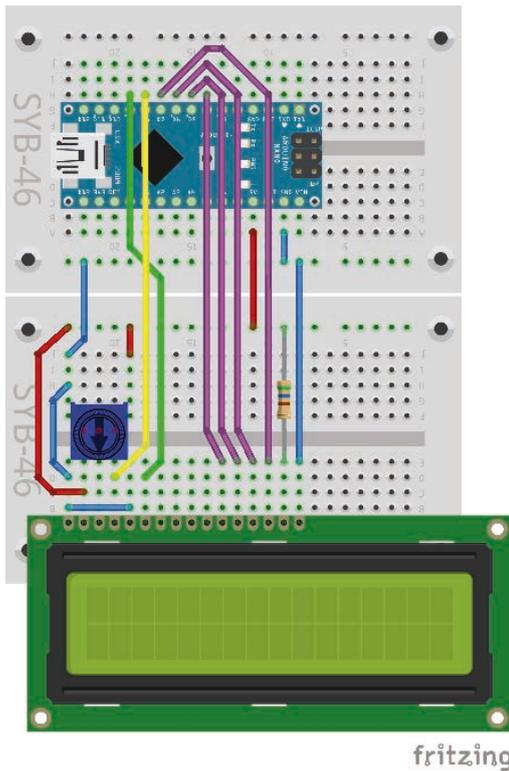
- 1x LCD-Modul
- 1x Widerstand 560 Ohm (grün - blau - braun)

13. Tag

Heute ist ein LCD-Modul im Adventskalender. Derartige Anzeigemodule sind in vielen elektronischen Geräten eingebaut. Das LCD-Modul hat eine 16-polige Anschlussleiste. Damit wird es auf das Steckbrett gesteckt.

Das LCD-Modul im Adventskalender arbeitet zeichenorientiert. Das bedeutet, Sie brauchen sich nicht um die Ansteuerung einzelner Pixel zu kümmern. Ein eigener Controller auf dem LCD-Modul verarbeitet Zeichen, die als ganze oder halbe Bytes auf den Datenleitungen an das LCD-Modul gesendet werden. Fast alle derartigen Anzeigemodule, wie auch das im Adventskalender enthaltene, sind zum Quasi-Standard HD44780 kompatibel. Dieser bezeichnet den Hitachi-Chip, der solche LCD-Module steuert.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LCD Modul, 1x 560-Ohm-Widerstand (grün - blau - braun), 1x Potentiometer, 7x Verbindungskabel, 7x Drahtbrücke (unterschiedliche Längen)



Die Steuerleitungen auf Ihrem Board sind gelb und grün, die Datenleitungen violett, +5 V ist rot und die Masse blau.

Der Aufbau der Schaltung mag auf den ersten Blick etwas unübersichtlich aussehen. mBlock 3 gibt diese Anschlussbelegung vor. Bei Verwendung der Arduino IDE kann man sie frei ändern. Mit dem Potentiometer stellen Sie den Kontrast des LCD-Moduls so ein, dass die Schrift gut zu erkennen ist.

### Pinbelegung eines HD44780-kompatiblen LCD-Moduls

Pin	Funktion	Beschreibung	Nano Pin
1	VSS	Stromversorgung Masseleitung 0 V	0 V
2	VDD	Stromversorgung +5 V	+5 V
3	VO	Kontrasteinstellung, 0 V ... 5 V	Potentiometer
4	RS	Register Select	Pin D8
5	RW	Read / Write, wenn vom LCD-Module nichts ausgelesen wird, mit 0 V verbinden	0 V
6	E	Enable (Umschaltsignal)	Pin D9
7	D0	Datenbit 0 (im 4-Bit-Modus nicht benötigt)	im 4-Bit-Modus nicht benötigt
8	D1	Datenbit 1 (im 4-Bit-Modus nicht benötigt)	im 4-Bit-Modus nicht benötigt
9	D2	Datenbit 2 (im 4-Bit-Modus nicht benötigt)	im 4-Bit-Modus nicht benötigt
10	D3	Datenbit 3 (im 4-Bit-Modus nicht benötigt)	im 4-Bit-Modus nicht benötigt
11	D4	Datenbit 4	Pin D4
12	D5	Datenbit 5	Pin D5
13	D6	Datenbit 6	Pin D6
14	D7	Datenbit 7	Pin D7
15	A	Hintergrundbeleuchtung, 560 Ohm Vorwiderstand erforderlich	Vorwiderstand 560 Ohm
16	K	Hintergrundbeleuchtung Masseleitung	0 V

#### Schaltung genau überprüfen

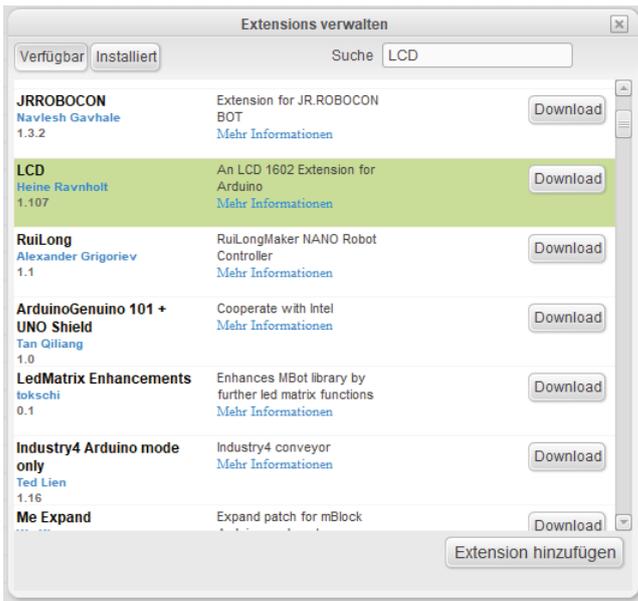
Bevor Sie den Nano und damit die Schaltung mit Strom versorgen, prüfen Sie alle Anschlussdrähte noch einmal genau. Eine falsch angeschlossene +5-V-Leitung kann das LCD-Modul beschädigen. Die maximale Spannung am Kontrast-Pin darf nur weniger als +5 V betragen. Drehen Sie also das Potentiometer nie ganz auf.

Das LCD-Modul unterstützt zwei Modi zur Übertragung der Daten. Im 8-Bit-Modus kann ein komplettes Zeichen auf einmal übertragen werden. Meistens wird aber der 4-Bit-Modus verwendet, da er vier Ports am sendenden Gerät spart. Hier werden die Daten eines Zeichens in zwei Blöcken hintereinander übertragen.

#### LCD-Modul mit mBlock 3 steuern

Der Funktionsumfang von mBlock 3 lässt sich durch vielfältige sogenannte Extensions erweitern. Darunter findet sich auch eine Extension für LCD-Module.

Wählen Sie im Menü **Extensions / Extensions verwalten**. Suchen Sie auf der Seite **Verfügbar** oben im Suchfeld nach **LCD**. Installieren Sie mit einem Klick auf **Download** die Extension **LCD**.



LCD-Extension für mBlock 3 herunterladen und installieren.

Nach der Installation erscheinen auf der Blockpalette **Roboter** automatisch die neuen Blöcke der LCD-Extension.

### Wichtig

Die LCD-Extension funktioniert nur im Arduino-Modus, nicht im interaktiven Modus von mBlock 3.

### Das Programm

Das Programm `13mblock` zeigt einen Text auf dem LCD-Modul.



Ein einfaches Programm für einen zweizeiligen Text auf dem LCD-Modul.

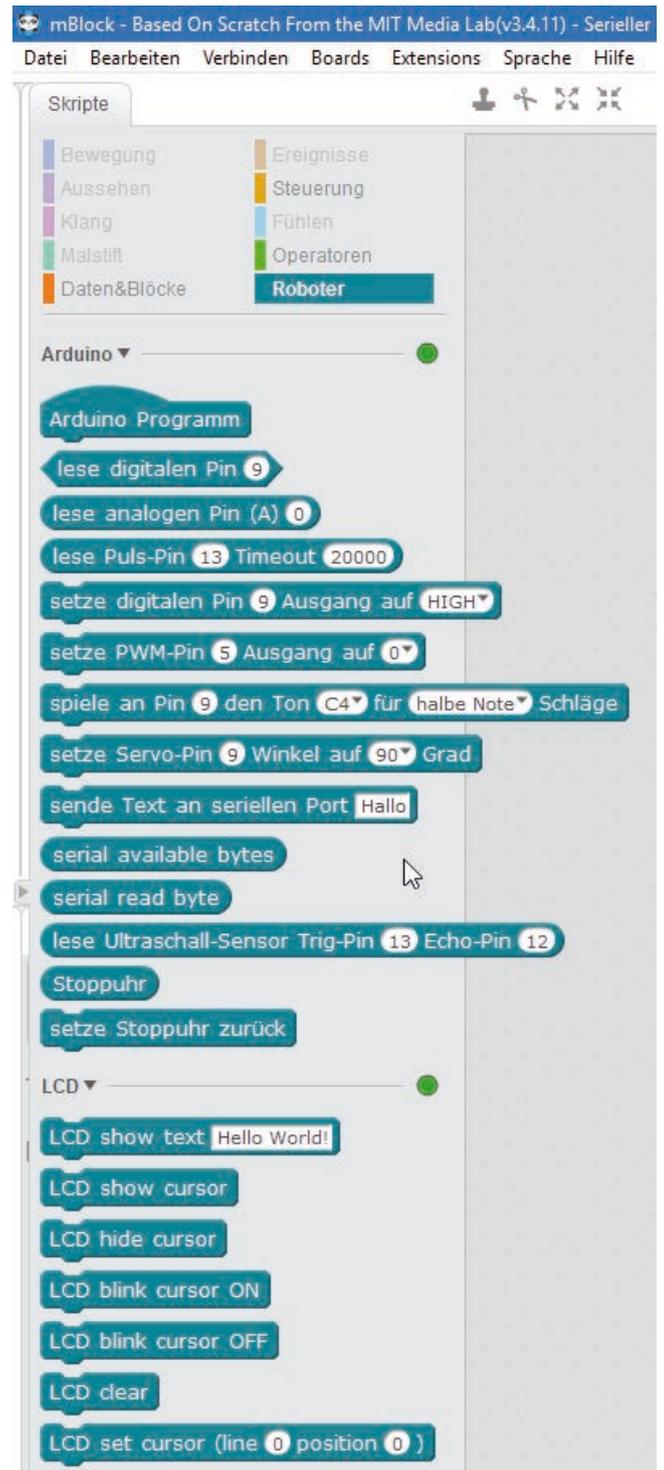
### So funktioniert das Programm

Das Programm braucht keine Schleife und keine Variablen. Nachdem es einmal gelaufen ist, bleibt der Text auf dem LCD-Modul stehen.

Der Block **LCD set cursor (line ... position ...)** aus der LCD-Extension setzt den Cursor auf die im ersten Feld angegebene Zeile (**0** oder **1**) und die im zweiten Feld angegebene Spalte (**0-15**), in diesem Fall an den Anfang der oberen Zeile 0.

Der Block **LCD show text ...** aus der LCD-Extension zeigt einen Text auf dem LCD-Modul an. Er beginnt an der aktuellen Cursorposition. Der erste Block schreibt das Wort `Adventskalender` an den Anfang der oberen Zeile.

Der Block **LCD set cursor (line ... position ...)** aus der LCD-Extension setzt den Cursor auf den Anfang der unteren Zeile 1. Hier wird der Text `* Arduino 2019 *` angezeigt.



Die Blöcke der LCD-Extension.

## 14. Tag

## 14. Tag

## Heute im Adventskalender

- 2x Verbindungskabel

Die Kabel werden erst an einem späteren Tag gebraucht. Der Schaltungsaufbau ist gegenüber dem 13. Tag unverändert.

**Laufschrift auf dem LCD-Modul**

Lässt man einen Text als Laufschrift in das LCD-Modul hinein laufen, fällt er deutlich mehr auf als nur ein statischer Text.

**Das Programm**

Das Programm `14mblock` zeigt einen Text, der animiert von rechts in das LCD-Modul hinein läuft.

**So funktioniert das Programm**

Am Anfang jedes Durchlaufs der Endlosschleife löscht ein Block **LCD clear** aus der LCD Extension die Anzeige auf dem LCD-Modul. Danach ist die Anzeige leer, und der neue Text kann hinein laufen.

Die Variable `i` bezeichnet die Position, an der der Text beginnen soll. Sie wird am Anfang auf 15 gesetzt und bezeichnet so das letzte Zeichen einer Zeile.

Jetzt läuft eine Schleife 16-mal durch. In jedem Durchlauf wird der LCD-Cursor um ein Zeichen weiter nach links gesetzt und danach die ganze Textzeile an dieser Position angezeigt. Buchstaben, die nicht mehr in die Anzeige passen, werden einfach abgeschnitten. Der anzuzeigende Text hat am Ende noch ein Leerzeichen, das das letzte Zeichen aus dem vorherigen Durchlauf überschreibt, damit keine Buchstaben in der Anzeige stehenbleiben.

fritzing

Nach dem ersten Text in der oberen Zeile wird auf die gleiche Weise ein zweiter Text in der unteren Zeile angezeigt.

Der unveränderte Schaltungsaufbau.

## 15. Tag

## 15 Tag

## Heute im Adventskalender

- 1x Batteriekasten

**Batteriekasten**

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

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

**Stoppuhr**

Arduino-kompatible Platinen haben zwar keine Echtzeituhr, die die wirkliche Zeit anzeigt, aber einen Timer, der ständig läuft und deshalb als Stoppuhr verwendet werden kann.

Stoppuhr mit LCD-Modul und Batteriekasten.

fritzing

Die Schaltung verwendet einen einzigen Taster für Start, Stopp und Reset der Stoppuhr. Die untere Zeile des LCD-Moduls zeigt, welche Funktion der nächste Tastendruck auslöst.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LCD-Modul, 1x 560-Ohm-Widerstand (Grün-Blau-Braun), 1x 10-kOhm-Widerstand (Braun-Schwarz-Orange), 1x Taster, 1x Potentiometer, 1x Batteriekasten, 9x Verbindungskabel, 7x Drahtbrücke (unterschiedliche Längen)

### Das Programm

Das Programm `15mblock` lässt auf dem LCD-Modul eine Stoppuhr laufen. Übertragen Sie das Programm im Arduino-Modus auf den Nano.

### So funktioniert das Programm

Das Programm besteht aus einer großen Endlosschleife, in der am Anfang die Stoppuhr und auch die Anzeige auf dem LCD-Modul zurückgesetzt werden. Nachdem der Benutzer die Taste gedrückt hat, läuft die Stoppuhr, bis er sie erneut drückt. Wird die Taste ein drittes Mal gedrückt, startet die Endlosschleife von Neuem.

Der Block **setze Stoppuhr zurück** aus der Arduino Extension setzt die interne Stoppuhr auf dem Nano zurück. Direkt zu sehen ist davon noch nichts.

Der Block **LCD clear** aus der LCD Extension löscht das LCD-Modul und setzt die Cursorposition für den nächsten Text in die linke obere Ecke.

Der Block **LCD show text ...** aus der LCD Extension zeigt einen Text auf dem LCD-Modul an. Dieser beginnt an der aktuellen Cursorposition. Der erste Block schreibt das Wort `Zeit:` an den Anfang der oberen Zeile.

Der Block **LCD set cursor (line ... position ...)** aus der LCD Extension setzt den Cursor auf die im ersten Feld angegebene Zeile (**0** oder **1**) und die im zweiten Feld angegebene Spalte (**0-15**), in diesem Fall an den Anfang der unteren Zeile. Hier wird das Wort `Taste:` geschrieben.

An der Position 7 in der oberen Zeile wird der Text `0.00` angezeigt. Hier erscheint später, wenn die Stoppuhr läuft, die Zeit. An der gleichen Position in der unteren Zeile erscheint das Wort `Start`. Hier wird immer angezeigt, was der nächste Tastendruck bewirkt.



So sieht die Anzeige beim Programmstart aus.

Jetzt wartet das Programm, bis der Benutzer die Taste drückt.

Der Block **warte bis ...** von der Blockpalette **Steuerung** wartet, bis der Logikwert im Feld wahr ist. An dieser Stelle wird der Wert des digitalen Pin **D2** ausgelesen, an dem der Taster angeschlossen ist. Damit ein längerer Tastendruck nicht zu Fehlinterpretationen führt, wartet das Programm auch noch, bis der Taster losgelassen wird. Erst dann soll die Stoppuhr starten. Mit dem Block **nicht ...** von der Blockpalette **Operatoren** lässt sich ein Logikwert umkehren. Somit wartet der zweite **warte bis ...** Block, bis der Taster wieder losgelassen wird.

Danach erscheint in der unteren Zeile anstelle des dort angezeigten `Start` das Wort `Stopp`, da der nächste Tastendruck die Stoppuhr anhält.

Nun startet eine Schleife, in der die Stoppuhr so lange läuft, bis wieder die Taste am Pin **D2** gedrückt wird. In jedem Durchlauf wird der Wert des Blocks **Stoppuhr** aus der Arduino Extension an der Position 7 in der oberen Zeile des LCD-Moduls angezeigt. Dieser Block kann wie eine Variable auch für Berechnungen verwendet werden.

Drückt der Benutzer die Taste, wird die Schleife beendet. Danach wird wieder gewartet, bis die Taste losgelassen wird. Anschließend wird in der unteren Zeile das Wort `Reset` angezeigt. Der nächste Tastendruck soll die Stoppuhr zurücksetzen.

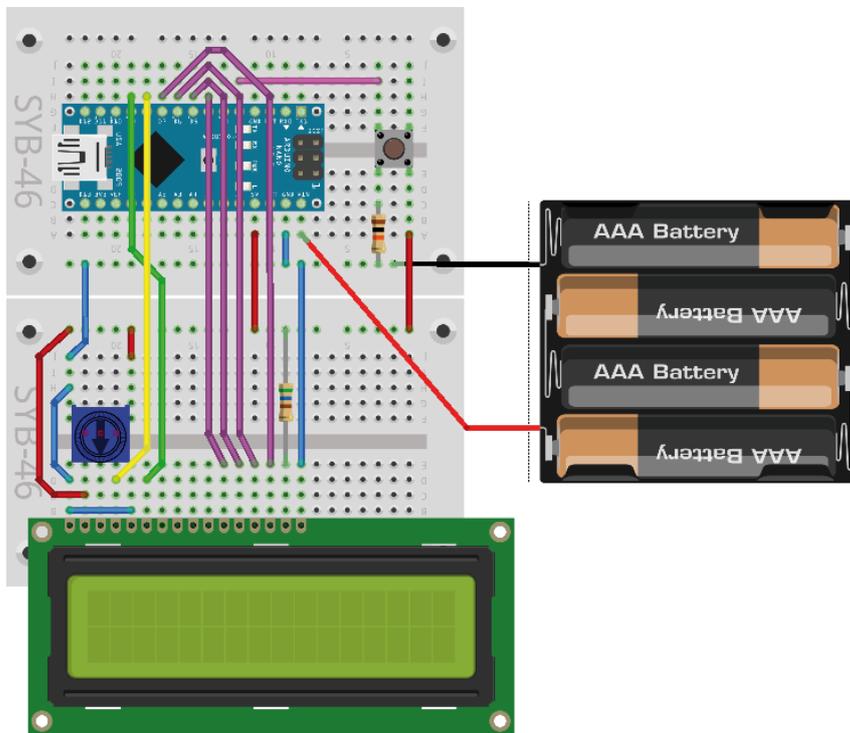
Drückt der Benutzer jetzt wieder die Taste, wird der aktuelle Durchgang der Hauptschleife beendet. Die Schleife springt wieder an den Anfang, wo die Stoppuhr zurückgesetzt wird.

## 16. Tag

## 16. Tag

## Heute im Adventskalender

- 1x Verbindungskabel



Der unveränderte Schaltungsaufbau.

Das Kabel wird erst an einem späteren Tag gebraucht. Der Schaltungsaufbau ist gegenüber dem 15. Tag unverändert.

### Weihnachtsbotschaft mit Schreibmaschineneffekt

Auf dem LCD-Modul erscheint eine Weihnachtsbotschaft. Sie leuchtet nicht auf einmal auf, sondern wird wie mit einer Schreibmaschine Buchstabe für Buchstabe geschrieben.

### Das Programm

Das Programm `16mblock` zeigt den laufenden Text auf dem LCD-Modul. Übertragen Sie das Programm im Arduino-Modus auf den Nano.

### So funktioniert das Programm

Eine Endlosschleife wartet zunächst, bis der Taster am Pin `D2` gedrückt wird. Dann wird die Anzeige auf dem LCD-Modul gelöscht, der Cursor angezeigt und auf Blinken gesetzt. Dazu gibt es eigene Blöcke. Sie brauchen das Cursor-Blinken nicht selbst zu programmieren.

Die Variable `n` wird auf 1 gesetzt. Sie gibt die Cursorposition beim Schreiben an. Eine weitere Schleife schreibt so oft, wie das Wort `Frohe` lang ist, nacheinander die einzelnen Buchstaben in die obere Zeile des LCD-

Moduls. Dazu wird ein Block **Zeichen ... von ...** verwendet, der aus einer Zeichenkette einzelne Zeichen herausgreift.

Auf die gleiche Weise wird anschließend das Wort `Weihnachten` in der unteren Zeile des LCD-Moduls geschrieben. Anschließend wird der Cursor wieder ausgeblendet. Der Text bleibt stehen, bis der Benutzer wieder die Taste drückt.

## 17. Tag

## 17. Tag

## Heute im Adventskalender

- 1x Verbindungskabel

Das Kabel wird erst an einem späteren Tag gebraucht.

### Analoge Eingänge auslesen

Das Experiment des 17. Tages zeigt die Werte von vier analogen Eingängen an. Solange an einem analogen Eingang des Nano kein definiertes Signal anliegt, liefern diese Eingänge relativ zufällige Werte, die durch Potentiale auf der Platine entstehen. Schließen Sie an einen unbeschalteten Eingang ein übrig gebliebenes Verbindungskabel oder einen Draht an, können Sie damit sogar den Elektrosmog aus der Umgebung auffangen, was sich am gemessenen Wert des Eingangs zeigt.

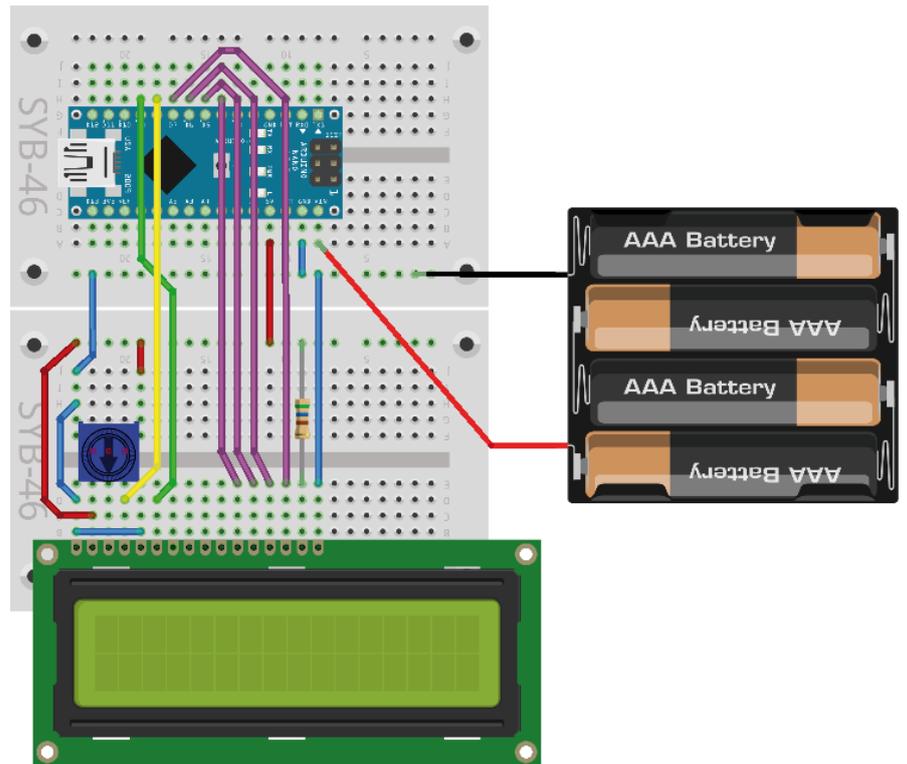
**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LCD-Modul, 1x 560-Ohm-Widerstand (Grün-Blau-Braun), 1x Potentiometer, 1x Batteriekasten, 7x Verbindungskabel, 7x Drahtbrücke (unterschiedliche Längen)

### Das Programm

Das Programm `17mblock` zeigt die Zustände von vier analogen Eingängen auf dem LCD-Modul an. Übertragen Sie das Programm im Arduino-Modus auf den Nano.

### So funktioniert das Programm

Am Anfang werden die Pin-Nummern der verwendeten analogen Eingänge auf dem LCD-Modul als Legende der Werte angezeigt. Anschließend liest eine Endlosschleife nacheinander vier analoge Eingänge aus und schreibt die Werte an die entsprechenden Positionen hinter die angezeigten Pin-Nummern. Nach 0,5 Sekunden läuft die Schleife erneut.



Diese Schaltung zeigt die Werte von vier analogen Eingängen auf dem LCD-Modul an.

fritzing

## 18. Tag

### Heute im Adventskalender

- 1x Knete
- 1x Widerstand 20 MOhm (Rot-Schwarz-Blau)

### Sensorkontakt aus Knete

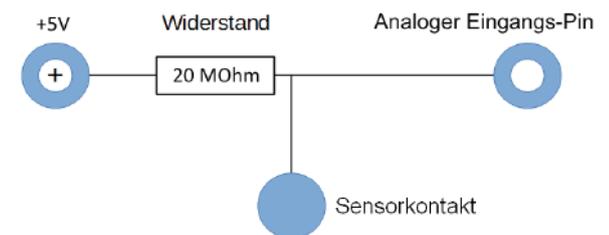
Ampeln, Türöffner, Lichtschalter und Automaten werden heute oft mit Sensorkontakten gesteuert, die man nur zu berühren braucht. Taster, die wirklich gedrückt werden müssen, werden immer seltener.

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

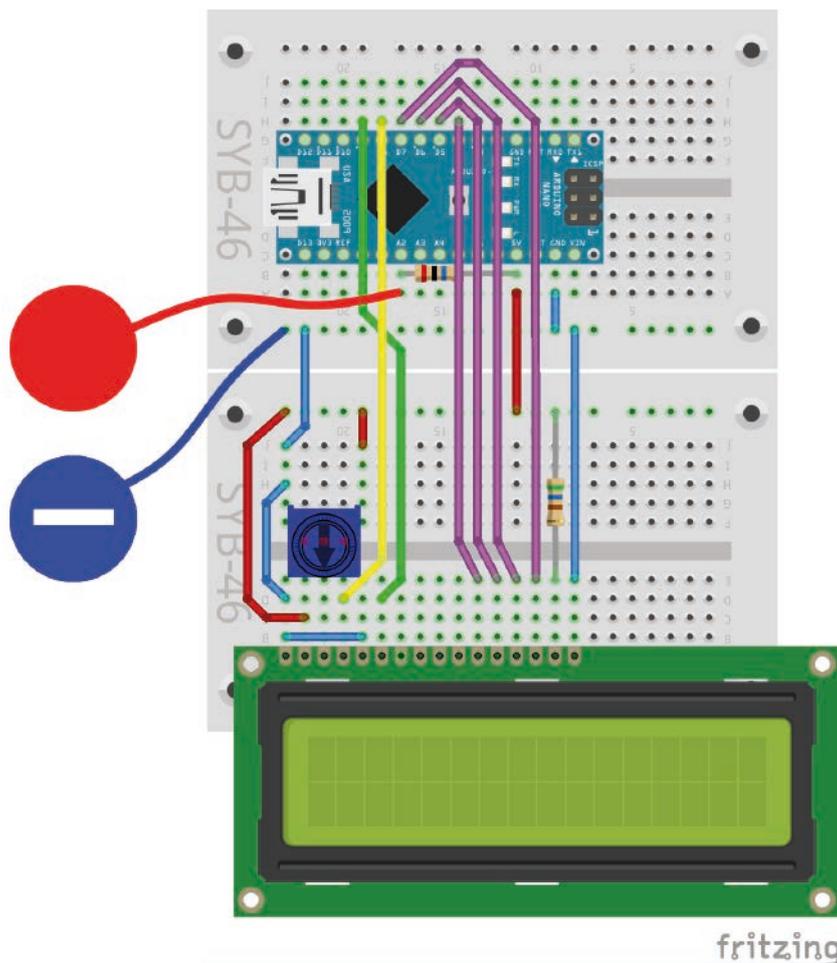
Wie hoch allerdings der Widerstand zwischen Hand und Masse wirklich ist, hängt von vielen Dingen ab, unter anderem von Schuhen und Fußboden. Barfuß im nassen Gras ist die Verbindung zur Masse der Erde am besten, aber auch auf Steinfußboden funktioniert es meistens gut. Holzfußböden isolieren stärker,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

Knete leitet den Strom etwa so gut wie menschliche Haut. Sie lässt sich leicht in jede beliebige Form bringen, und ein Knetekontakt fasst sich viel besser an als ein einfaches Stück Draht. Die Fläche, mit der die Hand den Kontakt berührt, ist deutlich größer. So kommt es nicht so leicht zu einem "Wackelkontakt". Stecken Sie das eine Ende eines Verbindungskabels in ein Stück Knete und das andere Ende wie auf der Abbildung in das Steckbrett.

18. Tag



Schaltenschema für einen Sensorkontakt.



Sensorkontakt und Massekontakt aus Knete.

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

Das Experiment des 18. Tages schaltet einen Text auf dem LCD-Modul über einen einfachen Sensorkontakt um.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LCD-Modul, 1x 560-Ohm-Widerstand (Grün-Blau-Braun), 1x 20-MOhm-Widerstand (Rot-Schwarz-Blau), 1x Potentiometer, 2x Knete, 10x Verbindungskabel, 6x Drahtbrücke (unterschiedliche Längen)

### Das Programm

Das Programm `18mBlock` zeigt den Zustand des Sensorkontaktes auf dem LCD-Modul an.

### So funktioniert das Programm

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

Berührt man den Sensorkontakt und am besten den Massekontakt gleichzeitig, fällt der Wert des analogen

Eingangs-Pins deutlich, bis fast auf 0. Das Programm prüft, ob der Grenzwert von 200 unterschritten wird. In diesem Fall wird in der unteren Zeile des LCD-Modus das Wort `ein` angezeigt. Solange der Eingangswert über 200 liegt, erscheint umgekehrt das Wort `aus`. Bei beiden anzuzeigenden Zeichenfolgen sind einmal hinten und einmal vorne je vier Leerzeichen angehängt. Damit erscheint das Wort `ein` weiter links als das Wort `aus`. Durch diesen Effekt fällt das Umschalten noch mehr auf, als wenn beide Wörter an der gleichen Position auf dem LCD-Modul angezeigt würden.

## 19. Tag

### 19. Tag

#### Heute im Adventskalender

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

#### Zahlen raten

Das Projekt des 19. Tages ist ein einfaches Ratespiel, in dem ein Spieler eine vom Nano zufällig gewählte Zahl in möglichst wenigen Schritten erraten soll.

Die Zahlen werden auf dem LCD-Modul angezeigt und über Taster eingegeben. Dazu zeigt die untere Zeile der Anzeige Symbole zur Erklärung der Taster:

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LCD-Modul, 1x 560-Ohm-Widerstand (Grün-Blau-Braun), 3x 10-kOhm-Widerstand (Braun-Schwarz-Orange), 3x Taster, 1x Potentiometer, 9x Verbindungskabel, 10x Drahtbrücke (unterschiedliche Längen)

#### Das Programm

Das Programm `19mBlock` generiert eine geheime Zahl kleiner als 100. Mit den beiden oberen Tastern stellen Sie einen Tipp ein und geben ihn mit dem unteren Taster ab. Das Programm zeigt auf dem LCD-

Modul an, ob die geheime Zahl kleiner oder größer als der letzte Tipp ist. Wenn die Zahl richtig getippt wurde, wird die Anzahl der Versuche angezeigt.

### So funktioniert das Programm

In der oberen Zeile des LCD-Moduls werden die statischen Textelemente der Spielanzeige dargestellt. Anstelle der drei Fragezeichen zeigt das Spiel später den letzten Tipp. In der unteren Zeile wird angezeigt, ob die Lösungszahl größer oder kleiner als dieser Tipp ist.

Der Block **Zufallszahl von ... bis ...** erzeugt zwar im interaktiven Modus auf dem PC zufällige Werte, im Arduino-Modus wird aber der Zufallsgenerator nicht richtig initialisiert, sodass bei jedem Programmstart immer wieder die gleiche Folge von Zufallszahlen erzeugt wird. Deshalb verwenden wir hier ein eigenes Verfahren zur Ermittlung einer Zufallszahl.

Zwei nicht beschaltete analoge Pins werden ausgelesen. Der Block **... mod ...** ermittelt den unteilbaren Rest bei einer Division, der noch unvorhersehbarer ist als der Wert des analogen Eingangs selbst. Aus den beiden so ermittelten Ziffern wird die geheime Zahl gebildet und in der Variable **x** gespeichert.

Der **Tipp** wird auf 50 voreingestellt, genau in der Mitte des möglichen Wertebereichs der Zufallszahl. Der Zähler **n** für die Anzahl von Versuchen startet mit 0. Die Variable **ende** wird auf 0 gesetzt. Wurde die Zahl richtig erraten, wird sie auf 1 gesetzt, was die Endlosschleife beendet. Diese verwendet einen Block vom Typ **wiederhole bis ....** Die Schleife wird so lange wiederholt, bis die Zahl erraten wurde und damit das Spielende erreicht ist.

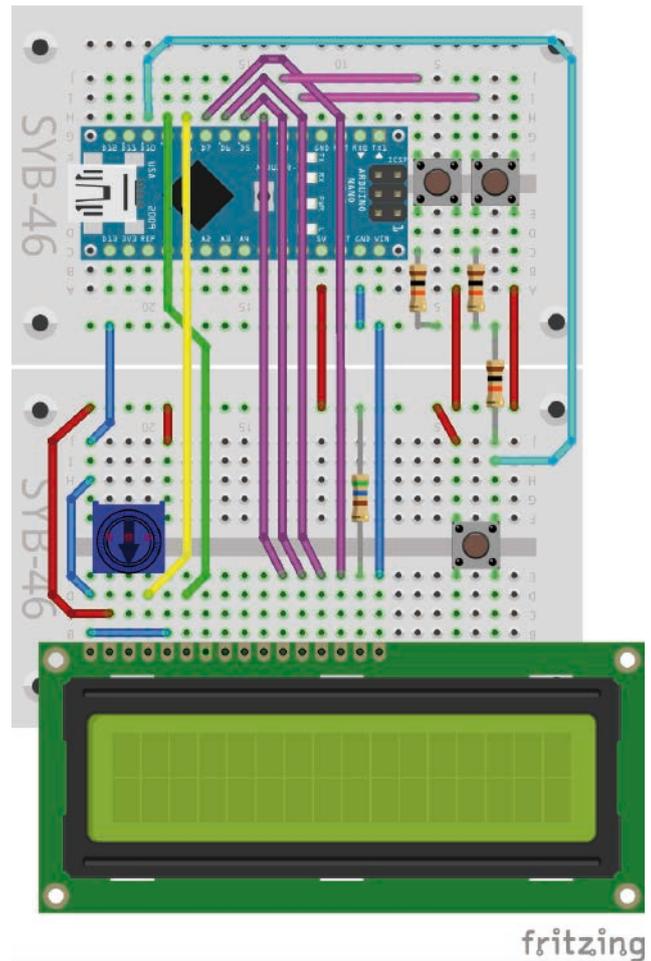
Solange nicht der untere Taster am Pin D10 gedrückt wird, kann der Benutzer über die anderen Taster die nächste getippte Zahl einstellen. Dazu werden nacheinander ständig die Taster abgefragt und ausgewertet. Der Benutzer kann so durch schnelles Drücken der Taster die gewünschte Zahl einstellen oder einen Taster länger gedrückt halten und damit schnell vor- oder zurückzählen.

Wurde der Taster am Pin **D2** gedrückt, wird die getippte Zahl um 1 erhöht. Sollte die Zahl dadurch größer als 100 werden, wird der aktuelle Tipp automatisch auf 100 gesetzt.

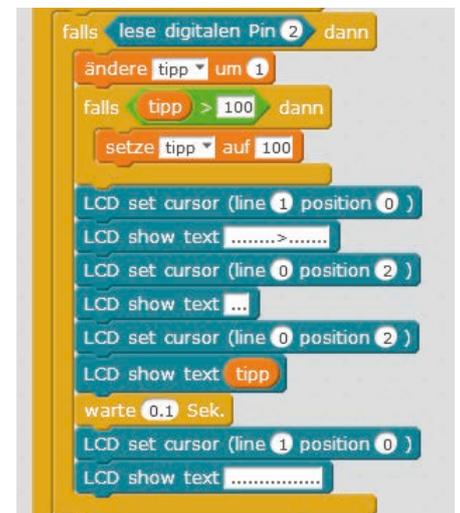
In der unteren Zeile des LCD-Moduls zeigt ein Pfeilsymbol nach rechts, dass die Zahl erhöht wurde.

In der oberen Zeile wird links die zuletzt gewählte Zahl mit drei Punkten überschrieben, damit keine Ziffer übrig bleibt, falls die neu gewählte Zahl eine Stelle weniger hat. Die Fläche mit Leerzeichen zu überschreiben funktioniert nicht, da mBlock 3 eine Zeichenfolge, die nur aus Leerzeichen besteht, im Arduino-Modus nicht korrekt auswertet. Danach wird die neue Zahl an dieser Stelle angezeigt.

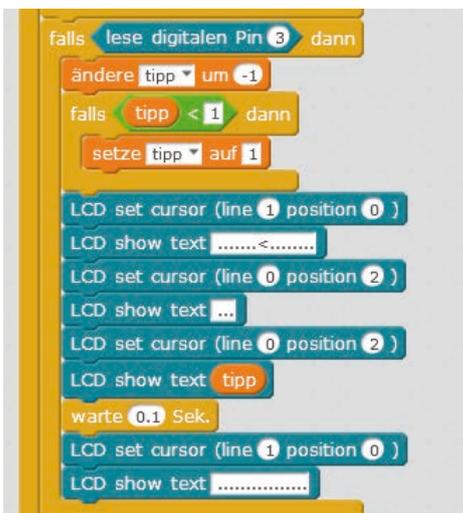
Nach einer Wartezeit von 0,1 Sekunden verschwindet das Pfeilsymbol in der unteren Zeile des LCD-Moduls. Jetzt kann der nächste Tastendruck ausgewertet werden.



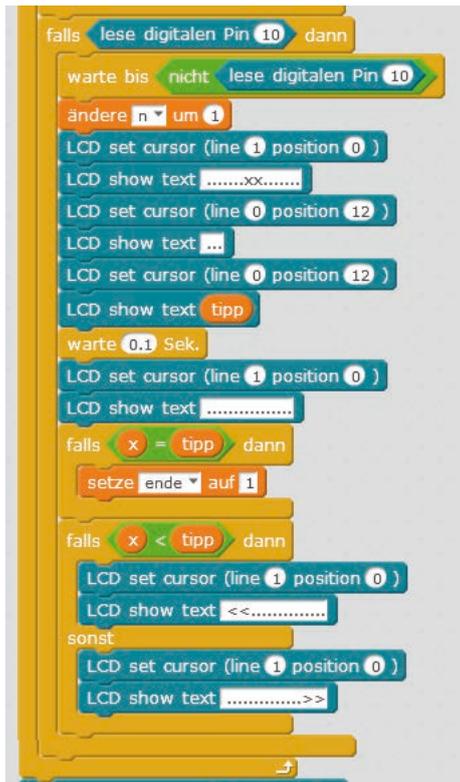
Ratespiel mit LCD-Modul und Tastern.



Der rechte Taster am Pin D2 erhöht die Zahl um 1.



Der linke Taster am Pin D3 verringert die Zahl um 1.



Der untere Taster am Pin D10 gibt den Tipp ab.

Wurde der Taster am Pin **D3** gedrückt, wird die getippte Zahl um 1 verringert. Sollte die Zahl dadurch kleiner als 1 werden, wird der aktuelle Tipp automatisch auf 1 gesetzt. Auch die weiteren Blöcke sind vergleichbar mit denen für den ersten Taster.

Wurde der Taster am Pin **D10** gedrückt, wird die eingestellte Zahl als Tipp abgegeben. Die Anzahl abgegebener Tipps in der Variable **n** wird um 1 erhöht und die Zahl als neuer Tipp rechts oben auf dem LCD-Modul angezeigt.

Falls die gesuchte Zahl **x** gleich dem abgegebenen Tipp ist, wird die Variable **ende** auf 1 gesetzt. Nach diesem Durchlauf wird die Schleife kein weiteres Mal wiederholt. Das Spiel ist zu Ende.

Ist die gesuchte Zahl **x** kleiner als der abgegebene Tipp, wird links unten ein Pfeil nach links angezeigt, ist sie größer, erscheint rechts unten ein Pfeil nach rechts. Damit erfährt der Spieler, in welcher Richtung er weiter tippen muss.

Ist das Spiel zu Ende, erscheint in der unteren Zeile des LCD-Moduls die Anzahl Versuche, die erforderlich waren, um die Zahl zu erraten.

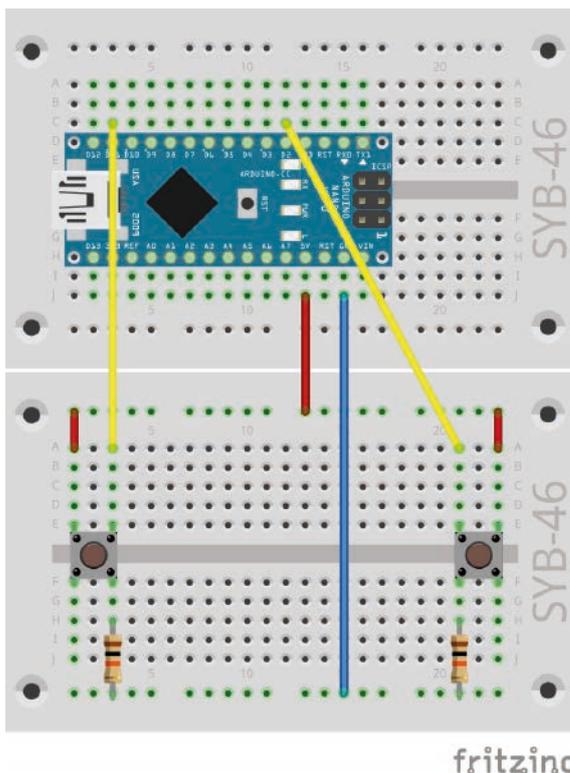
## 20. Tag

## 20. Tag

### Heute im Adventskalender

- 1x Verbindungskabel

Das Kabel wird erst an einem späteren Tag gebraucht.



Zwei Taster steuern das Pong-Spiel.

### Pong-Spiel auf dem PC mit Tasten steuern

Spiele und Animationen in Scratch werden üblicherweise mit Maus und Tastatur gesteuert, da Scratch standardmäßig nur diese Eingabemethoden unterstützt. mBlock 3 macht es möglich, Tasten und andere Eingabegeräte über den Arduino zur Steuerung von Spielen in Scratch zu verwenden. Das nächste Projekt zeigt, wie man in einem vorhandenen Scratch-Spiel die klassische Steuerung durch Arduino-Tasten ersetzt.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 2x 10-kOhm-Widerstand (Braun-Schwarz-Orange), 2x Taster, 4x Verbindungskabel, 2x Drahtbrücke

Die Taster sind so angeordnet, dass sie sich bequem mit zwei Fingern bedienen lassen.

### Das Programm

Die beiden Taster sollen das Panel eines klassischen Pong-Spiels in beide Richtungen bewegen. Das Original-Scratch liefert bei seinen Beispielen ein solches Spiel mit, mBlock 3 leider nicht. Sie finden das Programm als `20pong01mblock` im Downloadarchiv.

### Interaktiver Modus

Das Spiel läuft im interaktiven Modus von mBlock 3. Schalten Sie, wie am 10. Tag beschrieben, mBlock 3 und den Nano in den interaktiven Modus um.

### So funktioniert das Spiel

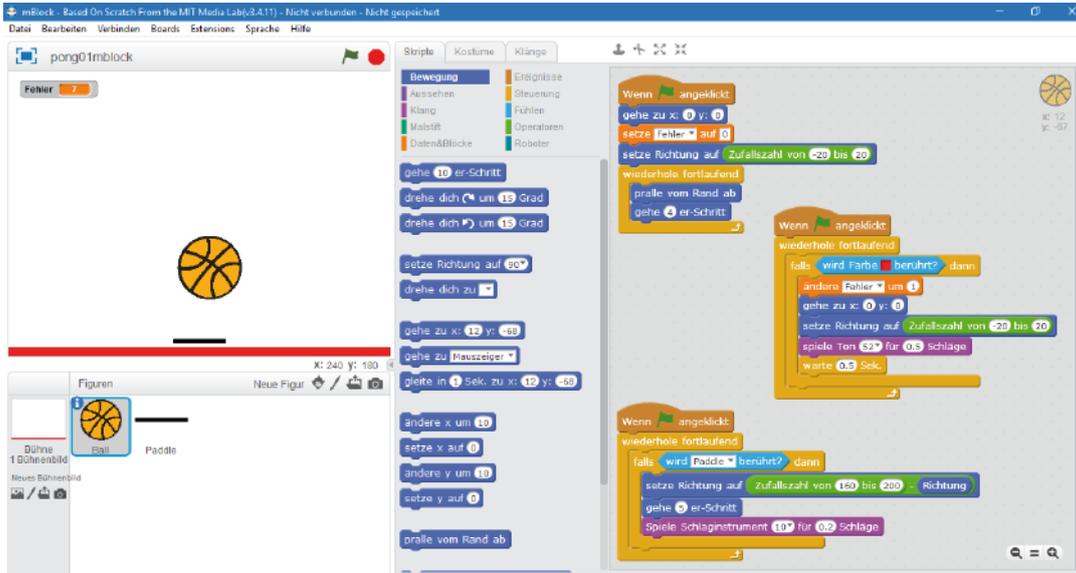
Bevor wir uns daran machen, die Steuerung des Spiels zu verändern, zunächst ein paar Grundlagen zur Funktionsweise des Spiels: Bei diesem Spieleklassiker, der

bereits in den 70er-Jahren auf frühen Computersystemen gespielt wurde, geht es darum, einen Ball möglichst lange mit einem simplen Schläger in der Luft zu halten. Das sogenannte Paddle lässt sich nur horizontal bewegen. Wenn der Ball kommt, muss der Schläger also an der richtigen Stelle sein.

Das Spiel besteht aus zwei Objekten, Ball und Paddle, von denen jedes - wie in Scratch üblich - eigene Programmblöcke verwendet, die unabhängig voneinander laufen.

## Der Ball

Der Ball wird durch drei Skripte gesteuert, die alle gleichzeitig laufen und gestartet werden, wenn der Benutzer auf das grüne Fähnchen klickt.



Das klassische Pong mit dem Skript für den Ball noch ohne Nano.

Die normale Ballbewegung wird über das ganz oben abgebildete Skript gesteuert. Es bringt zuerst den Ball an seine Startposition, setzt den Zähler für die Fehler auf 0 und legt eine zufällige Richtung für die erste Bewegung fest. Scratch liefert einen eigenen Block, der dafür sorgt, dass ein Objekt, das den Rand der Bühne berührt, abprallt und in der richtigen Richtung weiterfliegt. Befindet sich das Objekt nicht am Rand der Bühne, wird dieser Block ignoriert und ein einfacher Bewegungsschritt der Länge 4 in der voreingestellten Richtung ausgeführt.

Der zweite Skriptblock wartet darauf, bis der Ball die rote Farbe am unteren Rand der Bühne berührt. In diesem Fall ist er heruntergefallen, und der Spieler erhält einen Fehlerpunkt. Danach wird der Ball wieder an die Startposition gesetzt, eine neue zufällige Bewegungsrichtung festgelegt und ein kurzes Tonsignal abgespielt.

Der dritte Skriptblock prüft in einer Endlosschleife, ob der Ball das Paddle berührt. In diesem Fall wird die Bewegungsrichtung zufällig so geändert, dass sich der Ball vom Panel weg bewegt und dabei noch um einen zufälligen Winkel verdreht wird, damit er in eine nicht vorhersehbare Richtung weiterfliegt. Andernfalls könnte es passieren, dass der Ball durch Abprallen an den drei Seiten der Bühne und am Panel einen ewigen Viereckkurs fliegt, ohne dass der Benutzer etwas zu tun braucht. Zum Schluss wird ein kurzer Klang abgespielt.

## Das Paddle

Das Paddle wird, nachdem das grüne Fähnchen angeklickt wurde, an seine Startposition gesetzt. Zwei weitere Skriptblöcke fragen die Tasten [Pfeil nach links] und [Pfeil nach rechts] ab. Drückt der Benutzer eine dieser Tasten, wird das Paddle um 20 Einheiten in die entsprechende Richtung bewegt. Dabei wird anhand der Koordinaten geprüft, ob es über die Ränder links oder rechts herauslaufen würde, und in diesem Fall direkt an den Rand gesetzt.

Klicken Sie unten links auf der Objektpalette auf das Paddle, um die Skriptblöcke für das Paddle zu sehen und zu bearbeiten.

Sie können das Spiel so schon spielen. Steuern Sie es über die Pfeiltasten auf der Tastatur. Der Nano wird nicht gebraucht.



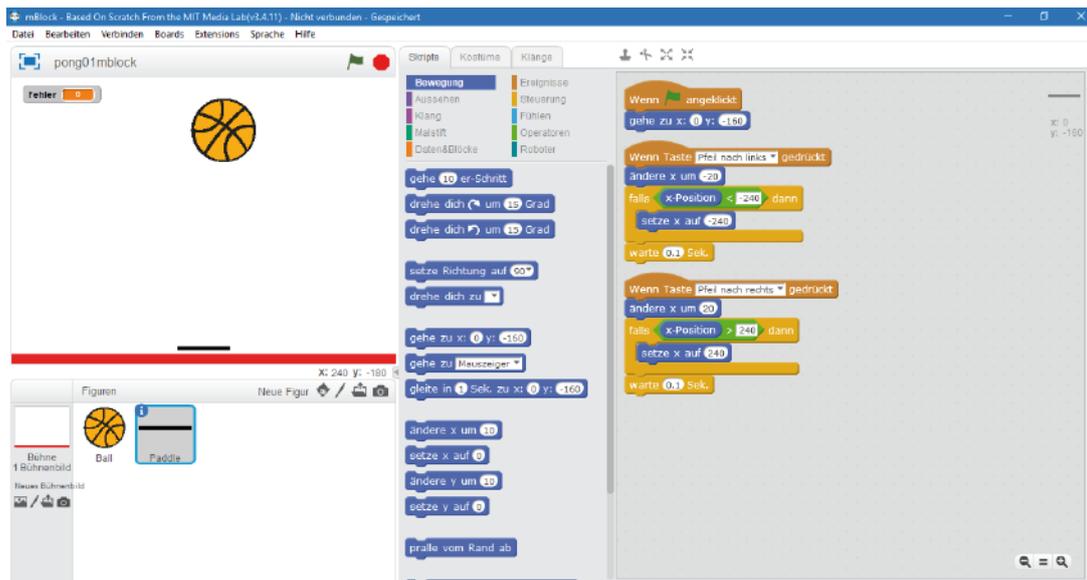
Die Blöcke für die normale Ballbewegung.



Die Blöcke für den Fall, dass der Ball die rote Linie berührt.



Die Blöcke für den Fall, dass der Ball vom Paddle zurückgeschlagen wird.



Das Skript für das Paddle - noch ohne Arduino.

## Pong mit Tastern am Nano steuern

In der nächsten Version soll das Paddle über die beiden Taster auf dem Steckbrett, die an den Pins **D11** und **D2** auf dem Nano angeschlossen sind, gesteuert werden.

mBlock 3 bietet auf der Blockpalette **Ereignisse** keinen zu **Wenn Taste ... gedrückt** vergleichbaren Block für den Fall, dass ein High-Signal an einem digitalen Pin anliegt. Das Programm kann also das Drücken eines der Taster nicht direkt als Ereignis auswerten. Also bauen wir das Programm so um, dass eine Endlosschleife abwechselnd die beiden Taster abfragt und auswertet.

Ziehen Sie zwei **falls... dann**-Blöcke in die Schleife. Jeder dieser Blöcke fragt über einen **lese digitalen Pin ...**-Block von der Blockpalette **Roboter** einen digitalen Pin auf dem Nano ab. Liefert der jeweilige Pin ein High-Signal, ist der Taster gedrückt. In diesem Fall werden die gleichen Blöcke zur Bewegung des Paddles abgearbeitet wie in der vorherigen Programmversion beim Drücken einer Pfeiltaste auf der Tastatur.

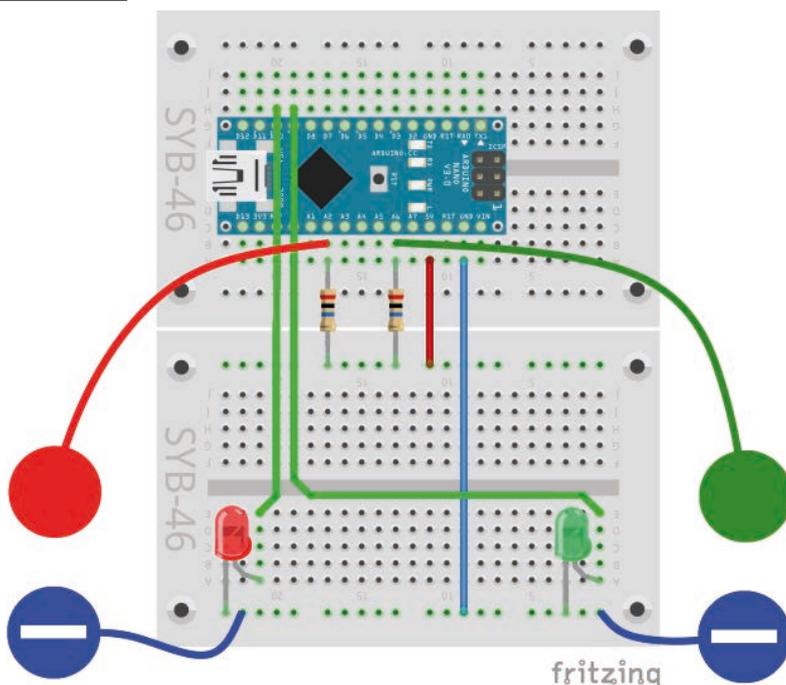
Klicken Sie jetzt auf das grüne Fähnchen, um das Spiel zu starten. Sie können das Paddle jetzt mit den beiden Tastern steuern. Sie finden das fertige Programm als `20pong02mblock` bei den Downloads.

## 21. Tag

### Heute im Adventskalender

- 1x Widerstand 20 MOhm (Rot-Schwarz-Blau)

21. Tag



Sensorkontakt und Massekontakt aus Knete.

## Tennisspiel mit Sensorkontakten steuern

Das heutige Spiel ist ein Tennisspiel für zwei Spieler, wie man es von Telespielen aus den 70er-Jahren kennt. Jeder Spieler bekommt zwei Sensorkontakte in die Hand, wobei der eine ein Massekontakt ist, und dazu eine LED, die die Berührung des Sensors anzeigt.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LCD-Modul, 1x 560-Ohm-Widerstand (Grün-Blau-Braun), 1x 20-MOhm-Widerstand (Rot-Schwarz-Blau), 1x Potentiometer, 2x Knete, 10x Verbindungskabel, 6x Drahtbrücke (unterschiedliche Längen)

### Das Programm

Um sofort loszuspielen, öffnen Sie in mBlock 3 das Programm `21mblock` aus den Downloads. Sie können das Spiel natürlich auch, um jeden Schritt nachzuvollziehen, anhand der Abbildungen selbst erstellen.

### Interaktiver Modus

Das Spiel läuft im interaktiven Modus von mBlock 3. Schalten Sie wie am 10. Tag beschrieben mBlock 3 und den Nano in den interaktiven Modus um.

Ein Klick auf das blaue Rechtecksymbol oben links neben dem Programmnamen schaltet die Bühne in den Vollbildmodus, damit auf dem ganzen Bildschirm spielen kann.

Klicken Sie auf das grüne Fähnchen, startet das Spiel und der Ball fliegt durch den Raum. Jeder Spieler kann sein Paddle durch Antippen der Sensortasten nach oben bewegen. Berührt man den Sensorkontakt eine Weile nicht, fällt das Paddle langsam wieder nach unten. Mit etwas Geschicklichkeit müssen die Spieler versuchen, mit ihrem Paddle den Ball immer wieder zurückzuschlagen, damit er nicht gegen die rote bzw. die grüne Wand fliegt.

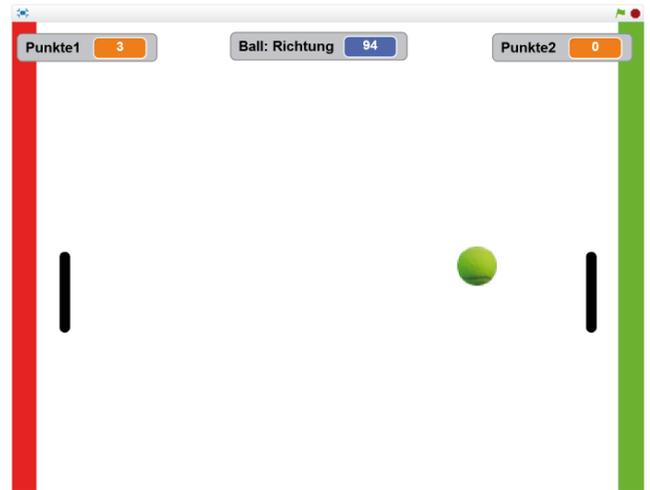
Berührt der Ball die eigene Wand, weil man ihm das Paddle nicht rechtzeitig in den Weg gesetzt hat, bekommt man einen Minuspunkt. Der Ball startet dann wieder an der Ausgangsposition. Durch Klicken auf das rote Stopp-Symbol lässt sich das Spiel jederzeit anhalten. Wer weniger Minuspunkte hat, hat gewonnen.

### So funktioniert das Programm

Im Spiel gibt es drei Objekte: den Ball, der das eigentliche Spiel steuert und der auch die Minuspunkte vergibt, wenn er an die farbigen Ränder stößt, sowie die Paddles für die beiden Spieler links und rechts. Die Paddles werfen auch die Kontaktsensoren aus und schalten die LEDs.

Die beiden farbigen Streifen an den Rändern sind direkt auf die Bühne gemalt.

Wenn Sie sie in der Objektpalette links anklicken, können Sie auf der Registerkarte **Bühnenbilder** (ganz oben) den Hintergrund mit dem in mBlock 3 integrierten Malprogramm bearbeiten.



Das Tennisspiel im Vollbildmodus.

### Der Ball

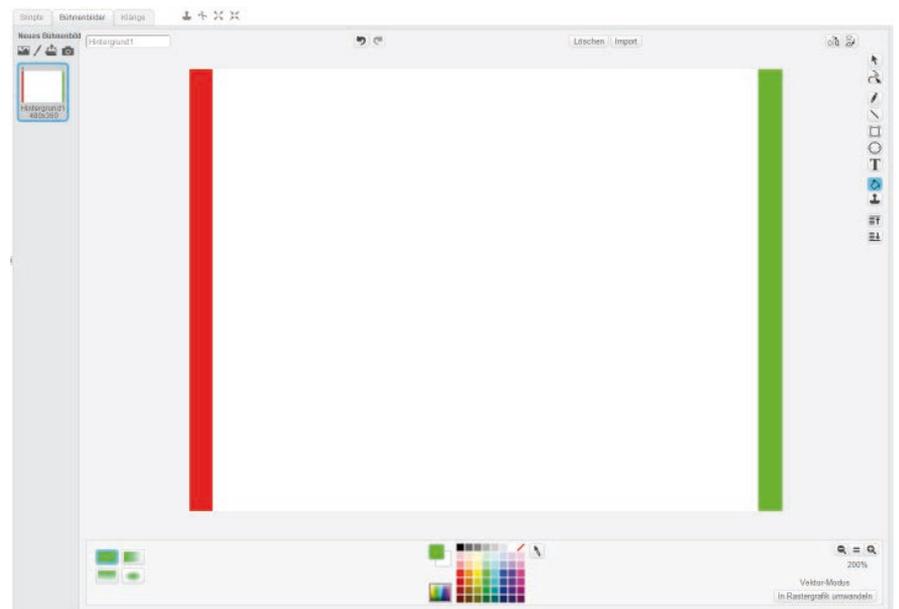
Der Ball ist ein Tennisball aus der Figurenbibliothek, die mBlock 3 mitliefert. Klicken Sie auf der Objektpalette auf das Symbol **Figur aus der Bibliothek wählen**. Hier finden Sie den **Tennisball** bei den Themen unter **Sport**.



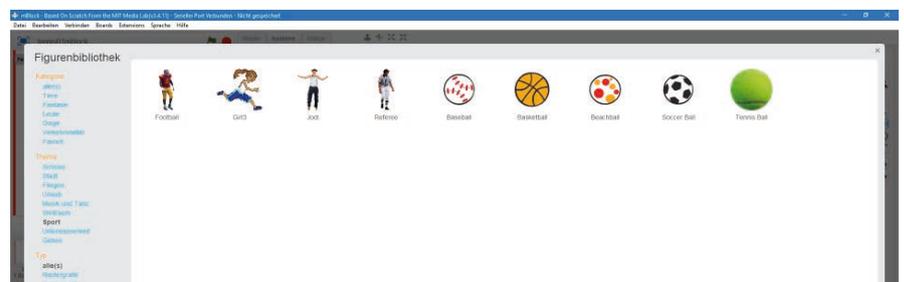
Neue Figur auf der Objektpalette aus der Bibliothek wählen.

### Die Paddles

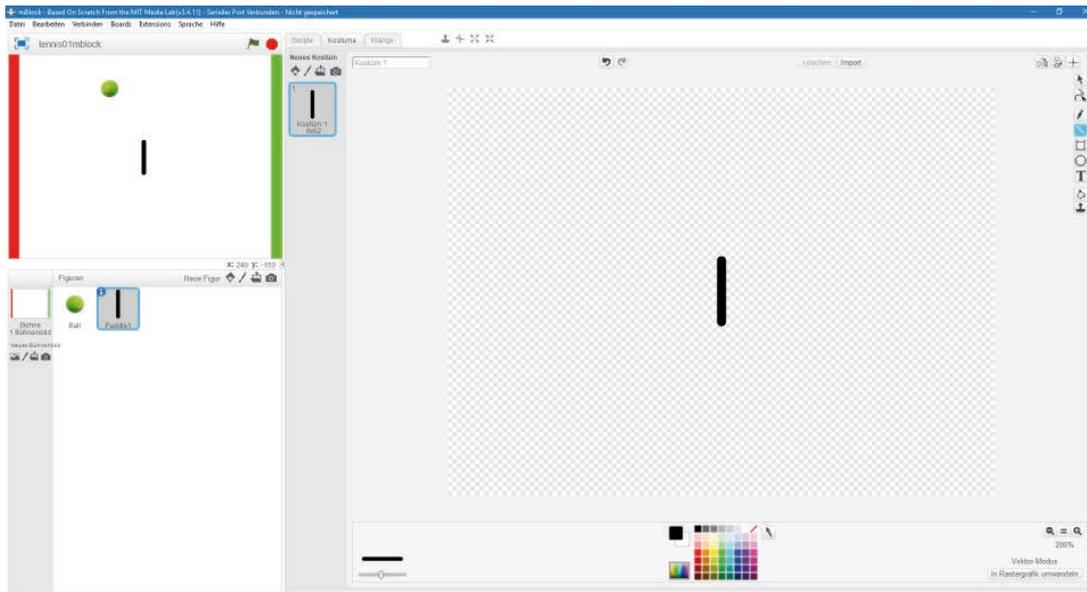
Klicken Sie auf der Objektpalette auf das Symbol **Neue Figur zeichnen**, um das erste Paddle zu zeichnen. Verwenden Sie im Malprogramm das Werkzeug **Linie**, wählen Sie die schwarze Farbe aus und stellen Sie unten links eine mittlere Linienbreite ein. Malen Sie die Linie mittig senkrecht über das kleine Kreuz in der Mitte, damit der Mittelpunkt des Paddles auf dem Koordinatennullpunkt liegt. Bei gedrückter [Umschalt]-Taste wird die Linie genau senkrecht.



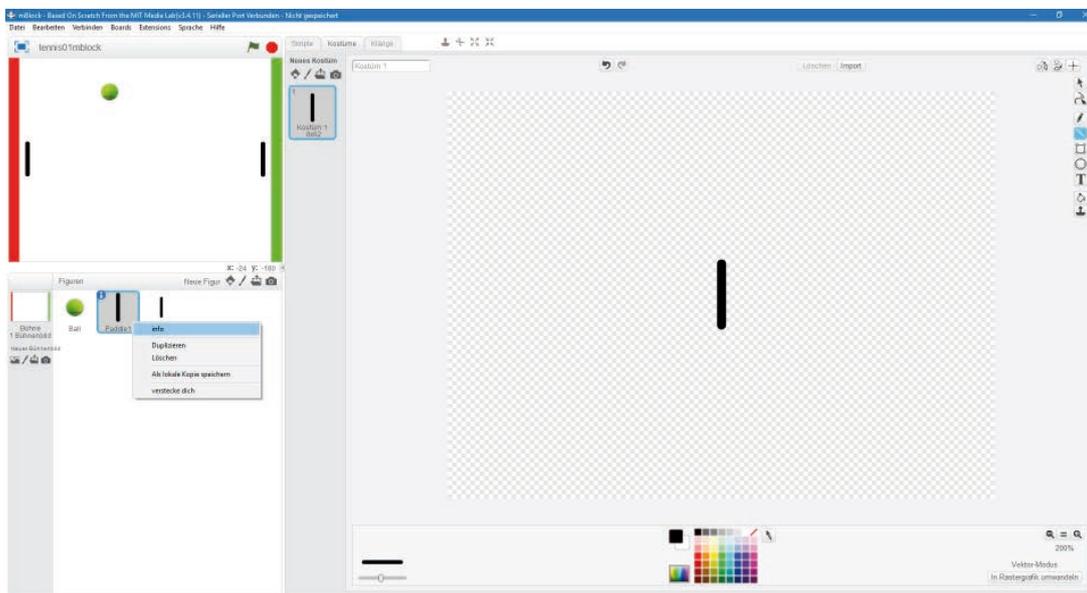
Der Hintergrund für das Tennisspiel im Malprogramm von mBlock 3.



Das Thema Sport in der Figurenbibliothek.



Das erste Paddle.



Die beiden Paddles.

Klicken Sie dann auf das blaue Info-Symbol bei der neuen Figur auf der Objektpalette und geben Sie dem Paddle den Namen **Paddle1**.

Klicken Sie mit der rechten Maustaste auf das **Paddle1** auf der Objektpalette und wählen Sie im Menü **Duplizieren**. Schieben Sie anschließend das **Paddle1** in die Mitte kurz vor dem roten Rand links und das **Paddle2** in die Mitte kurz vor dem grünen Rand rechts. Die genauen Positionen werden später vom Programm eingestellt.

### Programm für den Ball

Der Ball wird durch vier Skripte gesteuert, die alle gleichzeitig laufen und gestartet werden, wenn der Benutzer auf das grüne Fähnchen klickt.

Zunächst werden zwei Variablen definiert, die auf der Bühne in Echtzeit angezeigt werden. Die Anzeige für die Punkte des rechten Spielers schieben Sie auf der Bühne nach rechts außen.

- Punkte1** Minuspunkte für Spieler1 (links)
- Punkte2** Minuspunkte für Spieler2 (rechts)

Das Hauptskript schafft beim Klick auf das grüne Fähnchen die Grundvoraussetzungen für das Spiel. Zuerst wird der Ball

auf seine Ausgangsposition bei **x:0 y:0** gebracht. Anschließend werden die beiden Punktezähler auf ihre Ausgangswerte gesetzt. Beide Spieler haben **0** Punkte.

Der Ball soll in einem zufälligen Winkel losfliegen. Dazu wird die Richtung auf einen zufälligen Wert zwischen **-20** und **-160** gesetzt.



Die Skripte und Variablen für den Ball

Anschließend wird die Bewegung des Balls fortlaufend wiederholt. Er prallt vom Rand ab, sollte er ihn berühren. Andernfalls fliegt er einen Vierer-Schritt in die eingestellte Richtung. Diese Bewegung wiederholt sich theoretisch endlos. Da aber beim Klicken auf das grüne Fähnchen drei weitere Skriptblöcke für den Ball gestartet werden, kann es auch noch zu anderen Bewegungen kommen.

Falls der Ball eines der beiden Paddles berührt, wird die Bewegungsrichtung ins Negative umgekehrt. Der Ball fliegt im gleichen Winkel nach unten weiter, in dem er von oben kam. Die aktuelle Richtung kann über den Block **Richtung** aus der Blockpalette **Bewegung** wie eine Variable verwendet werden. Um die Bewegung etwas unvorhersehbarer zu gestalten, wird der Ball zunächst einen kleinen Fünfer-Schritt bewegt, damit er danach das Paddle nach Möglichkeit nicht mehr berührt. Anschließend wird die Flugrichtung gegenüber der bisherigen Richtung um einen zufälligen Wert zwischen **-20** und **+20** Grad verändert. Zum Schluss ertönt ein kurzer Ton auf dem PC-Lautsprecher.

Berührt der Ball nicht das Paddle, sondern den roten Balken am linken Rand, bekommt der linke Spieler einen Minuspunkt. Um die Farbe im Block **wird Farbe ... berührt** auszuwählen, tippen Sie in das Farbfeld des Blocks. Eine Pipette erscheint, mit der Sie auf der Bühne die gewünschte Farbe auswählen können.

Der Ball wird wieder an seine Ausgangsposition gesetzt und um einen zufälligen Wert zwischen **-20** und **+20** Grad gedreht, damit er nicht wieder exakt die gleiche Flugbahn nimmt, aber trotzdem ungefähr in die Richtung fliegt, in die er zuletzt geflogen ist. Wieder ertönt ein Ton - diesmal ein anderer, schrillerer.

Das Gleiche passiert, wenn der Ball den grünen Balken am rechten Rand berührt, nur mit dem Unterschied, dass diesmal der rechte Spieler den Minuspunkt bekommt.

### Programm für die Paddles

Die Paddles werden bei Berührung des Sensorkontaktes schrittweise nach oben bewegt. Lässt man den Sensorkontakt los, fällt das Paddle wieder in kleineren Schritten nach unten.

Die Sensorkontakte sind wieder an zwei analogen Eingängen angeschlossen. Liefert der analoge Eingang einen Wert kleiner als 200, geht das Programm davon aus, dass der Sensorkontakt berührt wurde. Sollte es vermehrt zu Fehlreaktionen kommen, probieren Sie kleinere oder größere Grenzwerte aus.

Bei Klick auf das grüne Fähnchen wird bei jedem der beiden Paddles ein ähnlicher Skriptblock gestartet. Diese setzen zuerst das Paddle auf die Startposition. Danach laufen Endlosschleifen. In der Endlosschleife liegt ein **falls/sonst**-Konstrukt, das jedes Mal eine von zwei möglichen Aktionen auslöst:

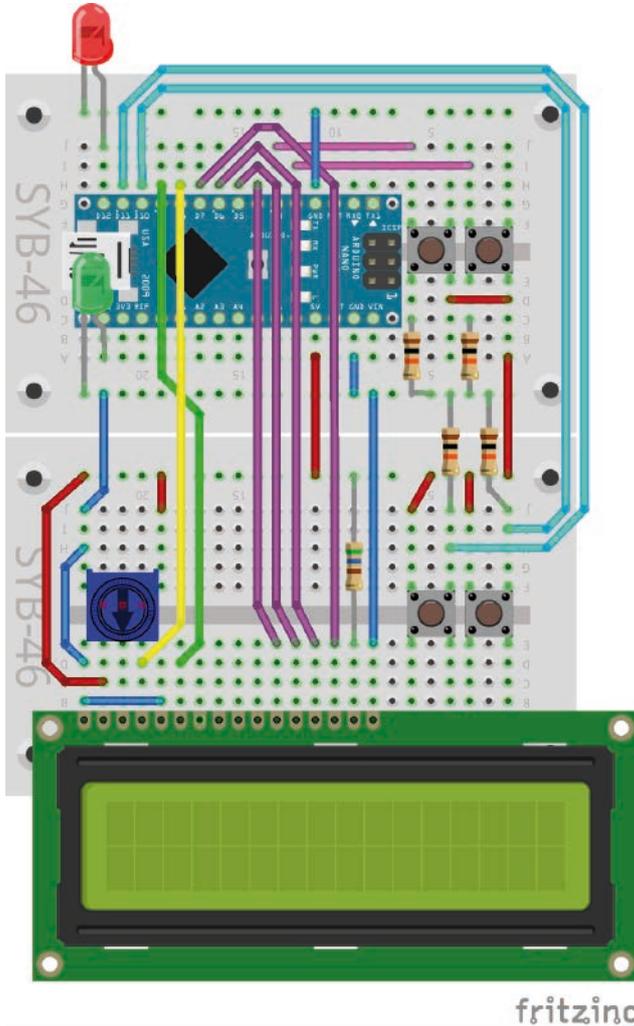
- Berührt der Spieler den Sensorkontakt, wird der Pin auf LOW gesetzt, verhält sich also wie ein nicht gedrückter Taster. Die y-Koordinate des Paddles wird um 40 erhöht, sodass es sich um 40 Einheiten nach oben schiebt. Die LED leuchtet für 0,1 Sekunden.
- Solange der Spieler den Sensorkontakt nicht berührt, ist die Abfragebedingung nicht erfüllt und die unter **sonst** aufgelistete Anweisung wird ausgeführt. Die y-Koordinate des Paddles wird um 20 Einheiten verringert, das Paddle rutscht um 20 Einheiten nach unten. Dieser Wert ist bewusst kleiner gewählt, damit das Paddle nicht so schnell fällt, aber in großen Schritten wieder schnell nach oben bewegt werden kann.

Das linke **Paddle1** verwendet den Sensorkontakt am Pin A2 und die LED am Pin D10. Das rechte **Paddle2** verwendet den Sensorkontakt am Pin A6 und die LED am Pin D9.

## 22. Tag

## Heute im Adventskalender

- 1x Taster
- 1x Widerstand 10 kOhm (Braun-Schwarz-Orange)



Reaktionsspiel mit LCD-Modul, zwei LEDs und vier Tastern.

## Reaktionsspiel

In diesem Reaktionsspiel muss man, wenn die rote LED leuchtet, möglichst schnell die richtige Taste drücken.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LCD Modul, 1x 560-Ohm-Widerstand (Grün-Blau-Braun), 4x 10-kOhm-Widerstand (Braun-Schwarz-Orange), 4x Taster, 1x Potentiometer, 12x Verbindungskabel, 11x Drahtbrücke (unterschiedliche Längen)

Das Spiel läuft eigenständig über den Arduino-Modus von mBlock 3. Es kann also mit dem Batteriekasten auch unabhängig versorgt werden.

## Das Programm

Das Programm `22mblock` verwendet die Stoppuhr auf dem Nano, um die Reaktionszeit des Spielers zu messen.

```

Arduino Programm
setze digitalen Pin 12 Ausgang auf LOW
setze digitalen Pin 13 Ausgang auf LOW
wiederhole fortlaufend
  setze x auf lese analogen Pin (A) 0 mod 4 + 1
  LCD clear
  LCD show text Taste:
  LCD set cursor (line 0 position 7 )
  LCD show text x
  setze digitalen Pin 12 Ausgang auf HIGH
  setze Stoppuhr zurück
  wiederhole bis a = x
    falls lese digitalen Pin 3 dann
      warte bis nicht lese digitalen Pin 3
      setze a auf 1
    falls lese digitalen Pin 2 dann
      warte bis nicht lese digitalen Pin 2
      setze a auf 2
    falls lese digitalen Pin 11 dann
      warte bis nicht lese digitalen Pin 11
      setze a auf 3
    falls lese digitalen Pin 10 dann
      warte bis nicht lese digitalen Pin 10
      setze a auf 4
  LCD set cursor (line 1 position 0 )
  LCD show text Stoppuhr
  warte 0.05 Sek.
  setze digitalen Pin 12 Ausgang auf LOW
  setze digitalen Pin 13 Ausgang auf HIGH
  LCD set cursor (line 0 position 0 )
  LCD show text Deine Zeit:
  warte 1 Sek.
  setze digitalen Pin 13 Ausgang auf LOW
  
```

Das Programm `22mblock` steuert das Reaktionsspiel.

## So funktioniert das Programm

Am Anfang jedes Durchlaufs der Endlosschleife wird wieder mit Hilfe eines nicht beschalteten analogen Eingangs ein Zufallswert zwischen 1 und 4 erzeugt. Diese Zahl wird in der oberen Zeile des LCD-Moduls angezeigt. Danach wird die rote LED am Pin 12 eingeschaltet und die Stoppuhr zurückgesetzt.

### Die Stoppuhr auf dem Nano

Alle Arduino-kompatiblen Platinen haben eine interne Uhr, die für Aufgaben verwendet werden kann, die bestimmte Zeiträume erfordern. Diese Uhr zeigt keine wirkliche Uhrzeit, sondern einfach die seit dem Programmstart verstrichene Zeit. Ein Programm kann die Uhr jederzeit zurücksetzen. Sie startet dann automatisch wieder neu. Auf diese Weise lassen sich Zeiträume messen.

Es startet eine Schleife, die so lange läuft, bis die Taste gedrückt wurde, die der angezeigten Zahl entspricht.

Dazu werden nacheinander die vier Tasten abgefragt. Damit das Spiel nicht reagiert, wenn der Benutzer alle Tasten gleichzeitig drückt, wird die Variable **a**, die die Nummer der gedrückten Taste enthält, erst gesetzt, nachdem die Taste wieder losgelassen wurde.

In jedem Schleifendurchlauf wird in der unteren Zeile des LCD-Moduls der aktuelle Wert der Stoppuhr angezeigt.

Hat der Spieler die richtige Taste gedrückt, wird die rote LED aus- und die grüne eingeschaltet. Die untere Zeile des LCD-Moduls zeigt die gestoppte Zeit an. Nach einer Sekunde startet das Programm wieder neu.

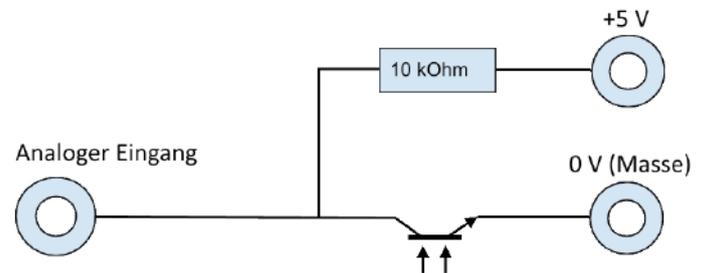
## 23. Tag

### Heute im Adventskalender

- 1x Fototransistor

### Fototransistor

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



Schaltschema für einen Fototransistor

### Weihnachtliche Lichteffekte in der Dunkelheit

Das Experiment des 23. Tages lässt LEDs bunt blinken, wenn es dunkel genug ist. Schließen Sie nach der Programmierung noch den Batteriekasten an und tragen Sie die Schaltung in eine dunkle Ecke. Dann beginnen die LEDs zu blinken. Eine Pegelanzeige auf dem LCD-Modul zeigt die aktuelle Beleuchtungsstärke.

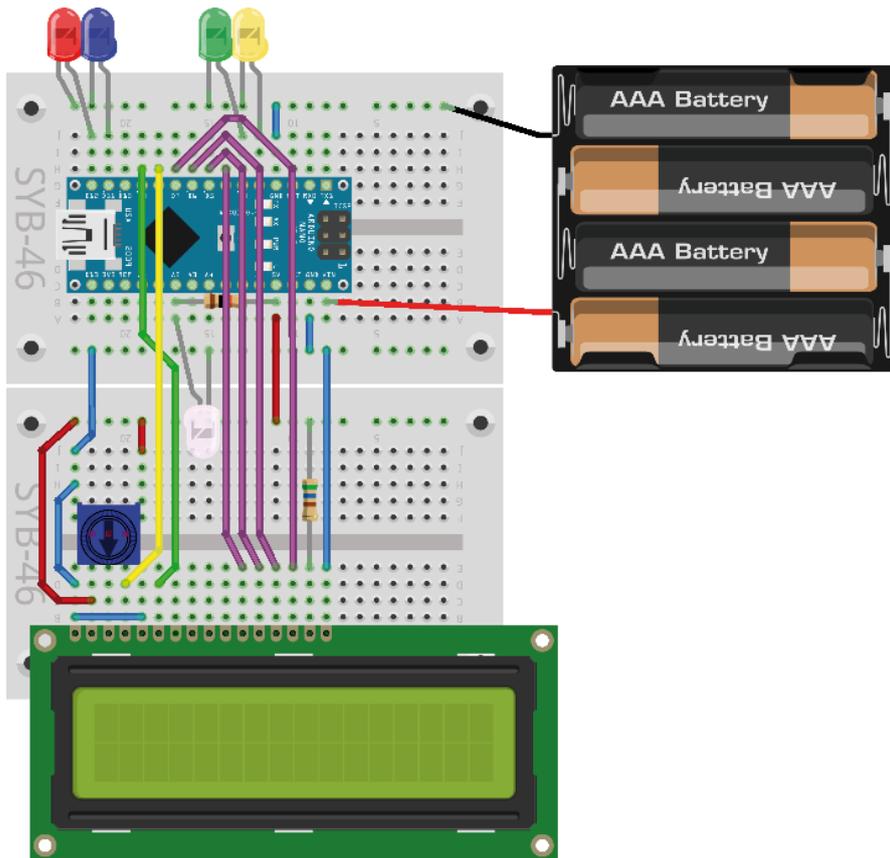
**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 1x LCD-Modul, 1x 560-Ohm-Widerstand (Grün-Blau-Braun), 1x 10-kOhm-Widerstand (Braun-Schwarz-Orange), 1x Fototransistor, 1x Potentiometer, 1x Batteriekasten, 8x Verbindungskabel, 7x Drahtbrücke (unterschiedliche Längen)

### Das Programm

Das Programm `23mb1ock` zeigt die Helligkeit, die auf den Fototransistor fällt, als Balken auf dem LCD-Modul an. Übertragen Sie das Programm im Arduino-Modus von mBlock 3 auf den Nano. Es kann danach mit dem Batteriekasten auch unabhängig versorgt werden.

### So funktioniert das Programm

Der Fototransistor liefert je nach Helligkeit einen analogen Wert zwischen 0 und 1023, der umso höher ist, je weniger Licht auf den Fototransistor fällt. Dieser Wert wird durch 64 geteilt, um 16 mögliche Stufen für die Balkengrafik zu erhalten. Damit helleres Licht einen längeren Balken erzeugt, wird in diesem Fall ein höherer Wert gebraucht als bei dunklem Licht. Deshalb wird der errechnete Wert von 15 subtrahiert und in der Variablen **a** gespeichert.



LCD-Modul, vier LEDs und Fototransistor.

fritzing

Der Zähler  $n$  beginnt bei 0. Jetzt startet wieder eine Schleife, die für die 16 Stellen des LCD-Moduls 16-mal durchläuft. Bei jeder Stelle wird geprüft, ob die Nummer  $n$  größer ist als der in  $a$  gespeicherte Pegelwert. Ist das der Fall, wird die entsprechende Stelle auf beiden Zeilen ausgeschaltet. Dazu wird an dieser Stelle eine Folge von Leerzeichen angezeigt, die auch Sternchen weiter rechts, die von früher noch dort stehen, überschreibt. Da Zeichenketten, die nur aus Leerzeichen bestehen, in mBlock 3 nicht korrekt auf den Nano übertragen werden, enthält die Zeichenfolge an der letzten Stelle, die wegen der Länge des LCD-Moduls nie angezeigt wird, einen Punkt.

Ist die Nummer  $n$  nicht größer, also kleiner oder gleich dem Pegelwert, wird an der Stelle auf dem LCD-Modul ein Sternchen angezeigt. Auf diese Weise werden immer alle Stellen geprüft, unabhängig davon, ob der Helligkeitswert gegenüber der letzten Messung höher oder niedriger ist.

Ist der Wert  $a$  kleiner als 1, fällt also nur minimales Licht auf den Fototransistor, blinken die vier LEDs abwechselnd paarweise kurz auf. Da sofort danach der nächste Schleifendurchlauf beginnt, erscheint bei Dunkelheit ein scheinbar gleichmäßiges Blinken, obwohl jede LED nach der Messung nur einmal ein- und wieder ausgeschaltet wird.

## 24. Tag

## 24. Tag

## Heute im Adventskalender

- 1x Downloadcode

## Weihnachtsjukebox

Das Programm des 24. Tages spielt beim Drücken einer Taste ein Weihnachtslied. Vier verschiedene Lieder stehen zur Auswahl. Über den Downloadcode im Adventskalender heute können Sie sich Weihnachtslieder im mp3-Format herunterladen.

**Bauteile:** 1x Nano-Board, 2x Steckbrett, 1x LED rot mit Vorwiderstand, 1x LED gelb mit Vorwiderstand, 1x LED grün mit Vorwiderstand, 1x LED blau mit Vorwiderstand, 4x 10-kOhm-Widerstand (Braun-Schwarz-Orange), 4x Taster, 8x Verbindungskabel, 6x Drahtbrücke (unterschiedliche Längen)

## Das Programm

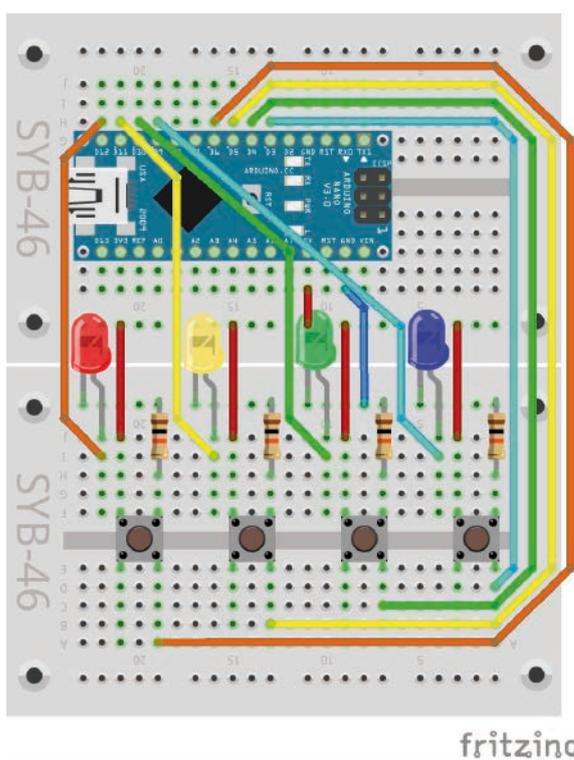
Das Programm `24mblock` spielt vier verschiedene Weihnachtslieder ab, je nachdem, welche Taste der Benutzer drückt. Zusätzlich leuchtet die passende LED.

## So funktioniert das Programm

Am Anfang werden Variablen für die Pinnummern der vier LEDs und Taster angelegt und alle LEDs ausgeschaltet.

Anschließend wartet eine Endlosschleife darauf, dass der Benutzer eine der Tasten drückt. Dazu laufen nacheinander vier gleich aufgebaute Abfragen, für jede Taste eine.

Erkennt das Programm, dass eine Taste gedrückt wurde, wird zunächst gewartet, bis diese Taste wieder losgelassen wird. Damit werden Fehlfunktionen durch längeres Drücken vermieden.



fritzing

Musikautomat mit Weihnachtsliedern.

Die LED wird eingeschaltet und eines der Lieder abgespielt. Danach wird die LED wieder ausgeschaltet.

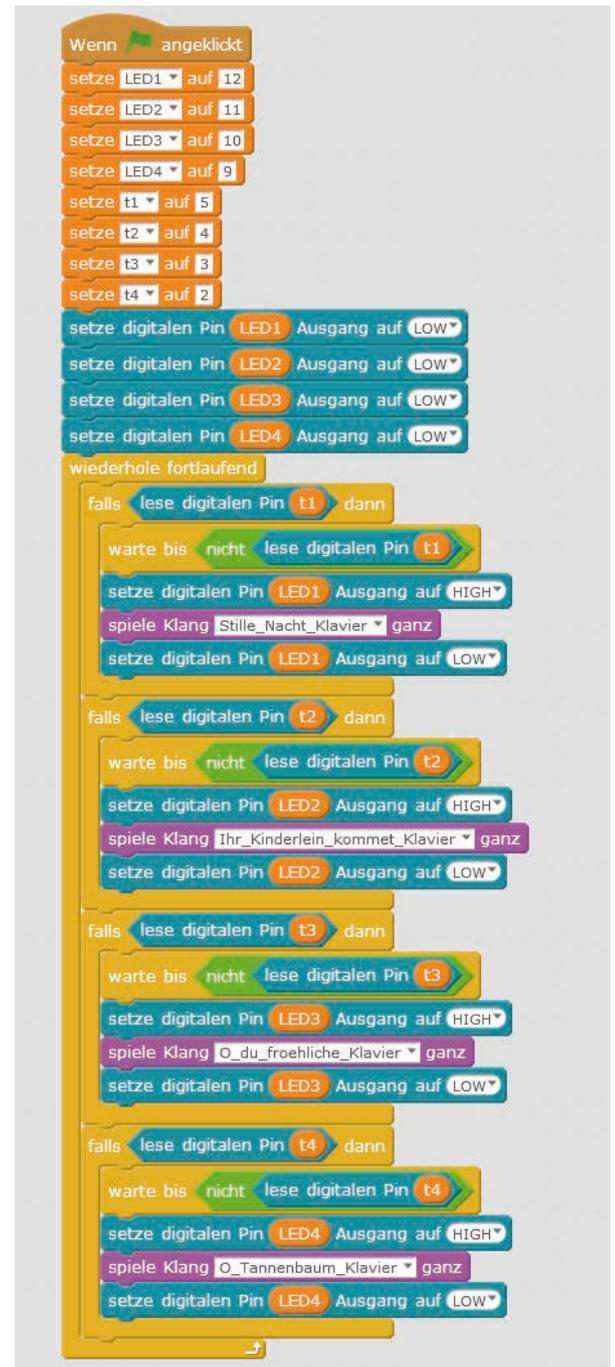
Die Weihnachtslieder sind im Programm gespeichert. Auf der Registerkarte **Klänge** können Sie mit dem Symbol **Klang aus einer Datei laden** MP3-Dateien importieren. Jeder importierte Klang bekommt automatisch einen Namen, der jederzeit verändert werden kann. Unter diesen Namen erscheinen die Klänge in der Auswahlliste im Block **spiele Klang ... ganz**.

Der Downloadcode heute im Adventskalender liefert einige weitere Weihnachtslieder im MP3-Format. Damit können Sie die im Programm vorinstallierten Lieder nach persönlichem Geschmack austauschen.

### We wish you a merry Christmas

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

Frohe Weihnachten!



Das Programm für die Weihnachtslieder.



Die Weihnachtslieder sind im Programm gespeichert.

### Vorsichtsmaßnahmen

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

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

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

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

### Warnung! Augenschutz und LEDs

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

### Liebe Kunden!



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

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



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

© 2019 Franzis Verlag GmbH, Richard-Reitzner-Allee 2, 85540 Haar

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

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

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

Arduino ist ein eingetragenes Markenzeichen der Arduino AG