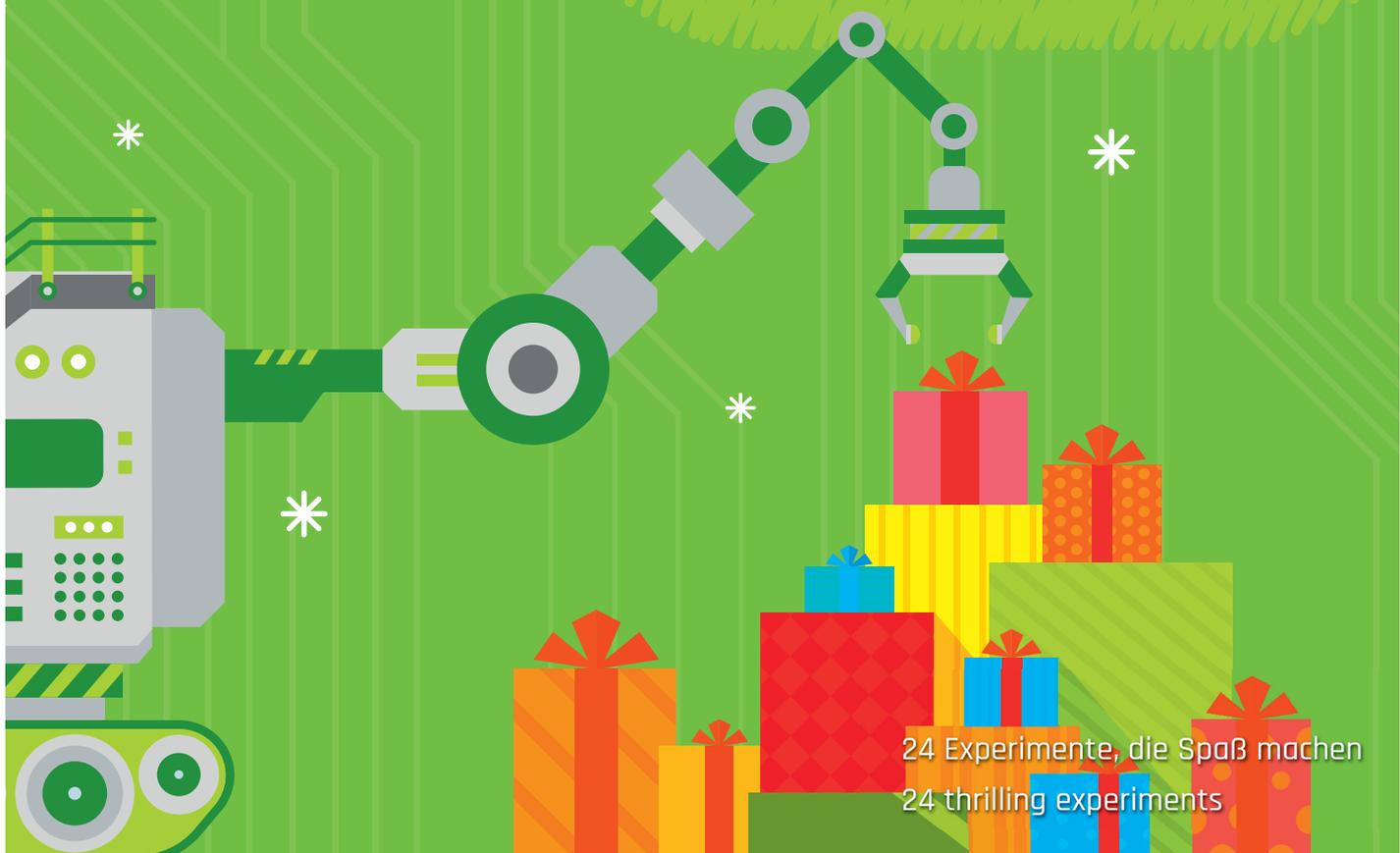




# ADVENTSKALENDER FÜR MICRO:BIT

COMPATIBLE WITH  
BBC  
micro:bit

# ADVENT CALENDAR FOR MICRO:BIT



## Alle Versuche im Überblick

<b>micro:bit-Adventskalender 2019</b> .....	<b>3</b>	<b>13. Tag</b> .....	<b>19</b>
<b>1. Tag</b> .....	<b>4</b>	Heute im Adventskalender .....	19
Heute im Adventskalender .....	4	Analoge Pegelanzeige auf der LED-Matrix .....	19
Den micro:bit vorbereiten .....	4	Das Programm .....	19
Der Makecode-Editor .....	4	So funktioniert das Programm .....	19
Frohe Weihnachten .....	5	<b>14. Tag</b> .....	<b>20</b>
Das Programm .....	5	Heute im Adventskalender .....	20
So funktioniert das Programm .....	5	Taster mit Pull-down-Widerstand .....	20
Programm auf den micro:bit übertragen .....	6	Das Programm .....	21
<b>2. Tag</b> .....	<b>7</b>	<b>15. Tag</b> .....	<b>22</b>
Heute im Adventskalender .....	7	Heute im Adventskalender .....	22
Frohe Weihnachten .....	7	Wechselblinklicht mit einstellbarer Geschwindigkeit .....	22
Das Programm .....	7	Das Programm .....	22
So funktioniert das Programm .....	7	So funktioniert das Programm .....	22
<b>3. Tag</b> .....	<b>8</b>	<b>16. Tag</b> .....	<b>23</b>
Heute im Adventskalender .....	8	Heute im Adventskalender .....	23
Zahlenwürfel mit Bananen oder Äpfeln .....	8	Bunte Blinkmuster mit vier LEDs .....	23
Das Programm .....	8	Das Programm .....	23
<b>4. Tag</b> .....	<b>9</b>	So funktioniert das Programm .....	23
Heute im Adventskalender .....	9	<b>17. Tag</b> .....	<b>24</b>
Echter Würfel mit Würfelsymbolen .....	9	Heute im Adventskalender .....	24
Das Programm .....	9	Reaktionsspiel .....	24
So funktioniert das Programm .....	9	Das Programm .....	24
<b>5. Tag</b> .....	<b>11</b>	So funktioniert das Programm .....	24
Heute im Adventskalender .....	11	<b>18. Tag</b> .....	<b>28</b>
Geschicklichkeitsspiel .....	11	Heute im Adventskalender .....	28
Das Programm .....	11	Spieleprogrammierung mit Sprites .....	28
So funktioniert das Programm .....	11	Das Programm .....	28
<b>6. Tag</b> .....	<b>12</b>	So funktioniert das Programm .....	29
Heute im Adventskalender .....	12	<b>19. Tag</b> .....	<b>30</b>
LED blinkt .....	12	Heute im Adventskalender .....	30
Das Programm .....	12	Wasserstandssensor .....	30
So funktioniert das Programm .....	12	Das Programm .....	30
<b>7. Tag</b> .....	<b>13</b>	So funktioniert das Programm .....	31
Heute im Adventskalender .....	13	<b>20. Tag</b> .....	<b>32</b>
LED mit Tastern ein- und ausschalten .....	13	Heute im Adventskalender .....	32
Das Programm .....	13	Nachtlicht in der Dunkelheit .....	32
So funktioniert das Programm .....	13	Das erste Programm .....	32
<b>8. Tag</b> .....	<b>14</b>	So funktioniert das erste Programm .....	33
Heute im Adventskalender .....	14	Das zweite Programm .....	33
LED blinkt auf Berührung .....	14	So funktioniert das zweite Programm .....	33
Das Programm .....	14	<b>21. Tag</b> .....	<b>34</b>
So funktioniert das Programm .....	14	Heute im Adventskalender .....	34
<b>9. Tag</b> .....	<b>15</b>	Lauflicht auf der LED-Matrix .....	34
Heute im Adventskalender .....	15	Das Programm .....	34
LEDs mit einer Taste abwechselnd umschalten .....	15	So funktioniert das Programm .....	34
Das Programm .....	15	<b>22. Tag</b> .....	<b>36</b>
So funktioniert das Programm .....	15	Heute im Adventskalender .....	36
<b>10. Tag</b> .....	<b>16</b>	Zahlen raten .....	36
Heute im Adventskalender .....	16	Das Programm .....	36
Mobiles Erdungsmessgerät .....	16	So funktioniert das Programm .....	36
Das Programm .....	16	<b>23. Tag</b> .....	<b>38</b>
So funktioniert das Programm .....	16	Heute im Adventskalender .....	38
<b>11. Tag</b> .....	<b>17</b>	Space Invaders .....	38
Heute im Adventskalender .....	17	Das Programm .....	38
Ampel mit Fußgängerampel mit Taster schalten .....	17	So funktioniert das Programm .....	38
Das Programm .....	17	<b>24. Tag</b> .....	<b>41</b>
So funktioniert das Programm .....	17	Heute im Adventskalender .....	41
<b>12. Tag</b> .....	<b>18</b>	Weihnachtsquiz .....	41
Heute im Adventskalender .....	18	Das Programm .....	41
Externer Taster schaltet LEDs um .....	18	So funktioniert das Programm .....	41
Das Programm .....	18		
So funktioniert das Programm .....	18		

## micro:bit-Adventskalender 2019

Der micro:bit ist eine Experimentierplatine, die sich sehr einfach programmieren lässt. Sie wurde ursprünglich für den Schulunterricht entwickelt, lässt sich aber auch für viele andere Experimente und Spiele einsetzen.

Natürlich ist der micro:bit nicht die erste programmierbare Platine. Bis vor einigen Jahren war die Programmierung von Microcontrollern und Einplatinencomputern nur etwas für Ingenieure. Erst die bekannten Platinen Raspberry Pi und Arduino machten diese Technik für jeden verständlich.

Im Gegensatz zum Raspberry Pi ist der micro:bit kein wirklicher Computer, sondern eine Microcontroller-Platine, die einfach ein einziges Programm abarbeitet, das vorher auf dem PC erstellt und dann per USB-Kabel übertragen wurde. Damit ist er vergleichbar mit dem Arduino und auch ähnlich einfach zu programmieren.

### Vorsichtsmaßnahmen

Auf keinen Fall sollte man irgendwelche Anschlüsse miteinander verbinden und abwarten, was passiert.

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

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

## 1. Tag

## 1. Tag

## Heute im Adventskalender

- Krokodilklemmenkabel

## Krokodilklemmenkabel

Mit diesen Kabeln können Sie Münzen, Löffel oder andere Gegenstände aus Metall als Sensorkontakte benutzen. Klemmen Sie dazu die eine Krokodilklemme an den Gegenstand, die andere an eine der fünf Kontaktflächen mit den Löchern auf dem micro:bit. Achten Sie dabei darauf, keinen Kurzschluss mit einem der benachbarten schmalen Kontakte zu verursachen. Am besten stecken Sie eine Seite der Krokodilklemme durch das Loch in der Platine.

## Den micro:bit vorbereiten

Um den micro:bit in Betrieb zu nehmen, braucht man:

- PC mit Windows 10, 8.1, 7 (oder Mac OSX)
- MicroUSB-Kabel

Die Verbindung zwischen PC und micro:bit 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 auch zur Datenübertragung verwendet.

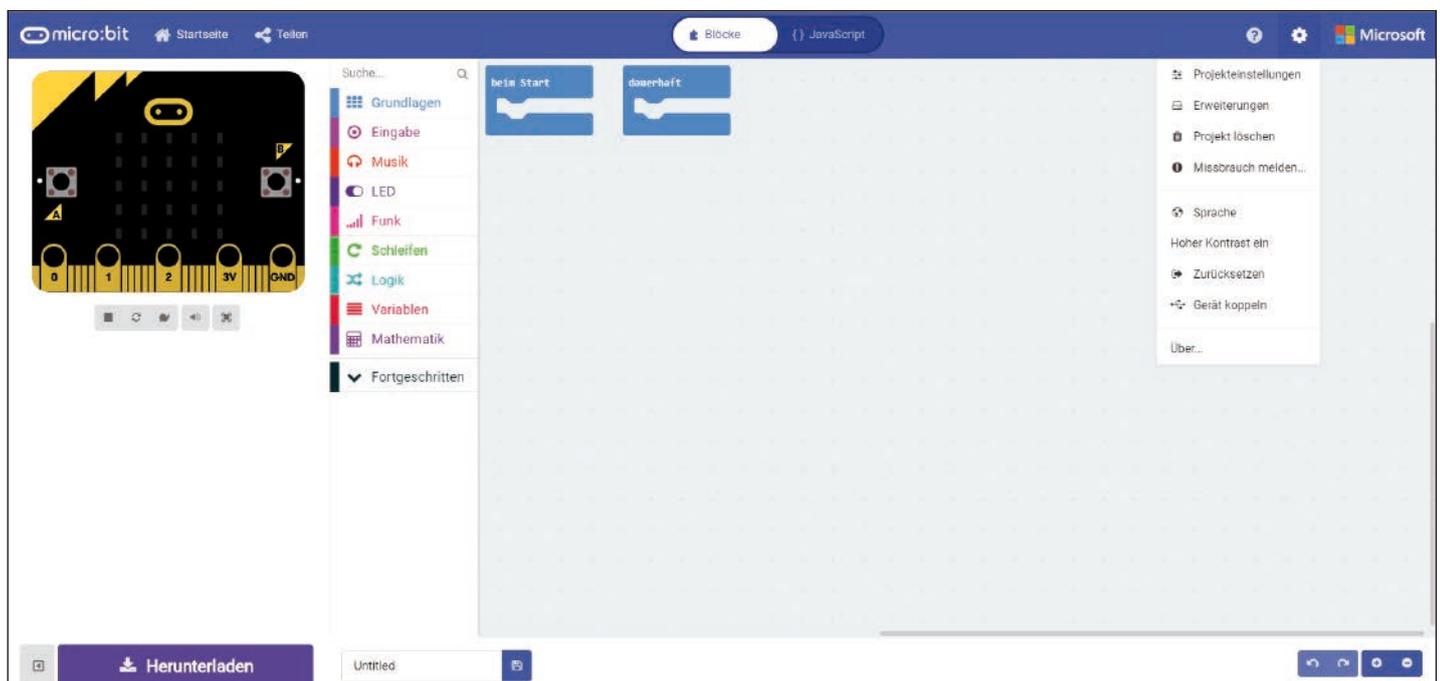
Schließen Sie den micro:bit mit einem USB-Kabel an einem freien USB-Anschluss Ihres PCs an. Beim ersten Mal wird automatisch ein Treiber installiert, damit der micro:bit in Zukunft als Laufwerk im Windows Explorer erscheint.

Nachdem der Treiber installiert ist, finden Sie den micro:bit im Windows Explorer bei den Laufwerken mit dem Namen **MICROBIT** und einem bisher nicht verwendeten Laufwerksbuchstaben. Der micro:bit ist aber trotzdem kein normales Laufwerk. Es kann immer nur eine Programmdatei darauf abgelegt werden. Sie ist im Windows Explorer auf dem Laufwerk nicht zu sehen. Auch das Speichern anderer Dateitypen ist nicht möglich.

## Der Makecode-Editor

Für die Programme in diesem Adventskalender verwenden wir die Programmierplattform Makecode von Microsoft - [makecode.microbit.org](https://makecode.microbit.org). In Makecode braucht man beim Programmieren keinen Programmcode zu tippen. Die Blöcke werden einfach per Drag and Drop aneinander gehängt. Der Makecode-Editor läuft im Browser. Es braucht kein Programm installiert zu werden.

Klicken Sie oben rechts auf das Einstellungen-Symbol, wählen Sie im Menü **Language** und dann die Sprache **Deutsch**.



Makecode Editor auf Deutsch umstellen.

Links oben ist ein micro:bit abgebildet, der das Programm abspielt. Dieser Simulator zeigt jederzeit auch Änderungen im Programm sofort, ohne dass Sie das Programm auf den "echten" micro:bit übertragen müssen.

Das Quadrat-Symbol hält den Simulator an, das Dreieck startet ihn danach wieder.

Im Makecode-Editor sind die verfügbaren Programmblöcke nach Gruppen geordnet.

Ziehen Sie einen Block aus der Blockpalette in den Arbeitsbereich, erscheint er zunächst grau. Erst Blöcke, die zusammen ein lauffähiges Programm ergeben, nehmen wieder ihre Grundfarbe an. Um einen Block zu löschen, ziehen Sie ihn einfach wieder aus dem Arbeitsbereich auf die Blockpalette zurück.

Die Blöcke sind so geformt, dass sie nur dann ineinander passen, wenn dies auch im Programm funktioniert. Dadurch können die gefürchteten Syntaxfehler nicht passieren, mit denen sich Programmierer alltäglich herumärgern müssen.

Zwei Blöcke sind in jedem neuen Programm vorgegeben:

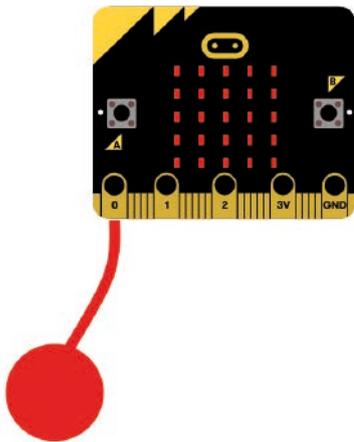
- **beim Start** - alle Blöcke in dieser Klammer werden beim Start einmal ausgeführt. Ein Programm startet nach der Übertragung auf den micro:bit automatisch. Außerdem startet das Programm neu, wenn man den Taster auf der Rückseite drückt oder den Stromanschluss neu anschließt.
- **dauerhaft** - Nachdem die Blöcke **beim Start** ausgeführt wurden, laufen die Programme in der Klammer **dauerhaft** als Endlosschleife, bis man den Stecker zieht.

### Die Programme zum Adventskalender

Die verwendeten Programme gibt es hier zum Download: [bit.ly/c-adventskalender-microbit-19](https://bit.ly/c-adventskalender-microbit-19). Entpacken Sie das Zip-Archiv in ein Verzeichnis auf Ihrer Festplatte. Die Programme sind im Download nach Tagen nummeriert. Klicken Sie auf der Startseite [makecode.microbit.org](https://makecode.microbit.org) auf **Importieren**, um ein Programm zu öffnen. Sie können die Programme aber auch einfach jeden Tag anhand der Abbildungen selbst zusammenbauen.

## Frohe Weihnachten

**Benötigte Teile:** 1x micro:bit, 1x Krokodilklemmenkabel



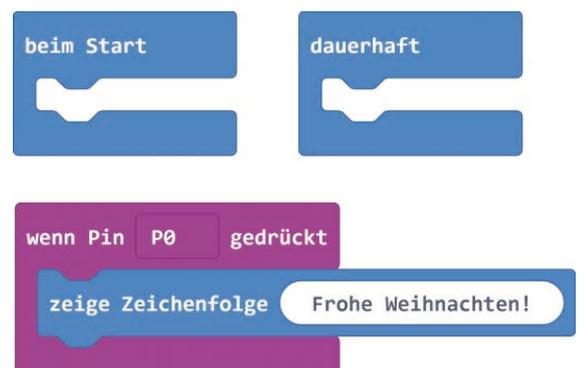
Klemmen Sie das Krokodilklemmenkabel am Kontakt 0 des micro:bit an. Klemmen Sie an das andere Ende eine Münze, einen Metalllöffel oder anderen leitfähigen Gegenstand.

### Das Programm

Das Programm `microbit-01.hex` zeigt einen Text auf der LED-Matrix, wenn der Sensor berührt wird.

### So funktioniert das Programm

Die Blöcke **beim Start** und **dauerhaft** werden in diesem Programm nicht gebraucht. Der Block **Wenn Pin P0 gedrückt** aus der Gruppe **Eingabe** arbeitet alle Blöcke innerhalb der Klammer ab, wenn der Pin 0 am micro:bit berührt wird. Das bedeutet, er ist zumindest schwach mit Masse verbunden.



Sollte das Programm bei einfacher Berührung des Sensors nicht reagieren, berühren Sie gleichzeitig den GND-Kontakt ganz rechts auf dem micro:bit.

### **Programm auf den micro:bit übertragen**

Um das Programm nicht nur auf dem Simulator, sondern wirklich auf dem micro:bit abzuspielen, geben Sie ihm in dem Feld ganz unten einen Namen und klicken dann auf den großen Button **Herunterladen**. Speichern Sie die heruntergeladene Datei direkt auf dem Laufwerk MICROBIT. Natürlich muss der micro:bit dazu über das USB-Kabel mit dem PC verbunden sein. Die gelbe LED beginnt, schnell zu blinken. Wenn die Übertragung beendet ist, leuchtet diese LED wieder gleichmäßig.

Das Programm startet kurz nach Abschluss der Übertragung automatisch. Auf dem micro:bit kann sich immer nur ein Programm befinden. Die \*.hex-Datei wird kurz nach dem Übertragen im Explorer nicht mehr auf dem Laufwerk MICROBIT angezeigt.

### **Browsereinstellungen zum Speichern der Programme**

Je nach Einstellung des Browsers haben Sie die Möglichkeit, das Verzeichnis zum Speichern der Programme und vor allem den Dateinamen frei zu wählen. Das ist wichtig, damit Sie die Datei auf der Festplatte wiederfinden und von dort auf den micro:bit übertragen können.

#### **Einstellung in Firefox**

Klicken Sie oben rechts auf das Menüsymbol und wählen Sie **Einstellungen**. Schalten Sie auf der Einstellungs-Seite **Allgemein** unter **Downloads** den Schalter **Jedes Mal nachfragen, wo eine Datei gespeichert werden soll** ein.

#### **Einstellung in Google Chrome**

Klicken Sie oben rechts auf das Menüsymbol und wählen Sie **Einstellungen**. Klicken Sie ganz unten auf **Erweitert**. Blättern Sie noch weiter nach unten und schalten Sie unter **Downloads** den Schalter **Vor dem Download von Dateien nach dem Speicherort fragen** ein.

#### **Einstellung in Microsoft Edge**

Klicken Sie oben rechts auf das Menüsymbol und wählen Sie **Einstellungen**. Schalten Sie ganz unten unter **Downloads** den Schalter **Bei jedem Download nach dem Speicherort fragen** ein.

## 2. Tag

### Heute im Adventskalender

· Krokodilklemmenkabel

### Frohe Weihnachten

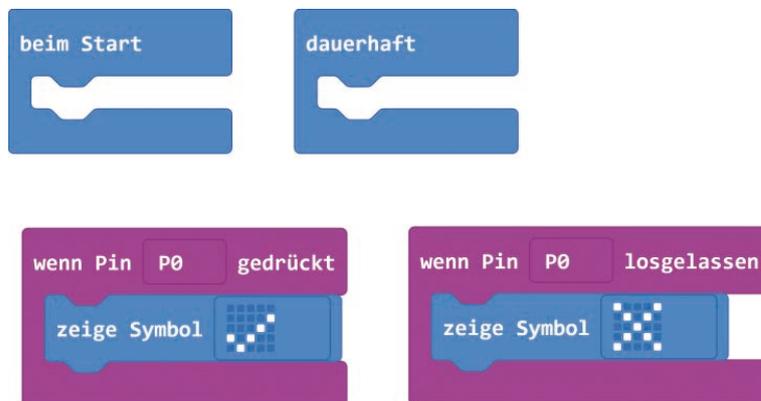
**Benötigte Teile:** 1x micro:bit, 2x Krokodilklemmenkabel

Klemmen Sie ein Krokodilklemmenkabel am Kontakt 0 des micro:bit und das zweite am GND-Kontakt an. Klemmen Sie an die anderen Enden Münzen, Metalllöffel oder andere leitfähige Gegenstände.

Durch die beiden Sensorkontakte ist auch bei schwacher Erdung eine Reaktion des Programms sichergestellt, wenn man beide Sensoren gleichzeitig berührt.

### Das Programm

Das Programm `microbit-02.hex` zeigt beim Berühren des Sensors ein Häkchen auf der LED-Matrix, beim Loslassen ein X-Symbol.

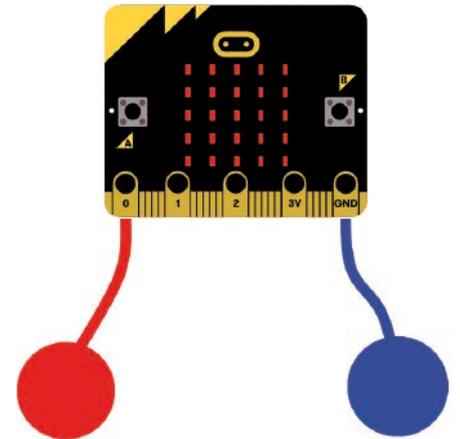


### So funktioniert das Programm

Der Block **zeige Symbol** aus der Gruppe **Grundlagen** zeigt vordefinierte Symbole auf der LED-Matrix, was deutlich schneller geht, als selbstdefinierte LED-Muster anzuzeigen. Klicken Sie auf das Symbol rechts im Block, erscheint eine Auswahlpalette mit allen vorgegebenen Symbolen.

Der Block **wenn Pin P0 losgelassen** arbeitet die Blöcke innerhalb der Klammer immer dann ab, wenn der Sensor am Kontakt 0 losgelassen wird, also keine Verbindung zur Masse mehr hat.

2. Tag

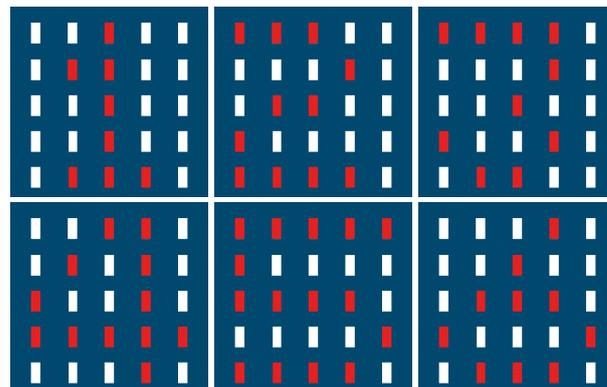
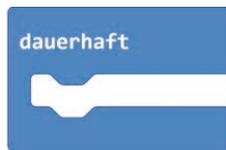
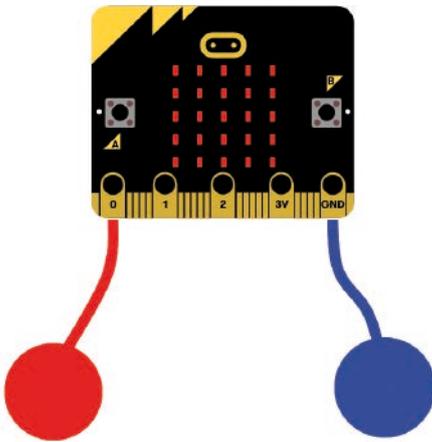


## 3. Tag

## 3. Tag

## Heute im Adventskalender

- 2x Drahtelektroden



## Zahlenwürfel mit Bananen oder Äpfeln

**Benötigte Teile:** 1x micro:bit, 2x Krokodilklemmenkabel, 2x Drahtelektrode

Mit den Drahtelektroden können Sie Bananen, Äpfel oder ähnliche Gegenstände als Sensorkontakte nutzen. Stecken Sie eine Elektrode in eine Frucht und klemmen Sie ein Krokodilklemmenkabel daran an.

## Das Programm

Für viele Spiele braucht man einen Würfel, aber oft ist gerade keiner griffbereit. Das Programm `microbit-03.hex` zeigt, wie einfach es ist, mit dem micro:bit einen Würfel zu programmieren. Die erste ganz einfache Version des Würfelprogramms soll eine Zahl zwischen 1 und 6 anzeigen, wenn man den Sensorkontakt - z.B. die Banane - berührt.

Der Block **wähle eine zufällige Zahl zwischen ... und ...** aus der Gruppe **Mathematik** erzeugt eine Zufallszahl aus dem angegebenen Zahlenbereich. Ziehen Sie diesen Block einfach in das Zahlenfeld im Block **zeige Nummer ...** aus der Gruppe **Grundlagen**.

Wenn Sie jetzt den Sensor berühren, zeigt der micro:bit eine zufällige Zahl zwischen 1 und 6.

## Wie entstehen eigentlich Zufallszahlen?

Man denkt, in einem Programm könnte nichts zufällig passieren, alles wäre geplant. Wie kann ein Programm dann in der Lage sein, zufällige Zahlen zu generieren? Teilt man eine große Primzahl durch irgendeinen Wert, ergeben sich ab der x-ten Nachkommastelle Zahlen, die kaum noch vorhersehbar sind und sich auch ohne jede Regelmäßigkeit ändern, wenn man den Divisor regelmäßig erhöht. Dieses Ergebnis ist zwar scheinbar zufällig, lässt sich aber durch ein identisches Programm oder 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.

## 4. Tag

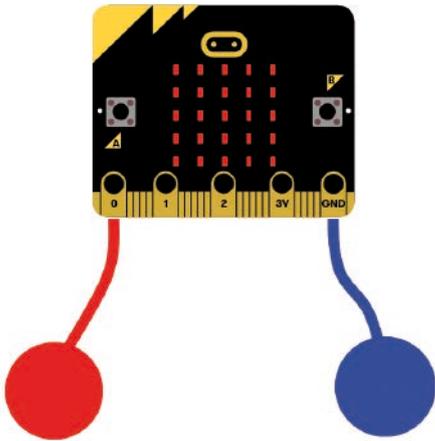
### Heute im Adventskalender

• Knete

Aus Knete lassen sich gute Sensorkontakte bauen, da sie den Strom etwa genauso gut leitet wie die menschliche Haut. Ein Stück Knete bietet aber eine größere Kontaktfläche mit der Hand als ein einfacher Draht. Dadurch kommt es beim Anfassen nicht so leicht zu einem Wackelkontakt. Stecken Sie je eine Krokodilklemme jedes der beiden Kabel in ein Stück Knete und klemmen Sie die anderen Enden an den Pins 0 und GND auf dem micro:bit an.

### Echter Würfel mit Würfelsymbolen

**Benötigte Teile:** 1x micro:bit, 2x Krokodilklemmenkabel, 2x Knete



### Das Programm

Das Programm von gestern erfüllt die gestellte Aufgabe, es sieht aber nicht wie ein echter Würfel aus. Das Programm `microbit-04.hex` ist etwas aufwendiger programmiert, zeigt aber dafür einen echten Würfel in der typischen Grafik, der wenn man den Knetekontakt berührt, auch jedes Mal wieder ein neues Würfelergbnis anzeigt.

### So funktioniert das Programm

Der neue Würfel zeigt das Ergebnis nicht direkt als Zahl an, sondern soll je nach gewürfeltem Wert eines von sechs Bildern anzeigen.

Die zufällig ermittelte Zahl wird diesmal nicht sofort angezeigt, sondern zunächst in der Variable `wuerfel` gespeichert.

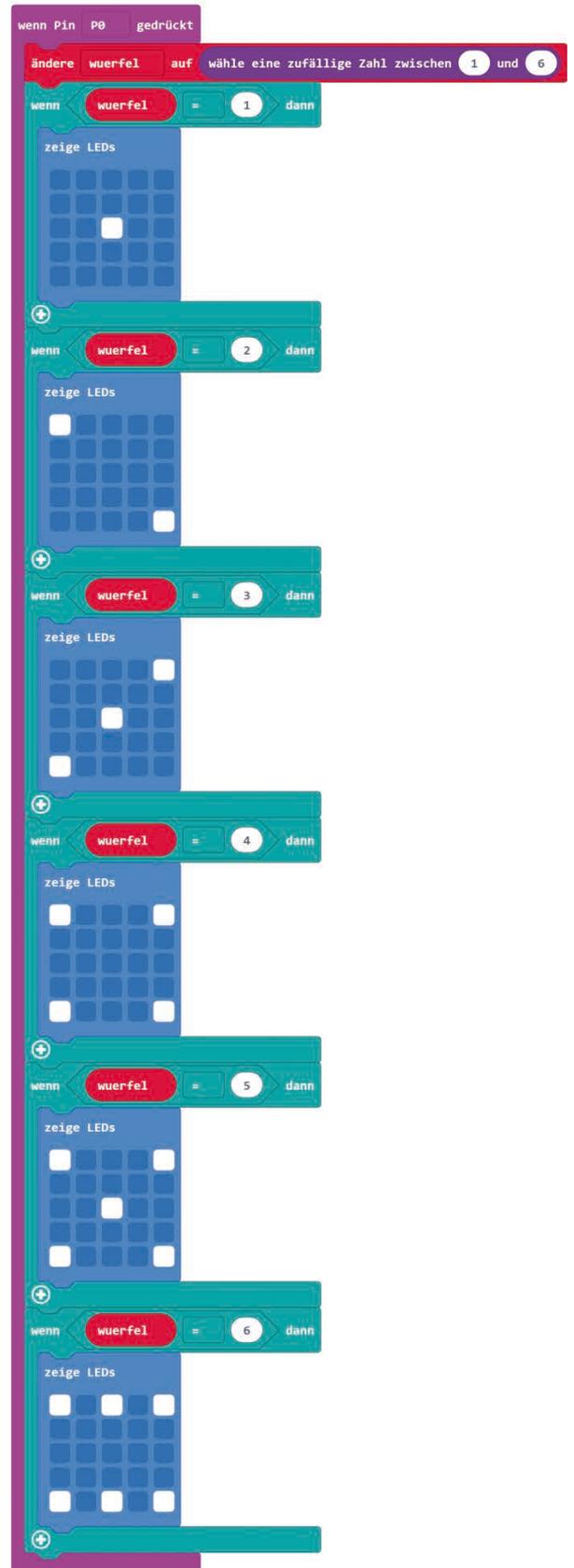
### Variablen im Makecode-Editor

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 im Makecode-Editor erst einmal auf der Blockpalette **Variablen** mit dem Button **Make a Variable** angelegt werden, bevor man sie benutzen kann. Anschließend können Sie das Symbol der neu angelegten Variablen in ein dafür vorgesehenes Feld eines Blocks im Programm ziehen. In der Gruppe **Variablen** stehen zusätzlich zwei verschiedene Blöcke zum Verändern der Variablen zur Verfügung, die leicht zu verwechseln sind, aber unterschiedliche Funktionen haben.

**ändere Variable auf ...** setzt die Variable auf einen bestimmten Wert.

**ändere Variable um ...** erhöht die Variable um einen bestimmten Wert.

4. Tag



Im Programm folgen sechs verschiedene **wenn ... dann ...**-Abfragen aus der Gruppe **Logik**. Die Blöcke innerhalb der Klammer dieser Abfrage werden immer dann ausgeführt, wenn die Bedingung in der Abfrage wahr ist.

Alle diese Abfragen verwenden den Block **... = ...** aus der Gruppe **Logik**, um zu prüfen, ob der Wert der Variable **wuerfel** einer der sechs möglichen Zahlen entspricht.

Ist das der Fall, wird das entsprechende Würfelmuster auf der LED-Matrix angezeigt. Dazu wird der Block **zeige LEDs** aus der Gruppe **Grundlagen** verwendet. Hier können Sie durch Anklicken das gewünschte LED-Muster festlegen.

#### **Blöcke duplizieren**

Ähnliche Programmblöcke, wie hier die sechs **wenn ... dann ...** Abfragen, brauchen nicht jedes Mal komplett neu zusammgebaut zu werden. Klicken Sie mit der rechten Maustaste auf einen Block und wählen Sie im Kontextmenü **Duplizieren**. Automatisch wird eine Kopie des Blocks und aller ineinander verschachtelten Blöcke angelegt, die Sie an die gewünschte Position im Programm ziehen und weiter bearbeiten können.

## 5. Tag

### Heute im Adventskalender

• Schaltdraht

Heute ist Schaltdraht im Adventskalender enthalten. Damit bauen Sie später verschiedene Drahtverbindungen zwischen Bauteilen.

5. Tag

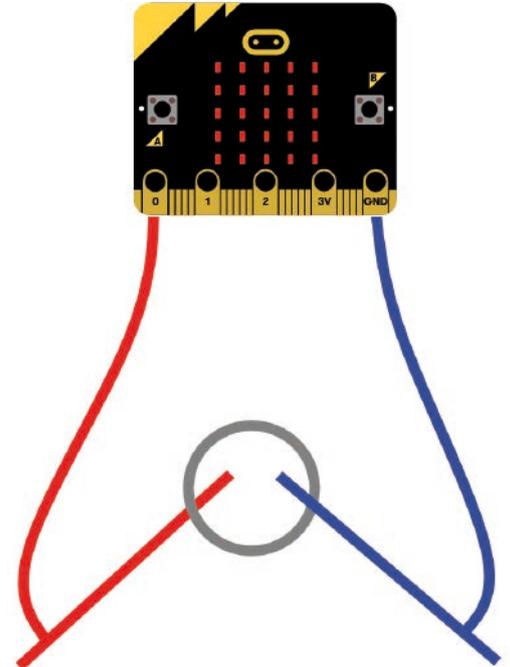
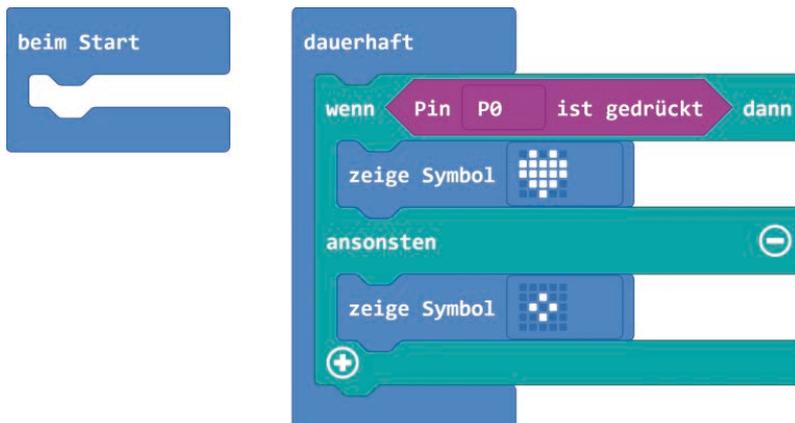
### Geschicklichkeitsspiel

Biegen Sie aus dem Schaltdraht verschieden geformte Drahringe. Entfernen Sie dazu vorher mit einem scharfen Messer die Isolierung. Klemmen Sie dann je eine Drahtelektrode in die beiden Krokodilklemmenkabel. Versuchen Sie nun, die Drahringe von einer auf die andere Elektrode zu übergeben, sodass Sie kurzzeitig beide berühren. In diesem Moment zeigt der micro:bit ein Herz auf der LED-Matrix an. Achten Sie darauf, nur die isolierten Krokodilklemmen anzufassen und nicht die Elektroden zu berühren, um mit der Hand keine Verbindung zu schließen.

**Benötigte Teile:** 1x micro:bit, 2x Krokodilklemmenkabel, 2x Drahtelektrode

### Das Programm

Das Programm `microbit-05.hex` zeigt eine weitere Methode, die Kontaktpins abzufragen. Sie ist oftmals schneller, da zu einem bestimmten Zeitpunkt im Programm der Kontakt ausgewertet wird, ohne dass eine Abfrage, die ständig im Hintergrund läuft, das Programm unterbrechen muss.



### So funktioniert das Programm

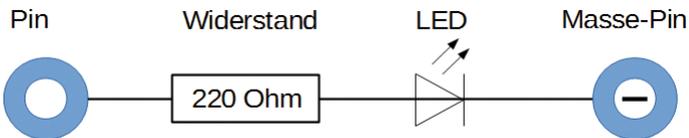
In der **dauerhaft**-Schleife fragt ein **wenn ... dann ... ansonsten ...**-Block immer wieder den Pin 0 ab. Die Blöcke in der ersten Klammer werden immer dann ausgeführt, wenn die Abfrage **wahr** ergibt, die Blöcke in der zweiten Klammer dann, wenn die Abfrage **falsch** ergibt. In diesem Programm wird das Herzsymbol angezeigt, wenn der Kontakt mit Masse geschlossen ist, die kleine Raute wird angezeigt, wenn der Pin 0 nicht mit Masse verbunden ist.

## 6. Tag

## 6. Tag

## Heute im Adventskalender

- LED rot mit Vorwiderstand



Schaltplan einer LED mit Vorwiderstand.

## LEDs

LEDs (zu Deutsch: Leuchtdioden) leuchten, wenn Strom in Durchflussrichtung durch sie hindurch 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, da sie nur einen sehr geringen Widerstand hat. Um den Durchflussstrom zu begrenzen und so 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 micro:bit vor zu hohen Stromstärken. Die LEDs im Adventskalender haben den Vorwiderstand bereits eingebaut und können daher direkt an die Pins angeschlossen werden.

## 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.

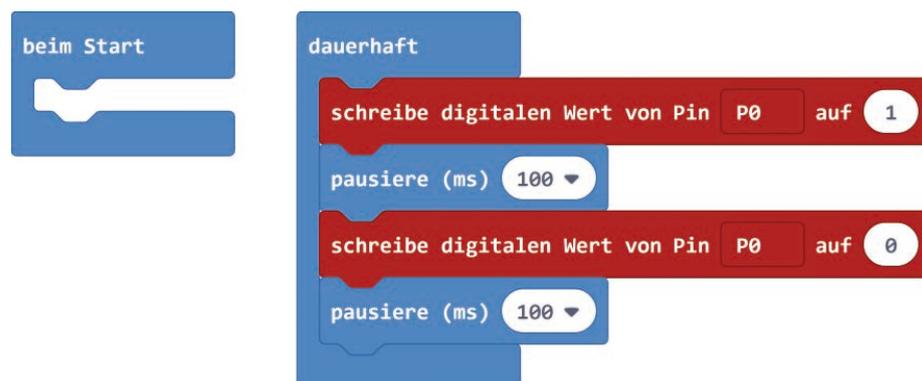
## LED blinkt

**Benötigte Teile:** 1x micro:bit, 2x Krokodilklemmenkabel, 1x LED rot

Schließen Sie die LED mit zwei Krokodilklemmenkabeln an den Pins 0 und GND an. Dabei muss der kürzere Draht am GND-Pin angeschlossen sein.

## Das Programm

Das Programm `microbit-06.hex` lässt die LED gleichmäßig blinken.

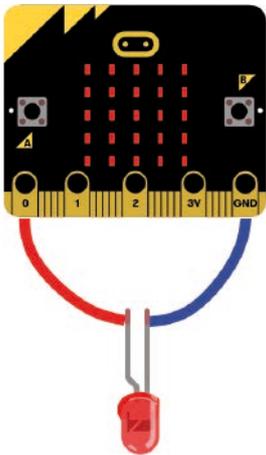


In der **dauerhaft**-Schleife wird die LED abwechselnd ein- und ausgeschaltet. Dazwischen wartet das Programm jedes Mal 100 Millisekunden lang.

## So funktioniert das Programm

Der Block **schreibe digitalen Wert von Pin P0 auf ...** schaltet den Pin **P0** ein oder aus. Steht im Zahlenfeld der Wert **1**, wird der Pin eingeschaltet, steht dort der Wert **0**, wird er ausgeschaltet.

Dieser Block ist etwas ungeschickt übersetzt. Die Formulierung **schreibe ... auf digitalen Pin P...** erklärt besser, was dieser Block bewirkt, nämlich einen Logikwert auf einem Pin auszugeben oder einen Pin auf einen Logikwert zu setzen. Logischer wäre auch, die Logikwerte **wahr** und **falsch** zu verwenden, dieser Block verwendet aber die Zahlenwerte **1** und **0**.



# 7. Tag

7. Tag

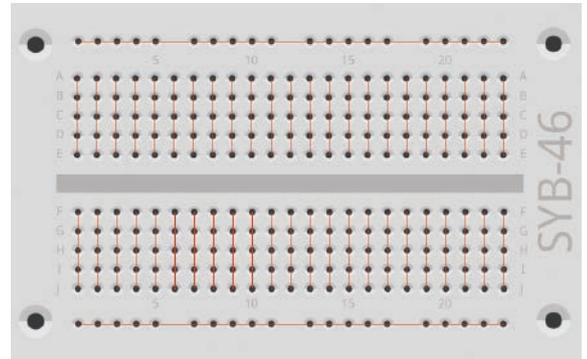
## Heute im Adventskalender

- Steckbrett (SYB 46)

### Steckbrett

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

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



Die Verbindungen auf dem Steckbrett.

### Steckbrücken und Anschlüsse für Krokodilklemmen

Mit dem Schaltdraht stellen Sie kurze Verbindungsbrücken her, mit denen dicht nebeneinander liegende Kontaktreihen auf der Steckplatine verbunden werden. Solche Drahtbrücken können auch dafür verwendet werden, Krokodilklemmenkabel am Steckbrett anzuschließen.

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 die Isolierung auf der ganzen Länge mit einem scharfen Messer. An diesen Drahtbücken können Krokodilklemmen befestigt werden. Besonders einfach geht das, wenn eine Drahtbrücke über den "Graben" in der Mitte des Steckbretts gespannt wird.

### LED mit Tastern ein- und ausschalten

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 2x Krokodilklemmenkabel, 1x LED rot, 2x Drahtbrücke blank

Drücken Sie auf den Taster A links auf dem micro:bit, wird die LED eingeschaltet, drücken Sie auf den Taster B rechts, wird sie wieder ausgeschaltet.

### Das Programm

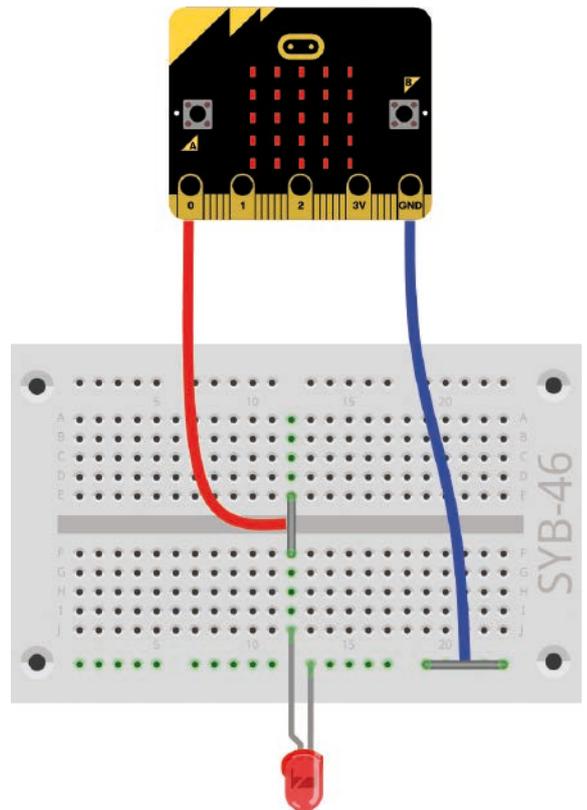
Das Programm `microbit-07.hex` lässt die LED gleichmäßig blinken.

```

beim Start
dauerhaft

wenn Knopf A gedrückt
schreibe digitalen Wert von Pin P0 auf 1

wenn Knopf B gedrückt
schreibe digitalen Wert von Pin P0 auf 0
    
```



fritzing

### So funktioniert das Programm

Das Programm besteht aus zwei Blöcken, die die beiden Taster abfragen. Wird der Taster A gedrückt, wird der digitale Wert 1 auf den Pin P0 geschrieben und die LED eingeschaltet. Beim Druck auf den Taster B wird der Wert 0 auf den Pin geschrieben.

## 8. Tag

### Heute im Adventskalender

· Krokodilklemmenkabel

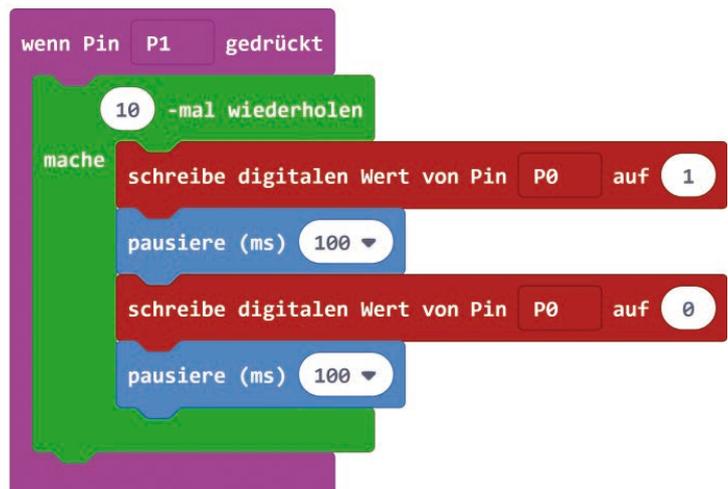
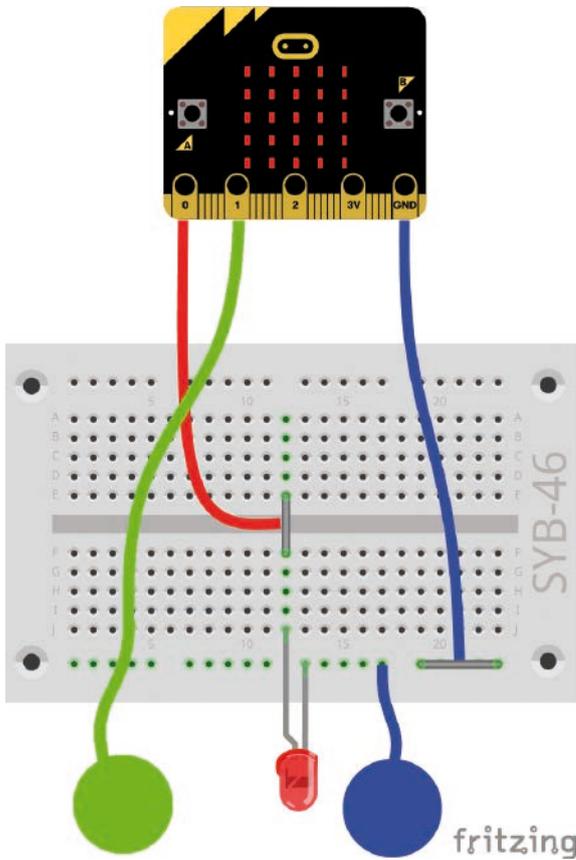
### LED blinkt auf Berührung

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 3x Krokodilklemmenkabel, 1x LED rot, 2x Drahtbrücke blank, 1x Schaltdraht, 2x Knete

Der Knetesensor für den Massekontakt ist mit einem längeren Stück Schaltdraht an der Masseleiste des Steckbretts angeschlossen. Bei diesem Draht brauchen Sie an beiden Enden nur je etwa 1 cm der Isolierung zu entfernen.

### Das Programm

Das Programm `microbit-08.hex` lässt die LED zehnmal blinken, wenn man den Sensor berührt.



### So funktioniert das Programm

Das Programm wartet, bis der Pin P1 berührt wird. Im Auswahlfeld des Blocks **wenn Pin ... gedrückt** können Sie den zu verwendenden Pin auswählen.

Der Block **... -mal wiederholen mache ...** aus der Gruppe **Schleifen** wiederholt die Blöcke innerhalb der Klammer so oft, wie oben in der Bedingung angegeben.

# 9. Tag



## Heute im Adventskalender

• LED grün mit Vorwiderstand

## LEDs mit einer Taste abwechselnd umschalten

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 3x Krokodilklemmenkabel, 1x LED rot, 1x LED grün, 3x Drahtbrücke blank

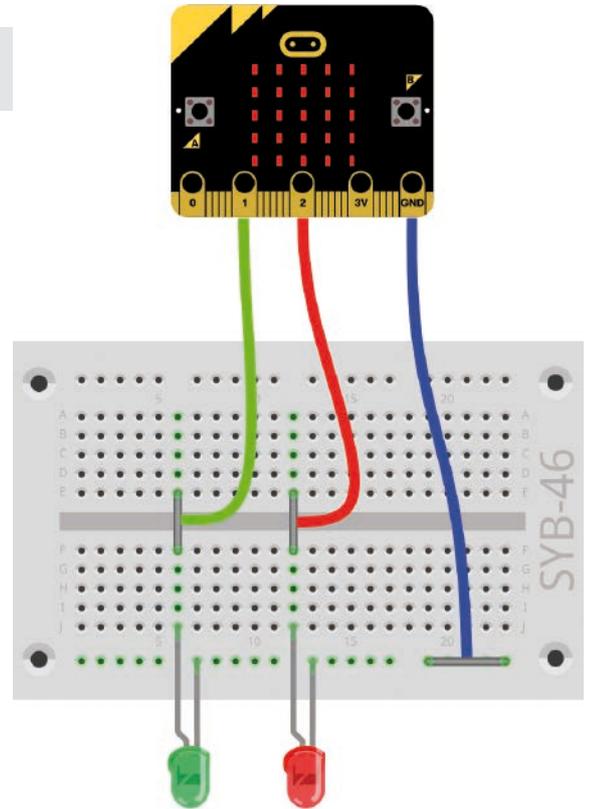
## Das Programm

Das Programm `microbit-09.hex` schaltet beim Druck auf die Taste A eine der beiden LEDs ein und die andere aus, beim nächsten umgekehrt.



```

wenn Knopf A gedrückt
  ändere LED2 auf LED1
  schreibe digitalen Wert von Pin P2 auf LED2
  ändere LED1 auf 1 - LED1
  schreibe digitalen Wert von Pin P1 auf LED1
  
```



fritzing

## So funktioniert das Programm

Damit das Programm weiß, ob der nächste Tastendruck eine LED ein- oder ausschalten soll, merkt es sich die aktuellen Schaltzustände der beiden LEDs in den beiden Variablen **LED1** und **LED2**.

Wenn der Knopf A gedrückt wird, wird als Erstes der Wert der Variablen **LED2** auf den bisherigen Zustand der **LED1** gesetzt. Die LED2 am Pin P2 wird entsprechend der Variable **LED2** ein- bzw. ausgeschaltet.

Anschließend wird die Variable **LED1** auf den jeweils anderen Schaltzustand gesetzt. Mit einem Block **1 - ...** aus der Gruppe **Mathematik** lässt sich das leicht erledigen. Damit wird aus einer 1 eine 0 und aus einer 0 eine 1.

Wert vor der Umrechnung	Formel	Ergebnis
0	1 - 0	1
1	1 - 1	0

Anschließend wird der neue Wert der Variablen **LED1** auf dem Pin **P1** ausgegeben, um die LED entsprechend zu schalten.

## 10. Tag

## 10. Tag

## Heute im Adventskalender

- Krokodilklemmenkabel

## Mobiles Erdungsmessgerät

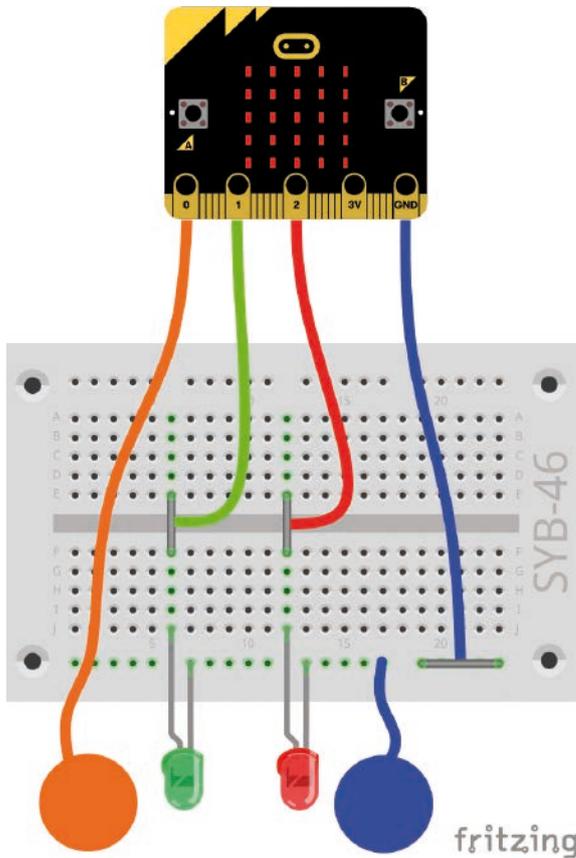
**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 4x Krokodilklemmenkabel, 1x LED rot, 1x LED grün, 3x Drahtbrücke blank, 1x Schaltdraht, 2x Knete

Die Verbindung zur Erde ist nicht immer gleich gut. Wie hoch der Widerstand zwischen der Hand, die den Sensorkontakt berührt, und der Erde ist, hängt von vielen Dingen ab, vor allem, was für Schuhe man trägt und auf welchem Fußboden man steht. Barfuß im nassen Gras ist die Verbindung zur Masse der Erde am besten, aber auch auf Steinfußboden funktioniert es meistens gut. Ein mobiles Erdungsmessgerät zeigt an, ob Sie eine für die Funktion von Sensorkontakten ausreichend gute Verbindung zur Erde haben.

Legen Sie den über ein langes Stück Schaltdraht mit der Masseleitung verbundenen Knetesensor auf den Fußboden oder auf verschiedene Möbelstücke und halten Sie den anderen Knetesensor in der Hand.

## Das Programm

Das Programm `microbit-10.hex` lässt die grüne LED am Pin P1 leuchten, wenn eine Verbindung zwischen dem Sensorkontakt am Pin P0 und der Erde besteht. Andernfalls leuchtet die rote LED am Pin P2.



beim Start

dauerhaft



## So funktioniert das Programm

In der **dauerhaft**-Schleife prüft eine **wenn ... dann ... ansonsten**-Abfrage, ob der Kontakt **P0** berührt wurde. Dazu verwenden wir den Block **Pin P0 ist gedrückt** aus der Gruppe **Eingabe**. Je nachdem, ob das Ergebnis dieser Abfrage wahr oder falsch ist, wird die grüne oder die rote LED eingeschaltet. Die jeweils andere LED wird ausgeschaltet.

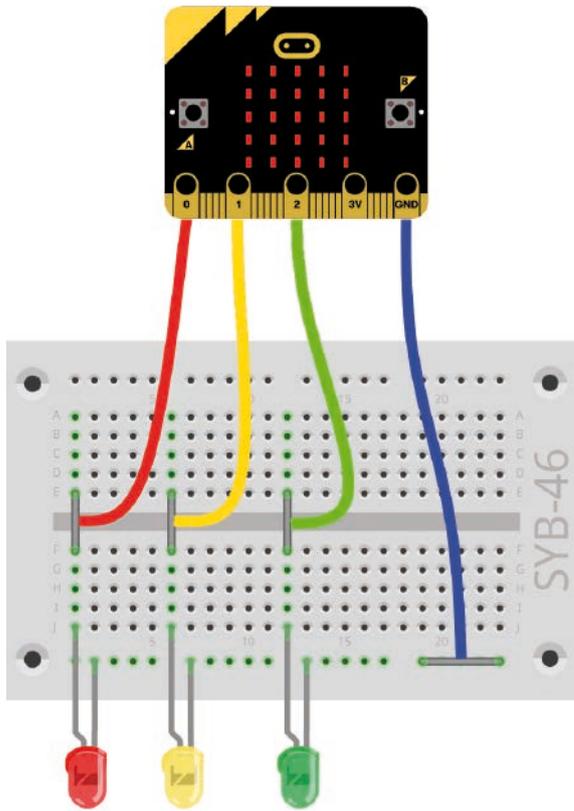
## 11. Tag

### Heute im Adventskalender

• LED gelb mit Vorwiderstand

### Ampel mit Fußgängerampel mit Taster schalten

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 4x Krokodilklemmenkabel, 1x LED rot, 1x LED gelb, 1x LED grün, 4x Drahtbrücke blank



Drei LEDs stellen eine einfache Ampelschaltung mit Fußgängerampel auf der LED-Matrix des micro:bit dar. Während der Rotphase der Verkehrsampel ändert sich das Fußgängersymbol auf der LED-Matrix. Da der micro:bit nur drei Anschlusspins hat, stellt die Schaltung keine Rot/Grün-Fußgängerampel dar.

### Das Programm

Das Programm `microbit-11.hex` startet den Ampelzyklus, wenn der Taster gedrückt ist. Solange der Taster nicht gedrückt ist, bleibt die Verkehrsampel grün und das Fußgängersymbol zeigt ein stehendes Ampelmännchen.

### So funktioniert das Programm

Beim Start wird auf der LED-Matrix das stehende Ampelmännchen angezeigt, die grüne LED am Pin P2 wird eingeschaltet. Die anderen beiden LEDs sind automatisch ausgeschaltet, da alle Pins standardmäßig beim Programmstart auf 0 stehen. Diese Einstellung bleibt bestehen, bis jemand die Taste A auf dem micro:bit drückt.

Jetzt startet der Ampelzyklus über Gelb auf Rot. Während der Rotphase der Verkehrsampel schaltet eine Schleife 40-mal das Symbol auf der LED-Matrix um. Zwischen den Schaltvorgängen sind keine Wartezeiten nötig. Da die Darstellung der Symbole bereits einige Millisekunden dauert, ist das Blinken gut zu erkennen. Anschließend erscheint wieder das stehende Ampelmännchen, und der Ampelzyklus läuft weiter über Rot/Gelb zu Grün. In diesem Zustand wartet die Ampel wieder darauf, dass die Taste gedrückt wird.

11. Tag

dauerhaft

```

beim Start
  zeige Symbol [Ampelmännchen]
  schreibe digitalen Wert von Pin P2 auf 1
  
```

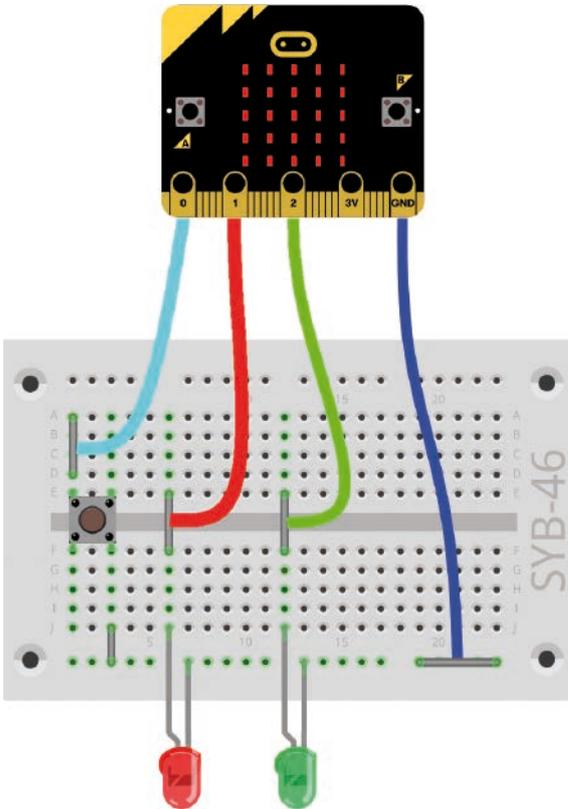
```

wenn Knopf A gedrückt
  schreibe digitalen Wert von Pin P2 auf 0
  schreibe digitalen Wert von Pin P1 auf 1
  pausiere (ms) 500
  schreibe digitalen Wert von Pin P1 auf 0
  schreibe digitalen Wert von Pin P0 auf 1
  pausiere (ms) 500
  mache 4 -mal wiederholen
    zeige Symbol [Ampelmännchen]
    zeige Symbol [Ampelmännchen]
  zeige Symbol [Ampelmännchen]
  pausiere (ms) 500
  schreibe digitalen Wert von Pin P1 auf 1
  pausiere (ms) 500
  schreibe digitalen Wert von Pin P0 auf 0
  schreibe digitalen Wert von Pin P1 auf 0
  schreibe digitalen Wert von Pin P2 auf 1
  
```

## 12. Tag

### Heute im Adventskalender

· Taster



#### Taster

Zusätzlich zu den auf dem micro:bit fest verbauten Tastern können an den Pins P0, P1 und P2 externe Taster angeschlossen werden. Der Taster heute im Adventskalender wird direkt auf das Steckbrett gesteckt. 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. Die Verbindung wird beim Loslassen sofort wieder getrennt.

#### Externer Taster schaltet LEDs um

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 4x Krokodilklemmenkabel, 1x LED rot, 1x LED grün, 1x Taster, 5x Drahtbrücke blank

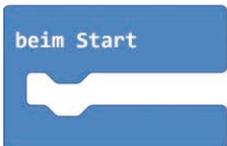
#### Das Programm

Das Programm `microbit-12.hex` schaltet über einen Taster zwei LEDs um. Ist der Taster gedrückt, leuchtet die grüne LED am Pin P2. Solange er nicht gedrückt ist, leuchtet die rote LED am Pin P1.

#### So funktioniert das Programm

Das Programm funktioniert nach bekanntem Schema. Eine **wenn ... dann ... ansonsten ...**-Abfrage prüft, ob der Taster gedrückt ist, ob also der Pin P0 mit Masse verbunden ist. Ist das der Fall, wird die LED am Pin P1 aus- und die am Pin P2 eingeschaltet, im anderen Fall umgekehrt.

fritzing



## 13. Tag

### Heute im Adventskalender

• 15-kOhm-Potentiometer

13. Tag

### 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.

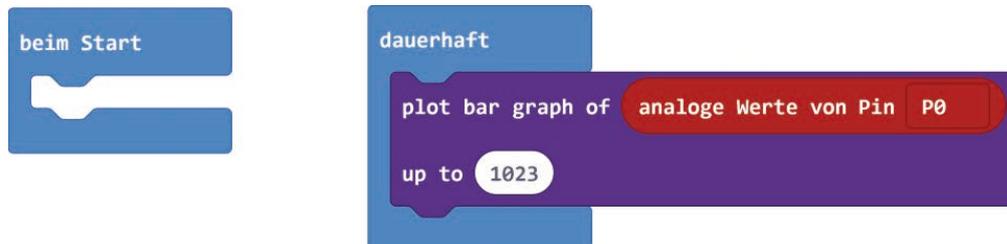
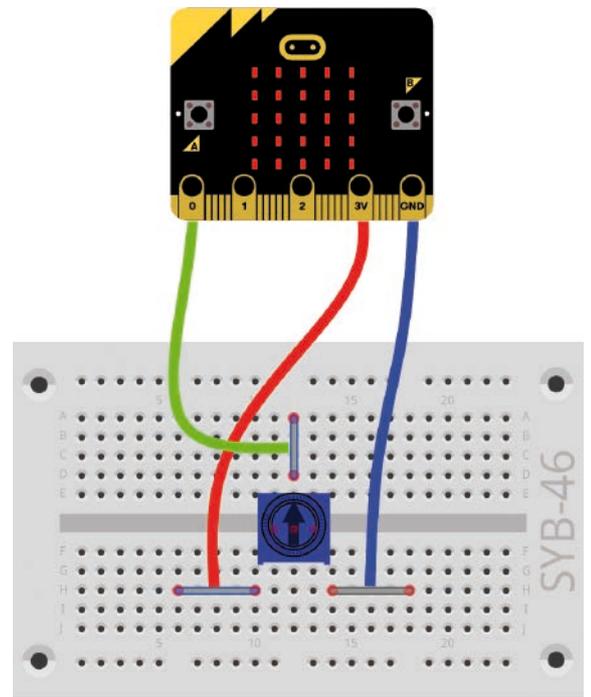
### Analoge Pegelanzeige auf der LED-Matrix

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 3x Krokodilklemmenkabel, 1x Potentiometer, 3x Drahtbrücke blank

Die drei Pins P0, P1 und P2 können anstatt digitaler Eingangswerte auch analoge Werte lesen. Diese werden je nach anliegender Spannung als Zahlenwert zwischen 0 und 1023 dargestellt.

### Das Programm

Auf einer Pegelanzeige lassen sich analoge Werte auf einen Blick ungefähr abschätzen. Solche Anzeigen aus mehreren LEDs werden zum Beispiel bei Lautstärke- oder Temperaturreglern verwendet. Das Programm `microbit-13.hex` zeigt den auf dem Potentiometer eingestellten Wert als Balkengrafik auf der LED-Matrix an.



### So funktioniert das Programm

Das Programm verwendet den Block **plot bar graph of ... up to ...** aus der Gruppe **LED**. Dieser Block stellt den analogen Wert, der mit dem Block **analoge Werte von Pin ...** aus der Gruppe **Pins** eingelesen wird, als Balkengrafik auf der LED-Matrix dar. Der zweite Wert in diesem Block gibt den maximal möglichen Wert der Balkengrafik an, um die Werte auf die LED-Matrix zu skalieren. 20% des maximal möglichen Wertes entspricht also einer Zeile von LEDs. Zwischenwerte werden dargestellt, indem die nächsthöhere Zeile in der Mitte beginnend aufgefüllt wird.

## 14. Tag

### Heute im Adventskalender

- 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.

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	$10^9 = 1.000.000.000$	

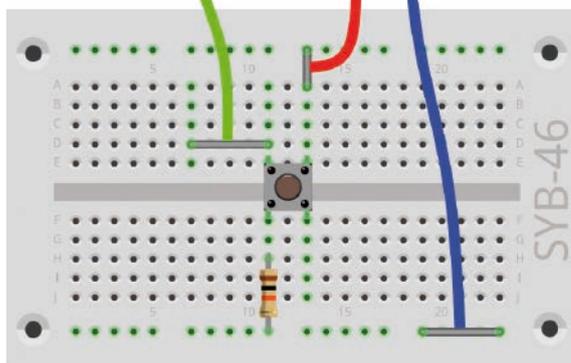
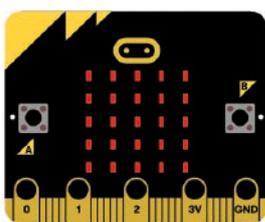
Die farbigen Ringe auf den Widerständen geben den Widerstandswert an. Mit etwas Übung sind sie deutlich leichter zu erkennen als winzig kleine Zahlen, die man nur noch auf ganz alten Widerständen findet.

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

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

### Taster mit Pulldown-Widerstand

Bei offenem Taster hat der Eingang, an dem der Taster angeschlossen ist, theoretisch keinen eindeutig definierten Zustand. Wenn ein Programm diesen Pin abfragt, kann es zu zufälligen Ergebnissen kommen, was bei digitalen Eingangssignalen durch eine auf dem micro:bit eingebaute Elektronik ausgeglichen wird.



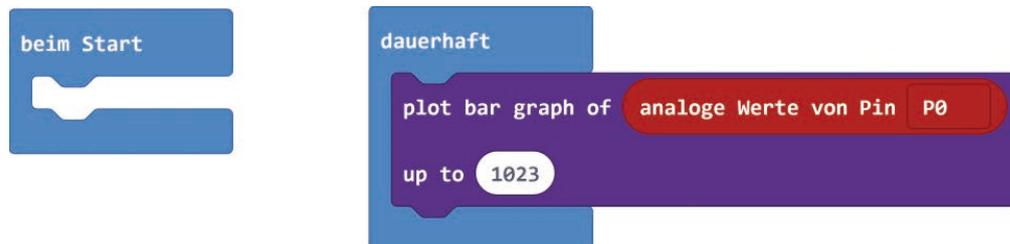
fritzing

Schließt man einen Taster an einen Eingangspin an und wertet das analoge Signal aus, ist dieser undefinierte Zustand deutlich zu sehen. Um das zu verhindern, schließt man einen vergleichsweise sehr hohen Widerstand - üblicherweise 10 kOhm - gegen Masse. Dieser sogenannte Pulldown-Widerstand zieht den Status des Eingangspins bei geöffnetem Taster wieder nach unten auf 0 V. Da der Widerstand sehr hoch ist, besteht, solange der Taster gedrückt ist, auch keine Kurzschlussgefahr. Im gedrückten Zustand des Tasters sind +3 V und die Masseleitung direkt über diesen Widerstand verbunden.

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 3x Krokodilklemmenkabel, 1x Taster, 1x Widerstand 10 kOhm (braun - schwarz - orange), 3x Drahtbrücke blank

### Das Programm

Das Programm `microbit-14.hex` zeigt den analogen Eingangswert am Pin P0 als Balkengrafik auf der LED-Matrix an. Das Programm entspricht dem Programm von gestern.



Drücken sie auf den Taster, wechselt der angezeigte Wert von 0 auf Vollausschlag, was etwa 1023 entspricht. Ziehen Sie einmal den Widerstand heraus, erreicht die Balkengrafik bei nicht gedrücktem Taster nicht mehr den Wert 0. Es wird immer ein undefinierbarer Wert in etwa 20% der Balkenhöhe angezeigt, der am nun offenen analogen Eingang anliegt.

15. Tag

## 15. Tag

## Heute im Adventskalender

· Krokodilklemmenkabel

## Wechselblinklicht mit einstellbarer Geschwindigkeit

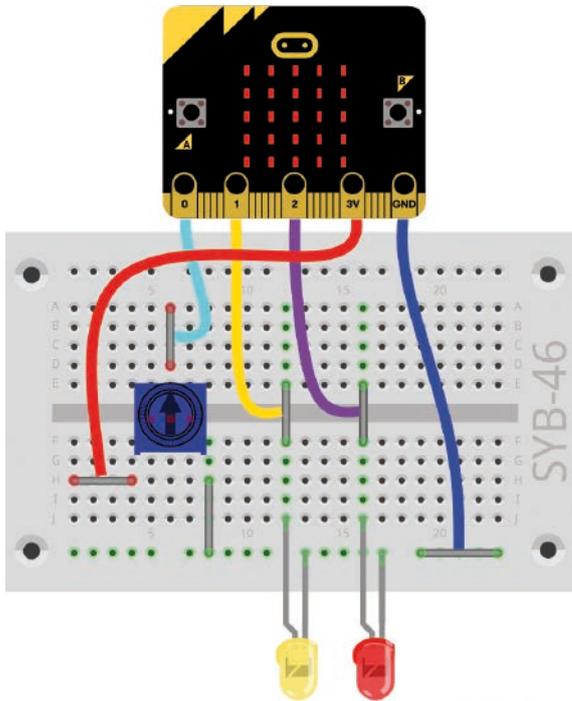
**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 5x Krokodilklemmenkabel, 1x Potentiometer, 1x LED rot, 1x LED gelb, 6x Drahtbrücke blank

## Das Programm

Das Programm `microbit-14.hex` lässt zwei LEDs abwechselnd blinken. Die Blinkgeschwindigkeit lässt sich mit dem Potentiometer einstellen.

## So funktioniert das Programm

In der **dauerhaft**-Schleife werden abwechselnd die beiden LEDs an den Pins P1 und P2 eingeschaltet. Die andere LED wird gleichzeitig ausgeschaltet. Die Wartezeiten zwischen den Schaltvorgängen sind nicht fest vorgegeben, sondern an diesen Stellen wird jedes Mal der analoge Wert von Pin P0 eingelesen. Dieser wird über das Potentiometer eingestellt und kann Werte zwischen 0 und 1023 annehmen. Damit ergeben sich Wartezeiten zwischen 0 und etwa einer Sekunde.



fritzing

beim Start

dauerhaft

```

schreibe digitalen Wert von Pin P1 auf 0
schreibe digitalen Wert von Pin P2 auf 1
pausiere (ms) analoge Werte von Pin P0
schreibe digitalen Wert von Pin P1 auf 1
schreibe digitalen Wert von Pin P2 auf 0
pausiere (ms) analoge Werte von Pin P0

```

## 16. Tag

### Heute im Adventskalender

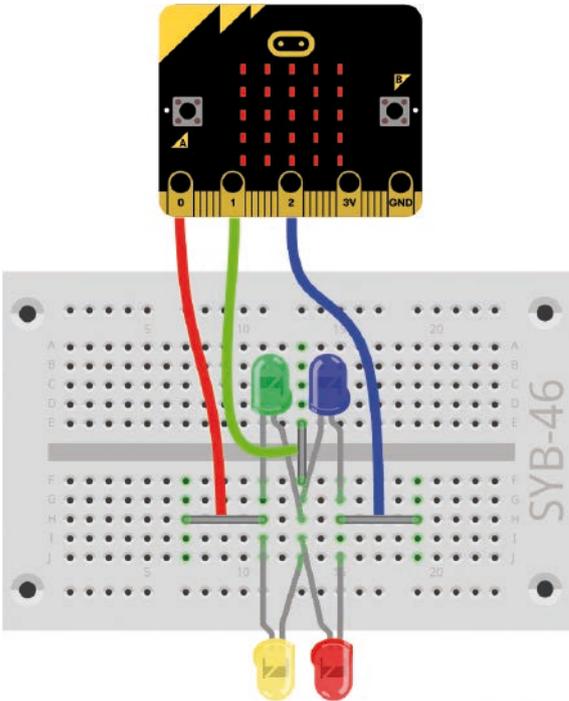
• LED blau mit Vorwiderstand

16. Tag

### Bunte Blinkmuster mit vier LEDs

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 3x Krokodilklemmenkabel, 1x Potentiometer, 1x LED rot, 1x LED gelb, 1x LED grün, 1x LED blau, 3x Drahtbrücke blank

beim Start



fritzing

An den drei Anschlusspins P0, P1 und P2 lassen sich theoretisch nur drei LEDs anschließen, die unterschiedlich blinken. Schaltet man die Anschlüsse jedoch wechselseitig als +3 V und 0 V ergeben sich mehrere mögliche Kombinationen.

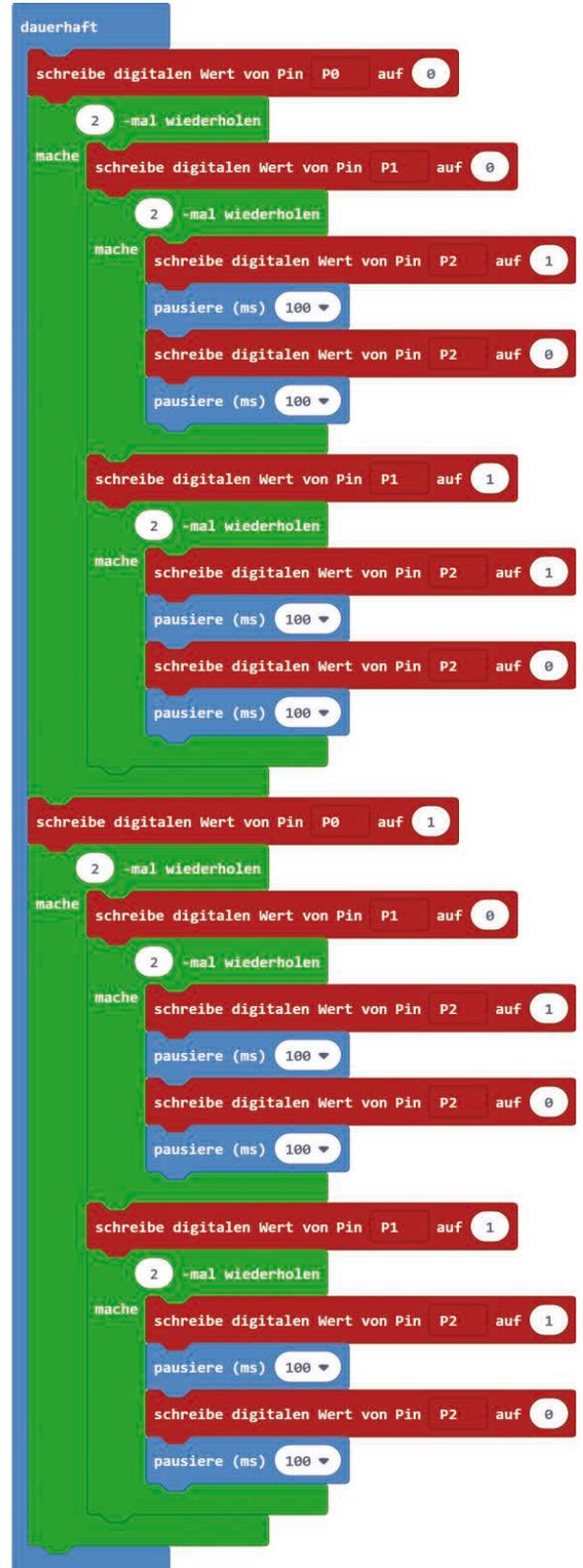
Die LEDs sind diesmal nicht an der Masseleitung des micro:bit angeschlossen, sondern immer zwischen zwei der Pins P0, P1 und P2. Die in der Abbildung oberen LEDs haben ihre Kathoden auf der linken Seite, die unteren LEDs auf der rechten Seite.

### Das Programm

Das Programm `microbit-16.hex` lässt vier LEDs abwechselnd blinken.

### So funktioniert das Programm

In der **dauerhaft**-Schleife liegen ineinander geschachtelte weitere Schleifen, die die drei Anschlusspins abwechselnd auf 0 und 1 setzen. Eine LED leuchtet immer dann, wenn der Pin an der Anode auf 1 geschaltet und der Pin an der Kathode auf 0 ist.



## 17. Tag

### Heute im Adventskalender

- Knete blau

### Reaktionsspiel

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 4x Krokodilklemmenkabel, 4x Knete

### Das Programm

Das Programm `microbit-17.hex` ist ein Spiel, in dem zwei Spieler möglichst schnell reagieren müssen. Wer zuerst seinen Sensorkontakt berührt, wenn das Herz aufleuchtet, bekommt einen Punkt. Wer fünf Punkte hat, gewinnt das Spiel. Jeder Spieler hat einen Knetekontakt für den Sensor. Dazu gibt es für jeden einen Massekontakt, den entweder beide in der einen Hand halten oder Sie legen ihn auf eine leitfähige Oberfläche, zum Beispiel eine Metallplatte, die beide Spieler berühren können.

Am besten halten beide Spieler den Massekontakt immer in der Hand und berühren dann mit der anderen Hand im richtigen Moment den Sensorkontakt. Dabei müssen Sie darauf achten, die Sensorkontakte, wenn Sie sie nicht berühren, auf einer nichtleitfähigen Oberfläche abzulegen, damit sie keine Verbindung zum Massekontakt haben.

### Die Spielregeln

- Am Anfang blinken zufällig Symbole.
- Wenn das Herz erscheint, müssen die Spieler schnell ihren Sensorkontakt berühren. Wer seinen Kontakt zuerst berührt, bekommt einen Punkt.
- Berührt jemand den Kontakt zu früh, bekommt der Gegner den Punkt. In jeder Runde wird nur ein Punkt vergeben.
- Hat ein Spieler fünf Punkte, ist das Spiel zu Ende.
- Die LEDs zeigen am Ende die Punkte an.

### So funktioniert das Programm

Die Variablen im Spiel:

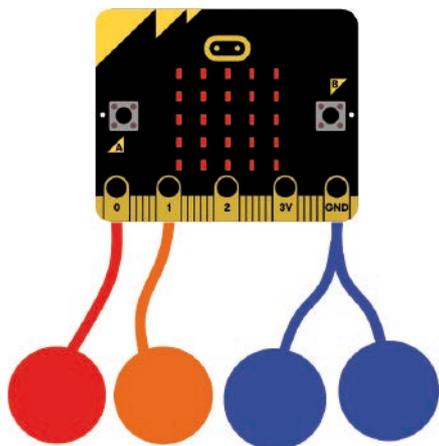
- **A** - Punkte des Spielers A.
- **B** - Punkte des Spielers B.
- **Ende** - Logikwert, der angibt, ob das Spielende erreicht ist. In diesem Fall werden keine Sensorkontakte mehr abgefragt, damit weitere Berührungen keine weiteren Punkte geben.
- **index** - Schleifenzähler.
- **Punkt** - Logikwert, der angibt, ob in einer Spielrunde noch ein Punkt zu vergeben ist.
- **Spiel** - Logikwert, der angibt, ob das Herz gerade leuchtet. In diesem Moment bekommt der Spieler beim Berühren seines Sensorkontaktes einen Punkt. Andernfalls, also wenn der Sensorkontakt zu früh berührt wird, bevor das Herz leuchtet, bekommt der Gegner den Punkt.

Das Spiel läuft in einem großen **beim Start**-Block. Am Anfang werden die Punktestände beider Spieler auf 0 gesetzt und die Variablen **Ende** und **Spiel** auf **falsch**. Das bedeutet: Jetzt ist nicht der richtige Zeitpunkt, den eigenen Sensorkontakt zu berühren. Tut man es doch, bekommt der Gegner den Punkt.

Das Spiel läuft so lange, bis einer der Spieler fünf Punkte geschafft hat. Die Hauptschleife des Spiels wird also solange wiederholt, wie beide Spieler noch weniger als fünf Punkte haben. Die Bedingung in der **während ... mache ...**-Schleife besteht aus einem Block **... und ...** und zwei Blöcken **... < 5**.

Beim Start einer jeden Spielrunde wird als Erstes die Variable **Punkt** auf **wahr** gesetzt. Das bedeutet, jetzt kann wieder ein Punkt gemacht werden. Sowie einer der Spieler in der Runde einen Punkt bekommt, wird diese Variable auf **falsch** gesetzt, damit kein zweiter Punkt in der Runde vergeben wird.

In jeder Runde blinken, bevor das Herz erscheint, nacheinander drei verschiedene Symbole auf. Damit das Spiel nicht vorhersehbar ist, wird diese Folge zufällig bis zu viermal wiederholt.



Nachdem die Symbole geblinkt haben, beginnt der Zeitpunkt, an dem man durch Berühren des eigenen Sensorkontaktes Punkte machen kann. Dazu wird die Variable **Spiel** auf **wahr** gesetzt und das Herz-Symbol angezeigt. Nach einer Wartezeit von einer Sekunde wird die Variable **Spiel** wieder auf **falsch** gesetzt. Hat während dieser Zeit keiner der beiden Spieler seinen Sensorkontakt berührt, gibt es in dieser Runde keinen Punkt und die nächste Runde startet.

Punkte werden immer dann vergeben, wenn ein Spieler einen Sensorkontakt berührt. Berührt der Spieler A (links) den Sensorkontakt P0 und ist das Spielende noch nicht erreicht, überprüft eine **wenn ... dann ... ansonsten ...**-Abfrage die Variable **Spiel**. Ist sie **wahr**, bekommt der Spieler A den Punkt, falls noch einer zu vergeben ist, denn er hat im richtigen Moment seinen Sensorkontakt berührt. Ist diese Variable **falsch**, bekommt der Spieler B den Punkt, falls noch einer zu vergeben ist. Spieler A hat seinen Sensorkontakt außerhalb der Zeit berührt, während der das Herz leuchtet.

Im Fall **dann ...** wird abgefragt, ob die Variable **Punkt** noch **wahr** ist, ob also in dieser Runde noch ein Punkt zu vergeben ist. In diesem Fall wird der Punktestand **A** um 1 erhöht und die Variable **Punkt** auf **falsch** gesetzt. Dadurch wird sichergestellt, dass nur die erste Berührung eines Sensorkontaktes als Punkt gewertet wird. Berührt der andere Spieler seinen Sensorkontakt kurz danach, bekommt er keinen Punkt mehr.

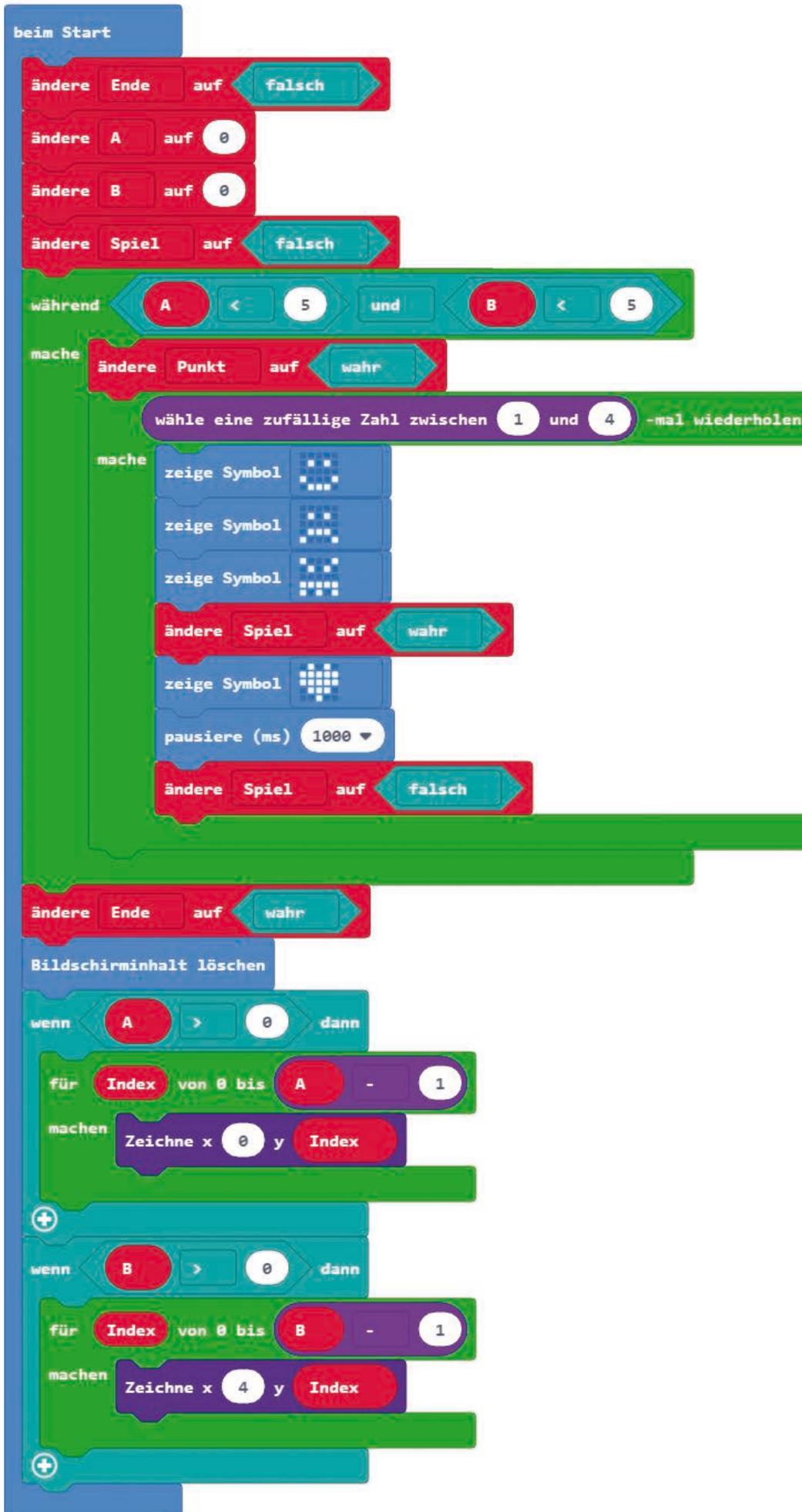
Auch im Fall **ansonsten ...**, wenn die Variable **Spiel** den Wert **falsch** hat, soll mit der Variable **Punkt** überprüft werden, ob in dieser Runde noch ein Punkt zu vergeben ist. Diesen Punkt bekommt aber der Gegner. Der Punktestand des Spielers B wird um 1 erhöht.

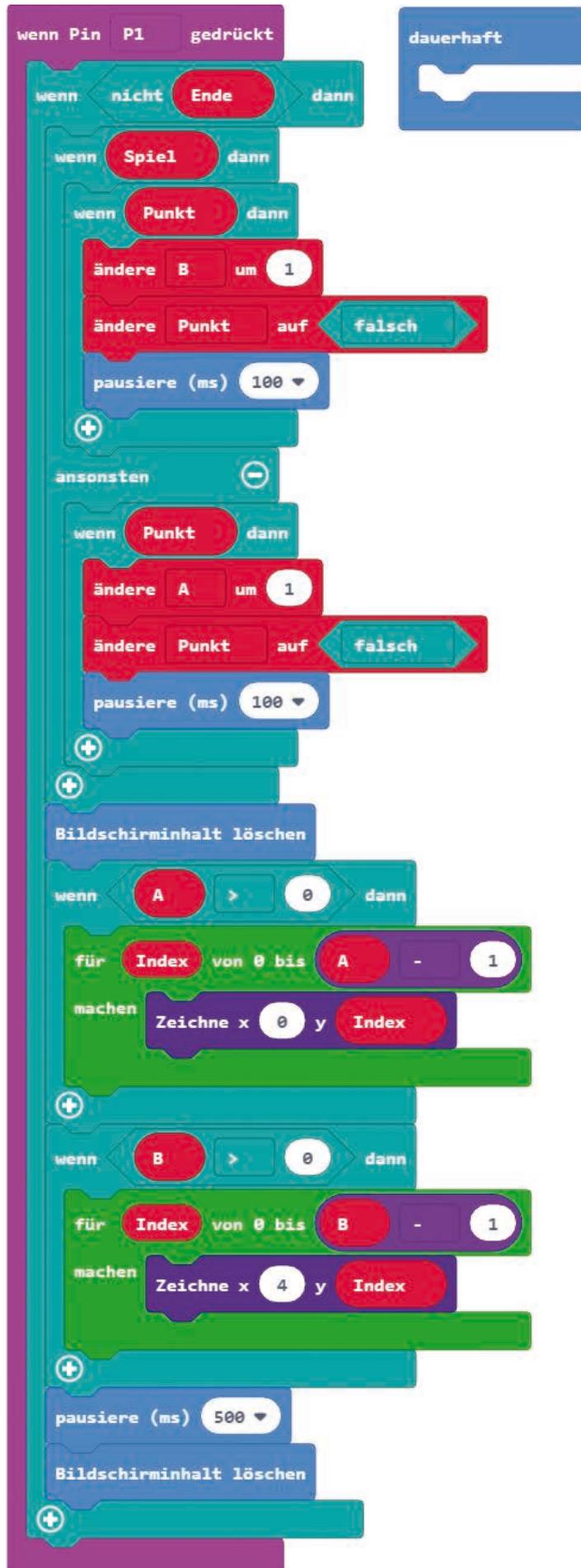
Berührt der Spieler B seinen Sensorkontakt P1, passiert im Prinzip das Gleiche. Im Fall, dass der Spieler B im richtigen Moment seinen Sensorkontakt berührt, soll der Punktestand **B** um 1 steigen. Im anderen Fall, wenn er den Sensorkontakt zu früh berührt, soll der Punktestand **A** um 1 steigen.

Die Hauptschleife des Spiels läuft solange, bis ein Spieler fünf Punkte hat, danach ist das Spiel beendet und die Punkte werden durch zwei Balken auf der LED-Matrix angezeigt. Dazu wird als Erstes hinter der Schleife mit einem Block **Bildschirminhalt löschen** das zuletzt angezeigte Herz-Symbol gelöscht.

Wenn der Spieler A mindestens einen Punkt gemacht hat, sollen in der linken Spalte der LED-Matrix bei der x-Koordinate 0 eine bis fünf LEDs leuchten. Um eine bis fünf LEDs leuchten zu lassen, die die y-Koordinaten 0 bis 4 haben, steht im Schleifenzähler der für **index von 0 bis ...**-Schleife ein Block **... - 1**. Ein Block **Zeichne x 0 y index** aus der Gruppe **LED** lässt dann innerhalb der Schleife genau so viele LEDs in der Spalte 0 aufleuchten, wie der Platzhalter **A** angibt. Wenn der Spieler B mehr als 0 Punkte hat, zeigen LEDs in der Spalte 4 der LED-Matrix seine Punkte auf die gleiche Weise an.

Die Punkte der Spieler werden nicht nur am Spielende, sondern auch jedes Mal, nachdem eine Taste gedrückt wurde, 500 Millisekunden lang angezeigt.





## 18. Tag

## Heute im Adventskalender

- 15-kOhm-Potentiometer

## Spieleprogrammierung mit Sprites

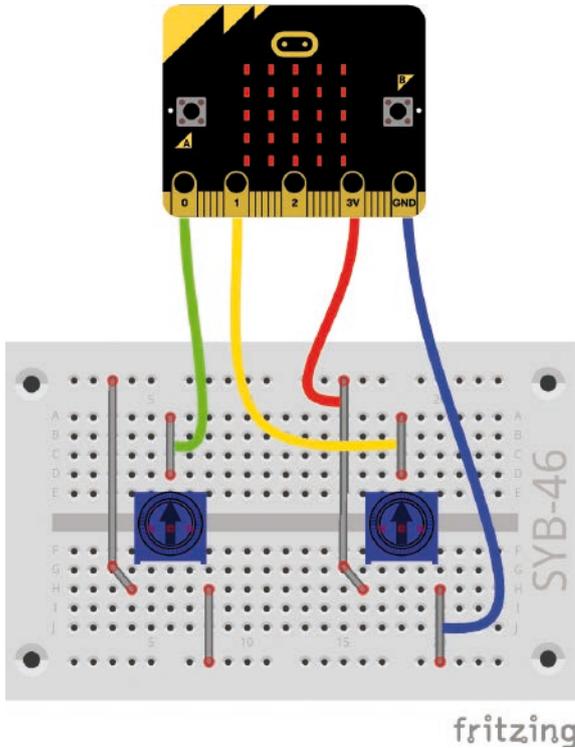
**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 4x Krokodilklemmenkabel, 2x Potentiometer, 6x Drahtbrücke blank

Die in der Abbildung obere Kontaktleiste des Steckbretts verbindet die beiden Potentiometer mit +3 V, die untere Kontaktleiste mit GND.

## Das Programm

Das Programm `microbit-18.hex` ist ein einfaches Spiel, bei dem es darum geht, mit einem einfachen Gamecontroller aus zwei Potentiometern ein Sprite zu steuern, das eine leuchtende LED einfängt. Das Programm zeigt eine neuartige Programmierung der LED-Matrix mit Sprites, wie in professionellen Computerspielen.

Ein Sprite ist ein bewegtes Objekt in einem Computerspiel, das auf PCs oder Spielkonsolen direkt vom Grafikprozessor gesteuert und über das Spielfeld oder ein Hintergrundbild bewegt werden kann. Durch diese Technik wurden die ersten schnellen Computerspiele überhaupt erst möglich, da die damalige Hardware nicht schnell genug war, ein bewegtes Komplettbild in Echtzeit zu animieren. Im Makecode-Editor auf dem micro:bit ist ein Sprite eine einzelne LED, die über verschiedene Blöcke scheinbar auf der LED-Matrix bewegt werden kann. Auf diese Weise lassen sich Animationen einfacher steuern als über die Blöcke aus der Gruppe **LED**, wo man bei jeder Bewegung die LED an der alten Position ausschalten, die neue Position berechnen und die LED dort wieder einschalten muss. Das englische Wort **sprite** bedeutet so etwas Ähnliches wie Kobold oder Geist.



beim Start

dauerhaft

Bildschirminhalt löschen

ändere **x** auf wähle eine zufällige Zahl zwischen 0 und 4ändere **y** auf wähle eine zufällige Zahl zwischen 0 und 4ändere **Ziel** auf erzeuge Sprite an Position x: **x** y: **y**ändere **sprite** auf erzeuge Sprite an Position x: 0 y: 0während nicht **sprite** touching **Ziel** ?mache **sprite** stelle **x** ein auf round analoge Werte von Pin **P0** ÷ 250**sprite** stelle **y** ein auf round analoge Werte von Pin **P1** ÷ 250delete **sprite**delete **Ziel**

zeige Symbol

pausiere (ms) 500

### So funktioniert das Programm

In jedem Durchlauf der Endlosschleife werden zuerst zwei zufällige Koordinaten  $x$  und  $y$  generiert und dort das Sprite **Ziel** erzeugt, das der Spieler einfangen muss. Dazu wird ein **Block erzeuge Sprite an Position  $x \dots y \dots$**  aus der Gruppe **Spiel** verwendet. Diese Gruppe ist erst nach einem Klick auf **Fortgeschritten** in der Liste der Blockgruppen zu finden.

Ein Sprite wird in einer Variablen gespeichert und kann darüber später bewegt werden. Eine Variable **sprite** wird mit Verwendung des ersten Sprite-Blocks automatisch angelegt. Die Variable **Ziel** für das zweite Sprite im Spiel muss wie jede andere Variable manuell angelegt werden.

Das Sprite **sprite**, das der Spieler über die Potentiometer bewegt, wird an der Position  **$x:0 y:0$**  erzeugt.

Solange sich die Sprites nicht berühren, kann das Sprite **sprite** bewegt werden. Der Block **... touching ...** prüft, ob sich zwei Sprites berühren. Zusammen mit einem Block **nicht ...** aus der Gruppe **Logik** bildet er die Bedingung für eine **während ... mache ...**-Schleife. Innerhalb dieser Schleife werden die beiden analogen Werte der Potentiometer ausgelesen, durch 250 geteilt und gerundet, sodass sich Ganzzahlwerte zwischen 0 und 4 ergeben.

Mit den Blöcken **sprite stelle  $x/y$  ein auf ...** wird das Sprite an die mit den Potentiometern eingestellte Position gesetzt.

Wenn sich beide Sprites berühren, endet die Schleife, die Sprites werden gelöscht und ein Häkchen auf der LED-Matrix angezeigt. Nach einer Wartezeit von 500 Millisekunden beginnt die **dauerhaft**-Schleife mit der nächsten Spielrunde.

19. Tag

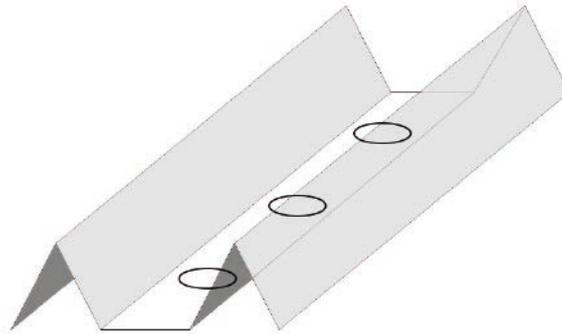
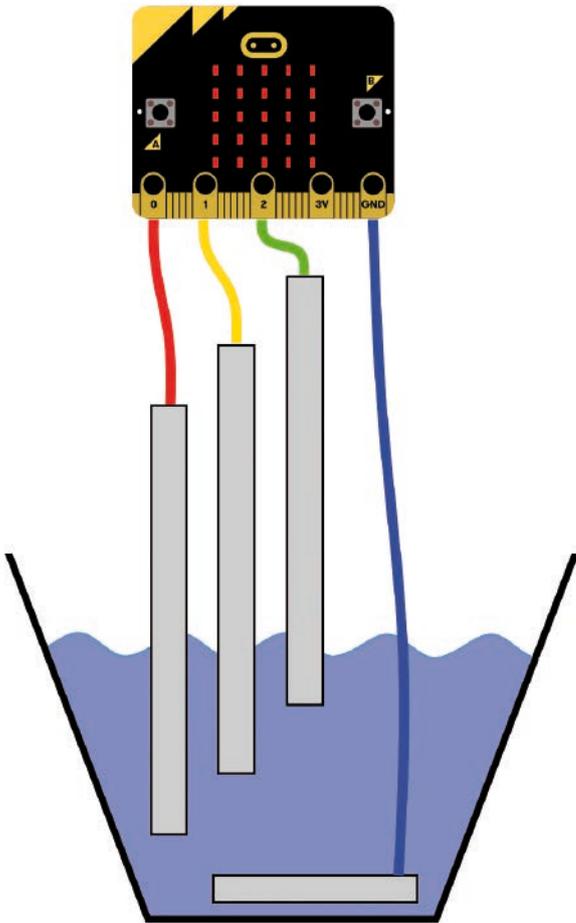
## 19. Tag

## Heute im Adventskalender

- 2x Drahtelektroden

**Wasserstandssensor**

Aus Drahtelektroden bauen wir in diesem Experiment einen Sensor, der die Füllhöhe einer Flüssigkeit in einem Glas anzeigt. Die Elektroden werden an einer Brücke über dem Glas aufgehängt. Falten Sie diese Brücke aus der Ausschneidevorlage auf der Rückseite des Adventskalenders. Klemmen Sie drei Drahtelektroden mit den Krokodilklemmenkabeln so an, dass sie unterschiedlich tief in das Glas reichen.



Biegen Sie für den Massekontakt aus der vierten Drahtelektrode einen Kreis, legen Sie diesen auf den Boden des Wasserglases und beschweren Sie die Elektrode dort mit einer Münze oder einem anderen Metallstück, damit sie auf dem Boden liegen bleibt und nicht nach oben treibt, sodass der Massekontakt auch bei ganz wenig Wasser im Glas hergestellt wird.

**Benötigte Teile:** 1x micro:bit, 4x Krokodilklemmenkabel, 4x Drahtelektroden

**Das Programm**

Das Programm `microbit-19.hex` prüft nacheinander immer wieder die drei Pins P0, P1 und P2, um den Wasserstand im Glas zu ermitteln.



### So funktioniert das Programm

In der **dauerhaft**-Schleife wird am Anfang jedes Durchlaufs die Variable **Wasser** auf 0 gesetzt, um sie dann über die unterschiedlich tief in das Glas reichenden Elektroden neu zu setzen.

Nacheinander werden die drei Elektroden an den Pins P0, P1 und P2 abgefragt. Ist die Elektrode am Pin P0 im Wasser, ist sie mit Masse verbunden. Der Pin wird als **gedrückt** erkannt. In diesem Moment muss das Glas zu mindestens einem Drittel mit Wasser gefüllt sein. Die Variable **Wasser** wird auf **33** gesetzt. Diese Zahl steht für den Füllstand des Glases in Prozent. Je nach Eintauchtiefe der Elektrode können Sie diesen Wert auch ändern.

Auf die gleiche Weise werden die anderen beiden Elektroden, die weniger tief in das Glas eintauchen, abgefragt. Bei der mittleren Elektrode am Pin P1 wird ein Füllstand von 66% angenommen, bei der oberen Elektrode am Pin P2 gilt das Glas als voll.

Am Ende jedes Schleifendurchlaufs zeigt das Programm auf der LED-Matrix ein Balkendiagramm an. Der angezeigte Wert entspricht der Variable **Wasser**, und die Skala soll bis 100 reichen. Stehen alle drei Elektroden im Wasser, leuchten alle LEDs der Matrix.

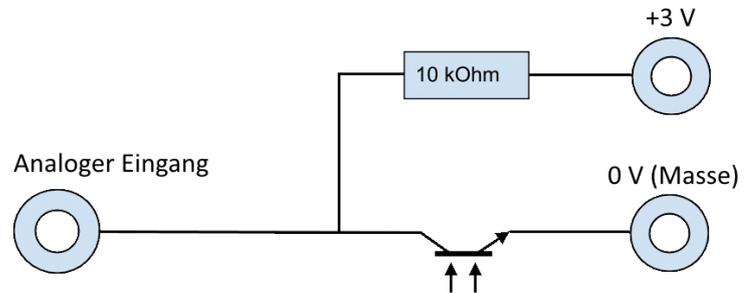
## 20. Tag

### Heute im Adventskalender

· 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 erzielen. Je heller das Licht auf den Fototransistor fällt, desto geringer wird 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

Auf dem micro:bit ist ein solcher Helligkeitssensor eingebaut, der allerdings nicht besonders feinfühlig reagiert. Heute ist im Adventskalender ein Fototransistor, mit dem sich ein Helligkeitssensor bauen lässt, der dazu noch den Vorteil hat, dass er weiter weg vom micro:bit aufgestellt werden kann, um Helligkeitswerte zu übermitteln.

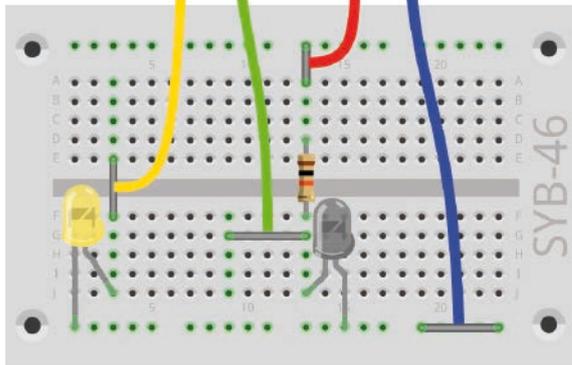
#### Nachtlicht in der Dunkelheit

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 4x Krokodilklemmenkabel, 1x LED gelb, 1x Fototransistor, 1x Widerstand 10 kOhm (braun - schwarz - orange), 4x Drahtbrücke blank

Wenn man nachts aufwacht, findet man sich in einem völlig dunklen Zimmer schwer zurecht. Ein Nachtlicht mit einer einzelnen LED reicht oft schon als Orientierungshilfe. Solche Nachtlichter schalten sich bei Dunkelheit automatisch ein und wenn es hell wird, wieder aus. Sie leuchten selber nur so schwach, dass sie den Helligkeitssensor nicht beeinflussen.

#### Das erste Programm

Das Programm `microbit-20-01.hex` verwendet den eingebauten Helligkeitssensor auf dem micro:bit und benötigt keinen zusätzlichen Fototransistor.



fritzing

### So funktioniert das erste Programm

Der eingebaute Helligkeitssensor liefert je nach Helligkeit einen analogen Wert, der umso höher ist, je mehr Licht darauf fällt. Liegt dieser Wert unter 100, soll die LED am Pin P0 leuchten. Ist das Umgebungslicht heller, wird sie ausgeschaltet. Der aktuelle Wert des Helligkeitssensors kann jederzeit über den Block **Lichtstärke** aus der Gruppe **Eingabe** ausgelesen werden.

### Das zweite Programm

Das Programm `microbit-20-02.hex` verwendet einen extern angeschlossenen Fototransistor.



Ein Fototransistor liefert je nach Helligkeit einen analogen Wert zwischen 0 und 1023, der allerdings umso höher ist, je weniger Licht auf den Fototransistor fällt.

### So funktioniert das zweite Programm

Das Programm liest den analogen Wert am Pin P1 und schaltet die LED am Pin P0 ein, wenn dieser Wert größer als 100 ist. Je nach Umgebungshelligkeit können Sie den Vergleichswert in der **wenn... dann ... ansonsten ...**-Abfrage auch ändern, falls sich das Nachtlicht zu früh oder zu spät einschaltet.

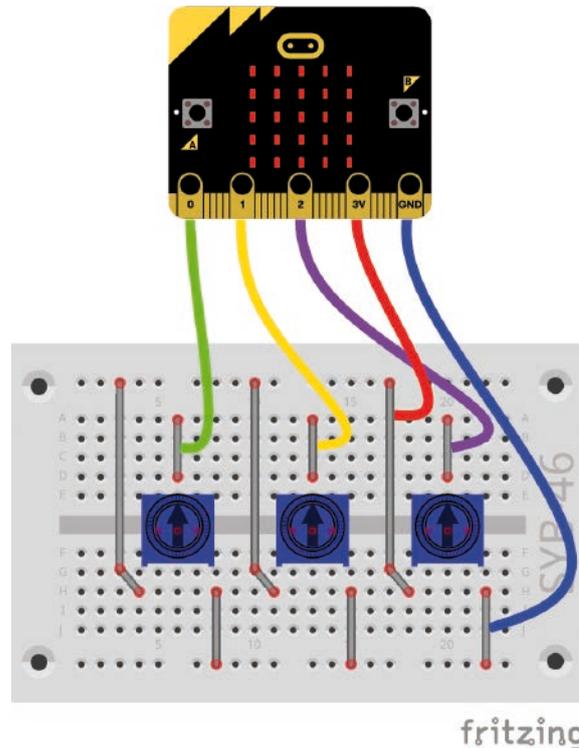
## 21. Tag

### Heute im Adventskalender

· 15-kOhm-Potentiometer

### Lauflicht auf der LED-Matrix

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 5x Krokodilklemmenkabel, 3x Potentiometer, 9x Drahtbrücke (5 davon blank)

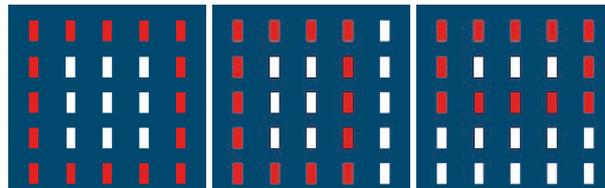


### Das Programm

Das Programm `microbit-21.hex` erzeugt je nach Einstellung der Potentiometer verschiedene Lauflichteffekte auf der LED-Matrix.

### So funktioniert das Programm

Am Anfang jedes Durchlaufs der **dauerhaft**-Schleife werden die analogen Werte der drei Potentiometer ausgelesen. Die ersten beiden Potentiometer geben die Breite ( $x$ ) und Höhe ( $y$ ) des Rechtecks an, in dem das Lauflicht läuft. Die analogen Werte liegen im Bereich zwischen 0 und 1023 und werden durch 250 geteilt, dadurch ergeben sich Werte zwischen 0 und 4, für die fünf Spalten und Zeilen der LED-Matrix. Diese Werte werden in den Variablen  $x$  und  $y$  gespeichert.



Der Wert des dritten Potentiometers gibt die Geschwindigkeit des Lauflichts an. Dieser Wert wird durch 4 geteilt. Damit ergeben sich zwischen den einzelnen Schaltvorgängen Wartezeiten zwischen 0 und 255 Millisekunden. Dieser Wert wird in der Variablen **z** gespeichert.



Anschließend laufen vier Schleifen, die nacheinander die obere, die rechte, die untere und die linke Seite des Rechtecks auf der LED-Matrix darstellen. Jede Schleife enthält eine **wenn ... dann ...**-Abfrage, die prüft, ob der aktuelle Schleifenzähler **Index** noch kleiner oder gleich der eingestellten Größe des Rechtecks ist. In diesem Fall leuchtet die entsprechende LED, und das Programm wartet die eingestellte Zeit. Die Berechnungsformeln in den beiden letzten Schleifen sind nötig, da hier rückwärts gezählt werden muss.

Am Ende des Durchlaufs nach der letzten LED wird noch einmal kurz gewartet. Danach wird der Bildschirminhalt gelöscht, und der nächste Durchlauf startet.

```

beim Start
dauerhaft
  ändere x auf analoge Werte von Pin P0 ÷ 250
  ändere y auf analoge Werte von Pin P1 ÷ 250
  ändere z auf analoge Werte von Pin P2 ÷ 4
  für Index von 0 bis 4
    machen
      wenn Index ≤ x dann
        Zeichne x Index y 0
        pausiere (ms) z
  für Index von 0 bis 4
    machen
      wenn Index ≤ y dann
        Zeichne x x y Index
        pausiere (ms) z
  für Index von 0 bis 4
    machen
      wenn Index ≤ x dann
        Zeichne x x - Index y y
        pausiere (ms) z
  für Index von 0 bis 4
    machen
      wenn Index ≤ y dann
        Zeichne x 0 y y - Index
        pausiere (ms) z
  pausiere (ms) z
  Bildschirminhalt löschen

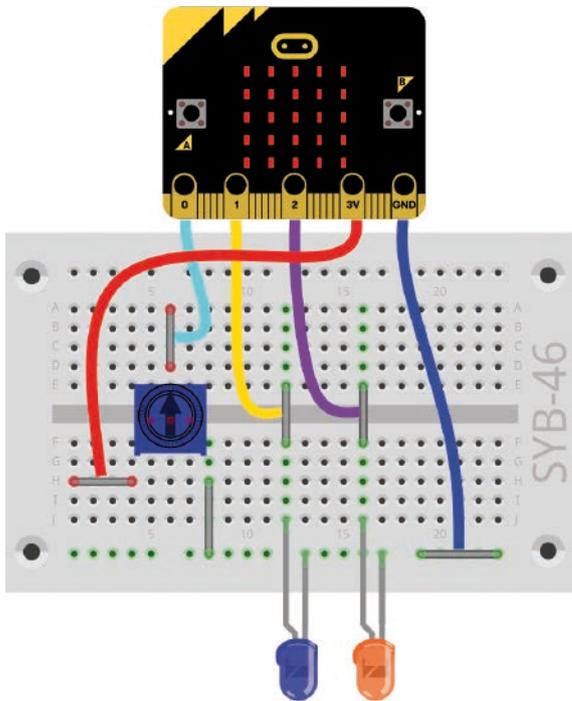
```

22. Tag

## 22. Tag

## Heute im Adventskalender

- LED orange mit Vorwiderstand



fritzing

## Zahlen raten

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 5x Krokodilklemmenkabel, 1x Potentiometer, 1x LED blau, 1x LED orange, 6x Drahtbrücke blank

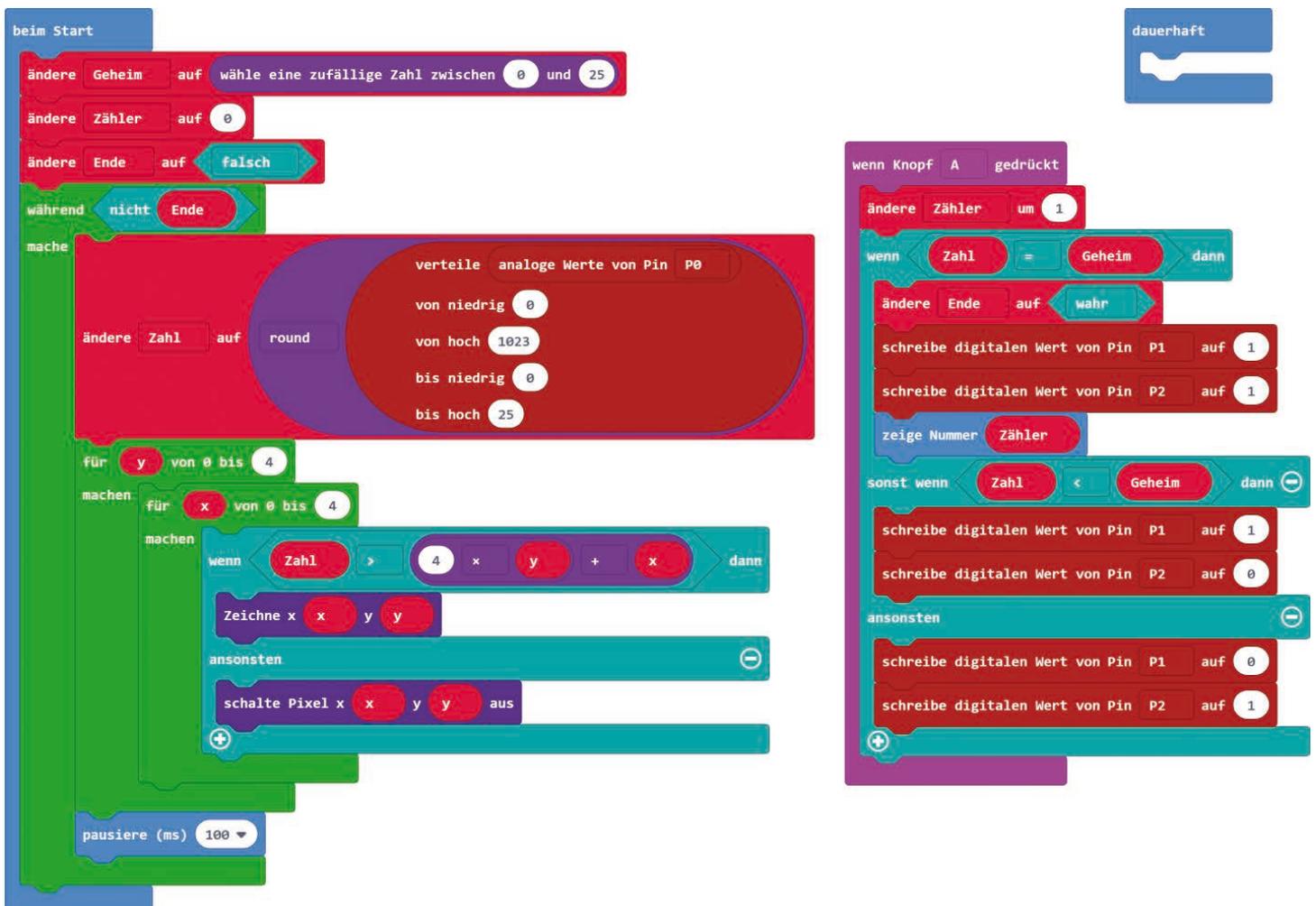
## Das Programm

Das Programm `microbit-22.hex` ist ein einfaches Ratespiel. Dabei soll eine zufällig erzeugte Zahl vom Spieler in möglichst wenigen Schritten erraten werden. Die Zahlen werden auf der LED-Matrix durch unterschiedlich viele Punkte angezeigt und über das Potentiometer eingegeben. Ein Druck auf die Taste A gibt den Tipp ab. Zwei LEDs zeigen, ob der letzte Tipp kleiner oder größer als die gesuchte Zahl ist. Am Ende zeigt die LED-Matrix, wie viele Tipps benötigt wurden, um die Zahl zu erraten.

## So funktioniert das Programm

Die Variablen im Spiel:

- **Ende** - Logikwert, der angibt, ob das Spielende erreicht ist.
- **Geheim** speichert die geheime Zahl
- **Index** - Schleifenzähler
- **x** - x-Wert für die Koordinaten auf der LED-Matrix zur Anzeige der eingestellten Zahl
- **y** - y-Wert für die Koordinaten auf der LED-Matrix zur Anzeige der eingestellten Zahl



- **Zahl** speichert die Zahl, die der Spieler einstellt und rät
- **Zähler** speichert die Anzahl der Rateversuche

Beim Start des Programms wird eine geheime Zahl zwischen 0 und 25 zufällig ausgewählt und in der Variable **Geheim** gespeichert. Der **Zähler** wird auf 0 und die Variable **Ende** auf **falsch** gesetzt.

Solange die Variable **Ende** noch nicht wahr ist, wird der Wert des Potentiometers abgefragt, um einen Tipp einzustellen. Der Block **verteile ...** aus der Gruppe **Pins** rechnet die möglichen Eingangswerte zwischen 0 und 1023 automatisch in passende Ausgangswerte zwischen 0 und 25 für die 25 LEDs der Matrix um. Liegt am Eingang P0 der Maximalwert des Potentiometers 1023 an, sollen alle 25 LEDs eingeschaltet und die Zahl 25 getippt werden.

Anschließend wird die eingestellte Zahl über zwei ineinander geschachtelte Schleifen auf der LED-Matrix dargestellt. Die äußere Schleife steuert die Zeilen mit je fünf LEDs, die innere die fünf LEDs einer Zeile.

Nach einer kurzen Wartezeit von 100 Millisekunden wird der Wert des Potentiometers erneut ausgewertet.

Drückt der Spieler die Taste, wird die eingestellte Zahl als Tipp abgegeben. Die Anzahl abgegebener Tipps in der Variable **Zähler** wird um 1 erhöht.

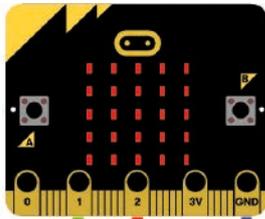
Falls der abgegebene Tipp **Zahl** gleich der gesuchten Zahl **Geheim** ist, wird die Variable **Ende** auf **wahr** gesetzt. Nach diesem Durchlauf wird die Schleife kein weiteres Mal wiederholt. Das Spiel ist zu Ende. Beide LEDs leuchten, und die Anzahl benötigter Tipps in der Variable **Zähler** wird als Ziffer auf der LED-Matrix angezeigt.

Ist der abgegebene Tipp kleiner als die gesuchte Zahl, wird die LED am Pin P1 eingeschaltet, ist sie größer, leuchtet die LED am Pin P2. So erfährt der Spieler, in welcher Richtung er weiter tippen muss.

## 23. Tag

## Heute im Adventskalender

LED lila mit Vorwiderstand



## Space Invaders

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 3x Krokodilklemmenkabel, 1x LED lila, 1x LED orange, 3x Drahtbrücke blank

## Das Programm

Das Programm `microbit-23.hex` ist eine einfache Variante des Computerspieleklassikers Space Invaders auf der LED-Matrix des micro:bit.

Die Spielregeln:

- Von oben fallen an zufälligen Positionen zwei LEDs breite feindliche Schiffe herunter.
- Das Schiff des Spielers ist eine LED groß und bewegt sich in der unteren Zeile der LED-Matrix.
- Ein Druck auf die Taste (A) bewegt das Schiff des Spielers um ein Rasterfeld nach links.
- Ein Druck auf die Taste (B) bewegt das Schiff des Spielers um ein Rasterfeld nach rechts.
- Ein Druck auf beide Tasten gleichzeitig feuert. Der Feuerstrahl soll sehr schnell von unten nach ganz oben schießen und ein feindliches Schiff, das sich in der gleichen Spalte befindet, erfassen und auflösen. Bei einem Treffer soll die LED orange flackern.
- Erreicht ein feindliches Schiff den Boden, soll die LED violett flackern.

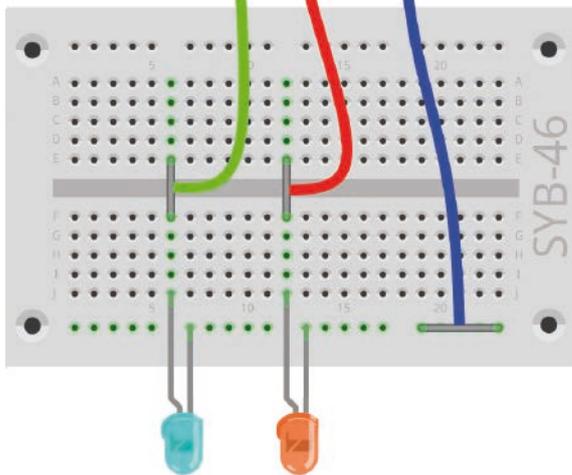
fritzing

## So funktioniert das Programm

Beim Start werden zunächst nur ein paar Variablen gesetzt. Anschließend läuft die Schleife dauerhaft. Im Gegensatz zu bisherigen Spielen hat dieses Spiel kein Ende. Sie können ewig weiterspielen.

Die Variablen im Spiel:

- **Feuer** - Schleifenzähler für den Feuerstrahl
- **Index** - Schleifenzähler
- **Invader** - Position des Invaderschiffs in x-Richtung
- **Position** - Position des Spielerschiffs in dem Moment, wo der Feuerstrahl ausgelöst wird
- **Spieler** - Position des Spielerschiffs in x-Richtung
- **Treffer** - Logikwert, wenn der Invader getroffen wurde



Das eigentliche Spiel läuft in einer Endlosschleife. Nachdem der x-Wert für den Start des Invaders festgelegt wurde, startet eine Schleife, die den Invader in insgesamt fünf Schritten bis in die unterste Zeile der LED-Matrix bringen soll. Die Variable **Invader** bezeichnet die Position des Invaders in x-Richtung. Da das Invaderschiff zwei LEDs breit ist, kann die Position seiner linken LED nur auf den ersten vier LEDs einer Zeile sein.

Dabei wird jedes Mal zuerst geprüft, ob der Invader vom Feuerstrahl getroffen wurde. Ist das der Fall, wird der Schleifenzähler **Index** wieder auf 0 gesetzt und der Invader startet wieder in der obersten Zeile an einer zufälligen Position.

Wurde der Invader nicht getroffen, werden die beiden LEDs ein- und nach 200 Millisekunden wieder ausgeschaltet. Danach beginnt der nächste Durchlauf eine Zeile weiter unten. Ist der Invader unten angekommen, flackert die lila LED am Pin P1 viermal kurz. Der Flackerlichteffekt entsteht dadurch, dass die Einschaltzeit in jedem Zyklus länger als die Ausschaltzeit ist.

Drückt man auf die Taste A, soll sich das Spielerschiff um ein Rasterfeld nach links bewegen. Dazu wird der Wert der Variable **Spieler** um 1 reduziert. Das soll aber nur passieren, solange dieser Platzhalter noch größer als 0 ist. Bei 0 steht das Spielerschiff in der ersten Spalte der LED-Matrix ganz links und kann sich nicht noch weiter nach links bewegen. Ein **wenn ... dann**-Block fragt innerhalb des Blocks **wenn Knopf A gedrückt** den aktuellen Wert des Platzhalters **Spieler** ab.

Auf die gleiche Weise bewegt sich das Spielerschiff nach rechts, wenn man die Taste B drückt.

Drückt man beide Tasten A und B gleichzeitig, soll das Spielerschiff feuern. Der Feuerstrahl soll sehr schnell von unten nach oben laufen und dann eine kurze Zeit lang in voller Länge leuchten.

Zuerst wird die aktuelle Position des Spielerschiffs zusätzlich in der Variablen **Feuer** gespeichert, damit es zu keinen Fehlern kommt, wenn sich das Spielerschiff während des Feuerns bewegt. Anschließend zählt eine Schleife hoch und schaltet schnell hintereinander die LEDs in der durch **Feuer** angegebenen Spalte von unten nach oben ein, was der Betrachter als kurzes Aufflackern von unten nach oben wahrnimmt.

Während der Feuerstrahl leuchtet, soll geprüft werden, ob der Invader getroffen wurde. Der Invader wurde getroffen, wenn sich das Spielerschiff und damit auch der Feuerstrahl während des Feuerns auf der gleichen X-Koordinate wie eine der beiden LEDs des Invaders befinden. Das wird in einer **wenn ... dann ...**-Abfrage abgefragt.

Wenn der Invader getroffen wurde, flackert die orange LED am Pin P2 viermal. Die Variable **Treffer** wird auf **wahr** gesetzt, was der Hauptschleife signalisiert, dass in der oberen Zeile ein neuer Invader starten muss.

Nach einer deutlich längeren Pause von 100 ms, während der alle vier LEDs des Feuerstrahls leuchten, schaltet eine weitere Schleife sie wieder aus. Diesmal ist keine Wartezeit eingebaut, die LEDs werden gleichzeitig ausgeschaltet. In Wirklichkeit wird eine nach der anderen LED ausgeschaltet, was aber so schnell geht, dass man es nicht wahrnimmt.



## 24. Tag

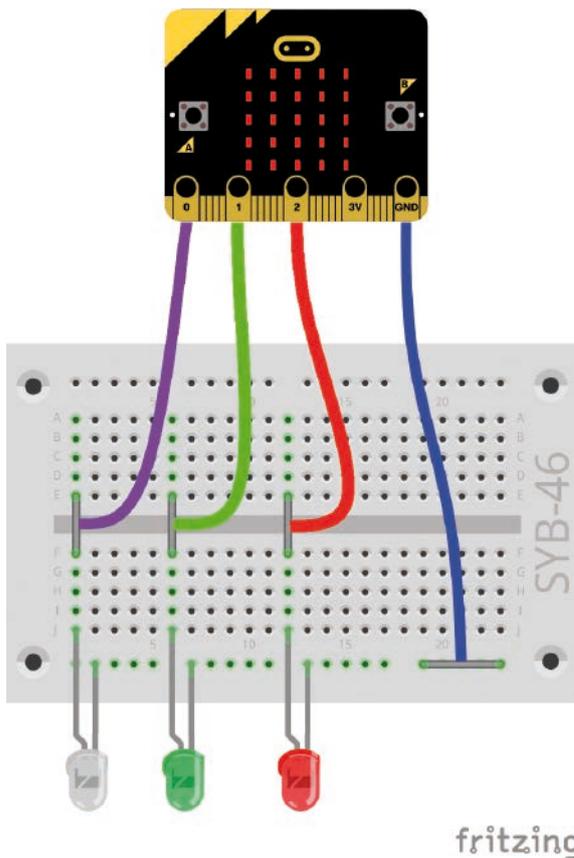
### Heute im Adventskalender

- Blink-LED mit Vorwiderstand

Als Überraschung zu Weihnachten ist heute eine LED im Adventskalender, die selbstständig blinkt, ohne dass ein Programm nötig ist. Solche LEDs werden für optische Signale verwendet, wo es darum geht, besondere Aufmerksamkeit zu erregen.

### Weihnachtsquiz

**Benötigte Teile:** 1x micro:bit, 1x Steckbrett, 4x Krokodilklemmenkabel, 1x LED rot, 1x LED grün, 1x Blink-LED, 4x Drahtbrücke blank



### Das Programm

Das Programm `microbit-24.hex` ist ein Quizspiel mit zehn weihnachtlichen Fragen, die auf der LED-Matrix angezeigt werden. Über die beiden Taster kann man auf jede Frage die Antworten Ja (Taste A) oder Nein (Taste B) geben.

### So funktioniert das Programm

Das Programm verwendet Listen, manchmal auch als Arrays bezeichnet, eine besondere Form von Variablen, die mehrere Werte in einer bestimmten Reihenfolge enthalten. Solche Listen werden mit den Blöcken **ändere Liste auf ...** für Listen mit Zahlen oder Logikwerten und **ändere text list auf ...** für Listen mit Texten aus der Gruppe **Arrays** angelegt.

Die Variablen im Spiel:

- **Antwort** - Antwort des Spielers
- **Liste** - Die Antworten auf die Fragen
- **Lösung** - Richtige Lösung der gerade angezeigten Frage
- **Punkte** - Punkte für richtige Antworten

beim Start

ändere text list auf

array of

- Der Weihnachtsbaum kommt aus Schweden
- Engel-Bengerl ist das gleiche wie Julklapp
- Die Weihnachtsgeschichte steht bei Markus
- Die Hirten brachten Gold und Weihrauch
- In Island gibt es 13 Weihnachtzwerge
- Christbaumloben ist ein Brauch aus Bayern
- Der Papst feiert am 24.12. Geburtstag
- 1906 gab es den ersten Adventskalender
- Das Himmelreich liegt in Niedersachsen
- Der Nikolaus kommt aus Bethlehem

ändere Liste auf

array of

- falsch
- wahr
- falsch
- falsch
- wahr
- wahr
- falsch
- wahr
- wahr
- falsch

ändere Zahl auf Array-Länge Liste

ändere Punkte auf 0

Zahl -mal wiederholen



mache

zeige Zeichenfolge rufe den ersten Wert ab und lösche ihn von text list

ändere Tipp auf falsch

während nicht Tipp

mache

wenn Button A ist gedrückt dann

ändere Antwort auf wahr

ändere Tipp auf wahr

+

wenn Button B ist gedrückt dann

ändere Antwort auf falsch

ändere Tipp auf wahr

+

ändere Lösung auf rufe den ersten Wert ab und lösche ihn von Liste

wenn Antwort = Lösung dann

ändere Punkte um 1

4 -mal wiederholen

mache

schreibe digitalen Wert von Pin P1 auf 1

pausiere (ms) 50

schreibe digitalen Wert von Pin P1 auf 0

pausiere (ms) 100

ansonsten

4 -mal wiederholen

mache

schreibe digitalen Wert von Pin P2 auf 1

pausiere (ms) 50

schreibe digitalen Wert von Pin P2 auf 0

pausiere (ms) 100

+

schreibe digitalen Wert von Pin P0 auf 1

wenn Punkte = 10 dann

schreibe digitalen Wert von Pin P1 auf 1

ansonsten

schreibe digitalen Wert von Pin P2 auf 1

+

zeige Nummer Punkte

- **test list** - Die Texte der Fragen
- **Tipp** - Logikwert, wenn der Spieler eine Antwort abgegeben hat
- **Zahl** - Anzahl der Fragen

Beim Start werden eine Textliste mit den zehn Fragen sowie eine weitere Liste mit zehn Logikwerten für die Antworten in der gleichen Reihenfolge angelegt. Der Block **Array-Länge ...** aus der Gruppe **Arrays** ermittelt die Länge der Liste mit den Antworten und speichert sie in der Variablen **Zahl**, die als Schleifen-zähler verwendet wird. Auf diese Weise kann das Spiel leicht um weitere Fragen ergänzt werden.

Die Hauptschleife des Programms läuft so oft, wie Elemente in der Liste sind, in der Grundeinstellung des Programms zehnmal.

Als Erstes wird der erste Text aus der Liste genommen und auf der LED-Matrix angezeigt. Danach wird er automatisch aus der Liste gelöscht.

Die Variable **Tipp** wird auf **falsch** gesetzt. Sie gibt an, ob der Spieler zur aktuellen Frage bereits einen Tipp abgegeben hat. Solange das nicht der Fall ist, werden die beiden Tasten abgefragt. Taste A setzt die Antwort auf **wahr**, Taste B auf **falsch**. Bei beiden Tasten wird die Variable **Tipp** auf **wahr** gesetzt, da der Spieler eine Antwort abgegeben hat.

Jetzt wird der erste Wert aus der Liste **Liste** geholt und in der Variablen **Lösung** gespeichert. Er enthält die richtige Antwort auf die aktuelle Frage. Ist die Antwort des Spielers gleich der Lösung, wird der Punktezähler um 1 erhöht und die grüne LED am PIN P1 blinkt viermal. Ist die Antwort dagegen nicht gleich der Lösung, blinkt die rote LED. In diesem Fall gibt es keinen Punkt.

Nachdem alle zehn Fragen beantwortet wurden, ist das Spiel zu Ende. Die Blink-LED am Pin P0 wird eingeschaltet. Hat der Spieler alle Fragen richtig beantwortet, leuchtet zusätzlich die grüne LED. Hat er dagegen keine zehn Punkte erreicht, leuchtet die rote LED. In beiden Fällen wird zum Schluss die erreichte Punktzahl auf der LED-Matrix angezeigt.

Natürlich können Sie das Spiel jederzeit um weitere Fragen erweitern. Wichtig ist nur, dass beide Listen gleich viele Elemente haben, es also auf jede Frage eine Antwort gibt. Mehr braucht im Programm nicht verändert zu werden.

Frohe Weihnachten!

### 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 micro:bit komplett zerstören. Verbindet man über einen Schalter oder eine LED zwei Pins miteinander, muss immer ein Schutzwiderstand dazwischengeschaltet werden.  
Für Logiksignale verwendet der micro:bit 3 V 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.

micro:bit unterliegt dem Copyright der Micro:bit Educational Foundation.