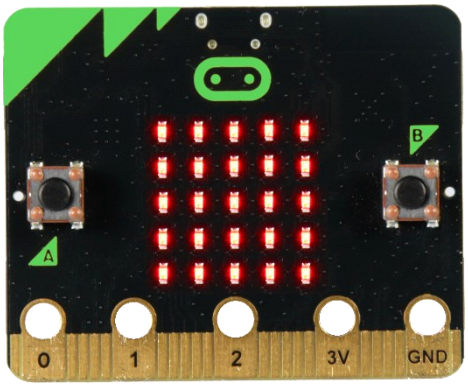
Digitales Lehren & Lernen
Didaktik der Informatik

Mikrocontroller-Programmierung mit micro:bit

Einführung für Einsteiger

Blockbasierte Programmierung mit MakeCode



Matthias Ehmann
Andreas Walter
Christoph Selbmann

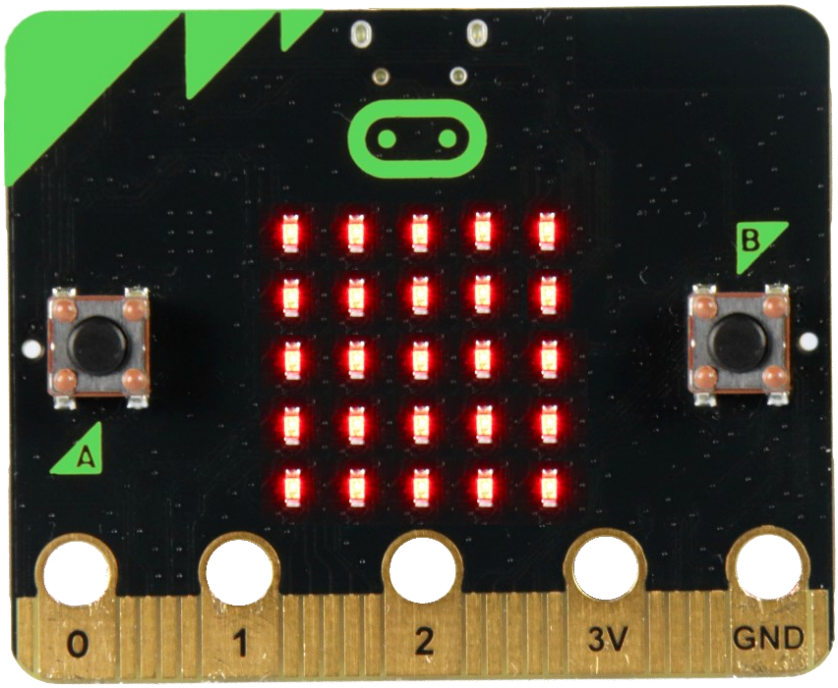
Digitales Lehren und Lernen &
Didaktik der Informatik
Universität Bayreuth
dlldi.uni-bayreuth.de

Dieses Skript gehört:

_____

Version: 01.02.2024 10:25 Uhr

Mikrocontroller-Programmierung mit micro:bit
Einführung für Einsteiger



Digitales Lehren und Lernen &
Didaktik der Informatik
Universität Bayreuth
dlldi.uni-bayreuth.de

Sicherheitshinweise

Lies dir die folgenden Punkte genau durch! Es wird an manchen Stellen im Skript noch einmal ausdrücklich auf wichtige Aspekte hingewiesen. Diese Hinweise solltest du bereits auf einem gesonderten Blatt bekommen und unterschrieben (Foto oder Scan) an uns geschickt haben.

Die Verwendung des BBC micro:bit ist einfach. Er ist jedoch so konzipiert, dass alle elektronischen Teile offen zugänglich sind. Bei einer sachgemäßen Verwendung mit Sorgfalt und Vorsicht ist die Benutzung aber ungefährlich. Deshalb beachte die folgenden Hinweise:

1. Anleitung befolgen
   Halte dich bei den folgenden Experimenten an die Anleitung. Beachte die angegebenen Schaltungen beim Aufbau der Experimente.

2. Stromversorgung
   Verwende nur das von uns gelieferte Batteriepack (und die Batterien) oder das von uns mitgelieferte USB-Kabel für die Stromversorgung deines micro:bit. Schließe auf keinen Fall beides gleichzeitig an!

   Verwende keinen portablen Batterie-Ladegeräte oder USB-Netzteile, da eine Überspannung den micro:bit zerstören könnte.

3. Lass deinen micro:bit nicht unbeaufsichtigt an der Stromversorgung stecken

4. Schaltungen stromfrei aufbauen
   Verändere keine Schaltungen, während der micro:bit mit der Stromversorgung verbunden ist. Trenne ihn vorher vom Strom.

5. Verwende nur die von uns mitgelieferten bzw. im Experiment beschriebenen Anschlussteile

6. Beschädigter micro:bit
   Falls dein micro:bit beschädigt ist, kontaktiere uns per E-Mail (Seite 1). Schließe ihn nicht mehr an die Stromversorgung an!

7. Vermeide heiße und kalte Temperaturen, lass den micro:bit nicht in der Sonne liegen

8. Bringe den micro:bit niemals mit Wasser oder nassen Händen in Berührung!

9. Vermeide Kurzschlüsse
   Berühre den micro:bit nicht mit metallischen Gegenständen, außer es wird ausdrücklich in der Anleitung verlangt (z. B. mit Krokodilklemmen).

   Verbinde niemals direkt einen Plus- mit einem Minuspol ohne einen „Verbraucher“ dazwischen! Dies könnte zu einem Kurzschluss führen und den micro:bit oder deinen Computer beschädigen.

10. Berühren der Kontaktflächen
    Der Strom, der an den Kontaktflächen anliegt, ist so gering, dass man ihn überhaupt nicht spürt! Anders als Strom aus der Steckdose ist hier die Spannung so gering, dass sie für gesunde Menschen ungefährlich ist. Berühre die Kontaktflächen trotzdem nur dann, wenn es ausdrücklich in der Anleitung beschrieben ist.

11. Trenne Kabelverbindungen nur, indem du am Stecker und nicht am Kabel ziehst

Weitere Hinweise zur Verwendung finden sich im „safety guide“ des micro:bit.

Allgemeine Informationen

Herzlich Willkommen im Kurs
„Mikrocontroller-Programmierung mit micro:bit –
Einführung für Einsteiger“

Für Kursinformationen sieh bitte
an dieser Stelle im gedruckten Skript nach!

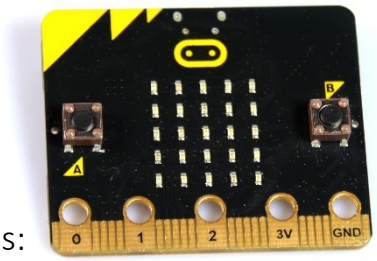
- Falls du Fragen hast:
  - Stelle sie uns bei einem unserer Termine.
  - Nutze das Forum im E-Learning-Kurs.
  - Schreibe uns eine E-Mail.
- In diesem Skript sind immer wieder Lösungen angegeben. Deine Lösungen müssen nicht immer genauso aussehen, um zu funktionieren. Oft gibt es mehrere Wege, die ans Ziel führen!
- Bitte verwende den Rücksendeaufkleber und sende das Hardware-Paket nach dem Workshop an uns zurück (siehe Datum oben)! Dieses Skripts darfst du behalten.

Inhalte

Experimente

Unboxing und erster Überblick

Was ist der micro:bit überhaupt?



- Öffne das Paket, dass du von uns erhalten hast!
- Darin befindet sich der micro:bit. Er sieht ungefähr so aus:

- Schieße das mitgelieferte Batteriepack auf der Rückseite an.

Achtung!

Beachte auf Seite 4: „2. Stromversorgung“

Spiele „Snake“

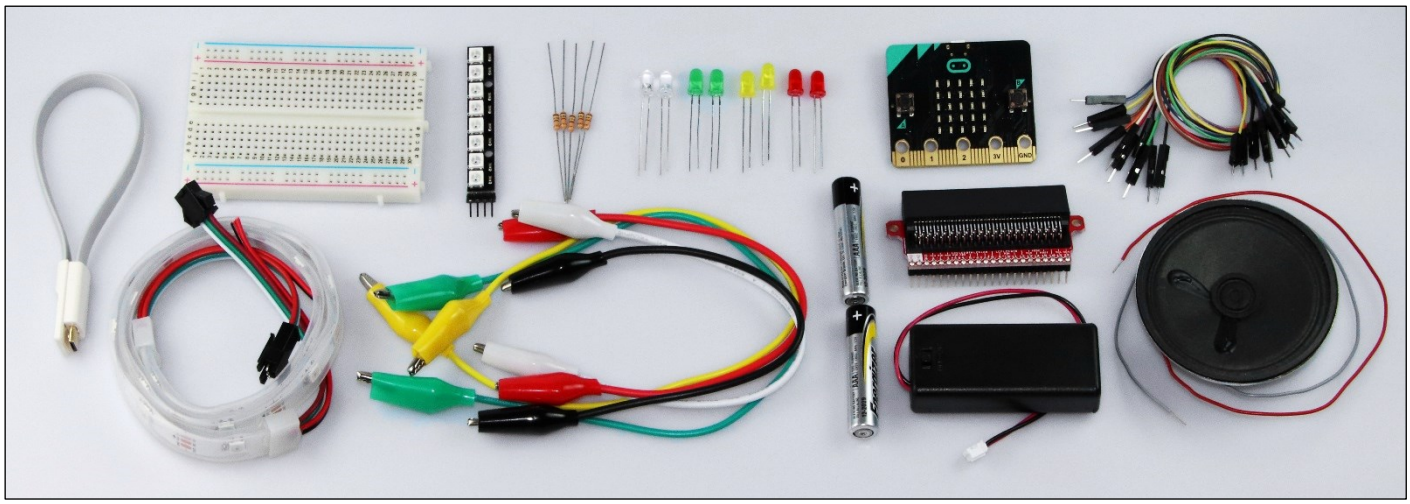
Auf dem micro:bit befindet sich das Programm „Snake“. Spiele es wie folgt:

- Drücke die Reset-Taste auf der Rückseite zum Starten des Spiels
- Rechter Button: drehe die Schlange nach rechts
- Linker Button: drehe die Schlange nach links

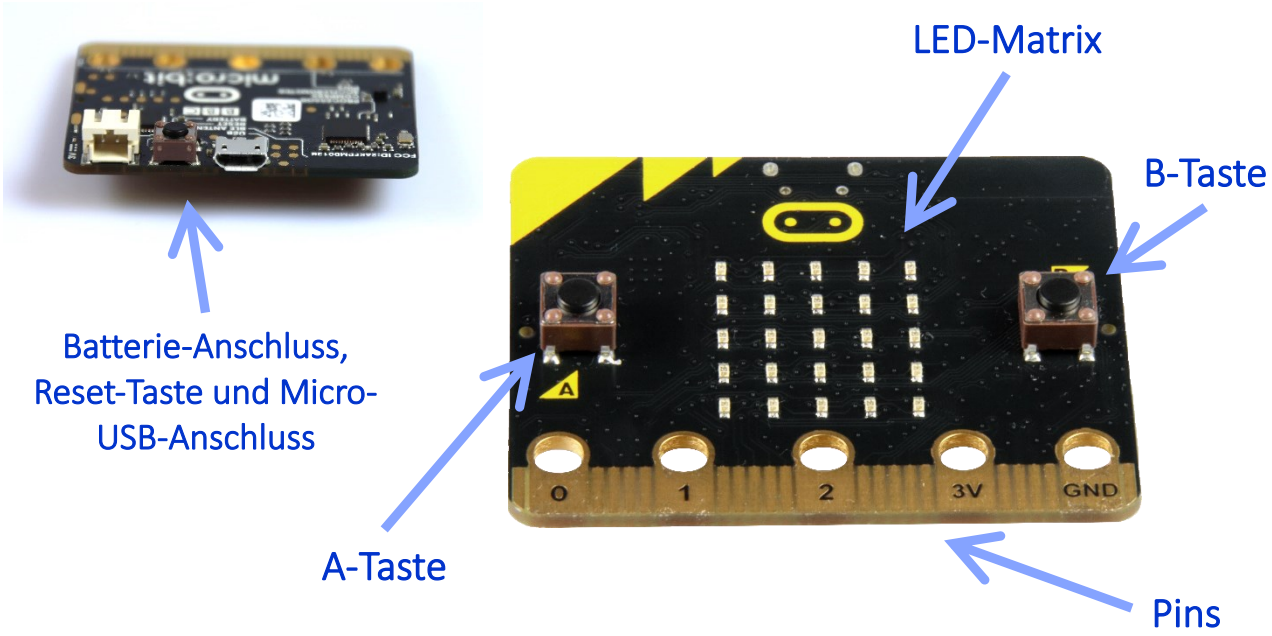
Die Schlange wächst, wenn sie Futter frisst. Versuche sie so lange wie möglich werden zu lassen, ohne dass sie sich selbst beißt! Unser Highscore ist 17.

Meine Bestleistung

Folgende Dinge findest du sonst noch in deinem Paket



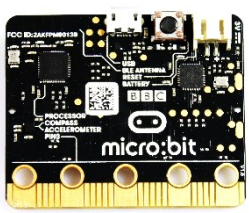
Wie funktioniert der micro:bit?



Der micro:bit ist ein sogenannter *Mikrocontroller*, ein einfacher kleiner Computer. Er kann *programmiert* werden, wie z.B. mit dem Spiel von gerade.

Wie jeder Computer arbeitet er nach dem **EVA-Prinzip:**

Sensoren



Mikrocontroller

Aktoren

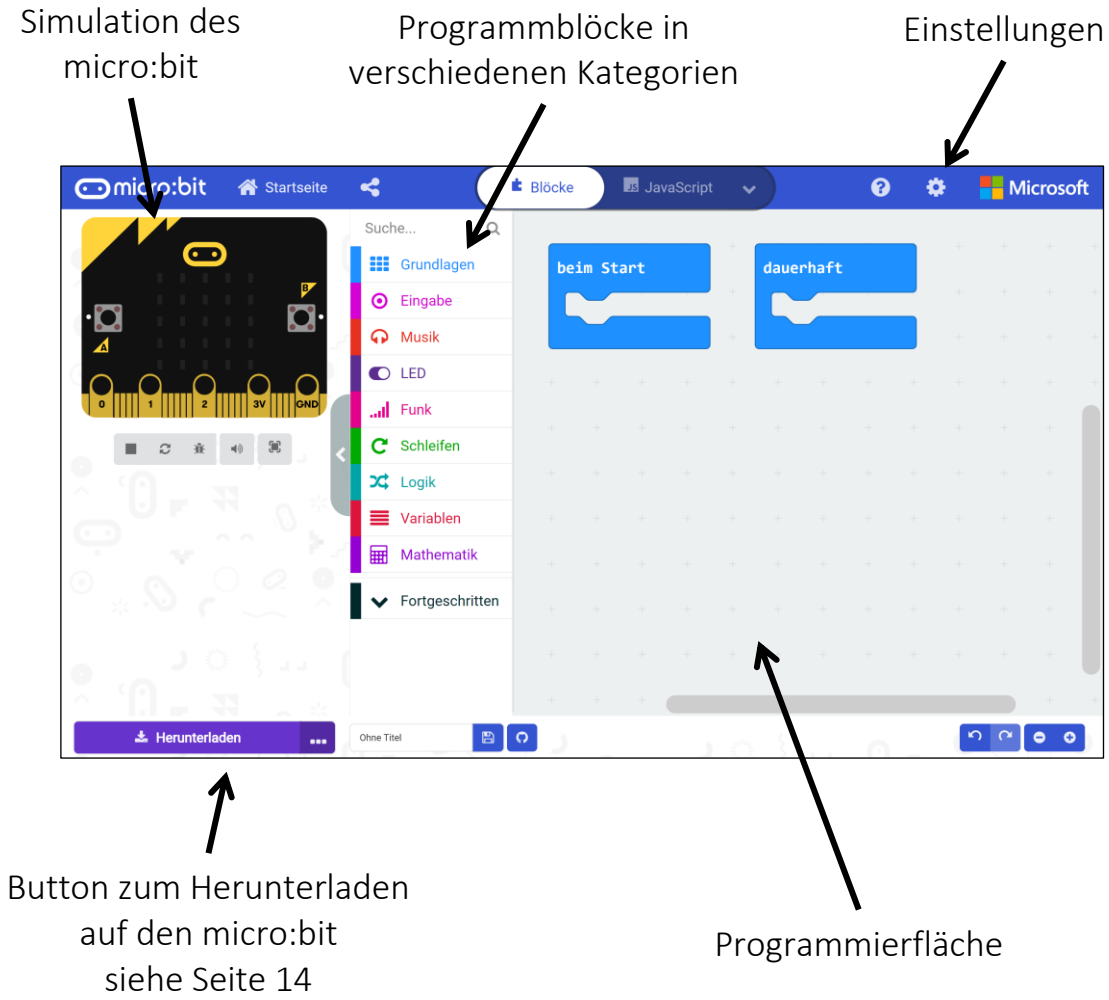
Die Programmierumgebung MakeCode

Öffne die Internetseite mit einem Webbrowser:

makecode.microbit.org

Klicke auf

Nachdem du deinem neuen Projekt einen Namen gegeben hast, siehst du die Programmierumgebung *MakeCode*:



Hallo micro:bit!

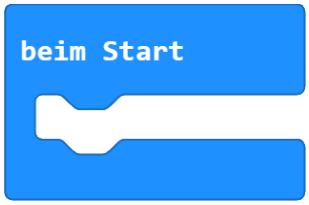
Wir wollen als erstes auf der Simulation in MakeCode einen Smiley anzeigen lassen. Der folgende Block aus der Kategorie „Grundlagen“ (blau) kann hierfür verwendet werden:

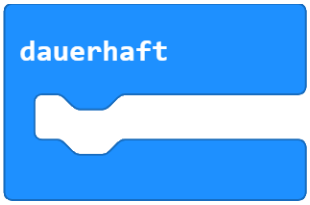
Bestimme durch Anklicken, welche LEDs der Matrix am micro:bit leuchten sollen oder nicht.

Experiment 1: Smiley

Lasse auf der Simulation in MakeCode einen Smiley mit dem obigen Block anzeigen.

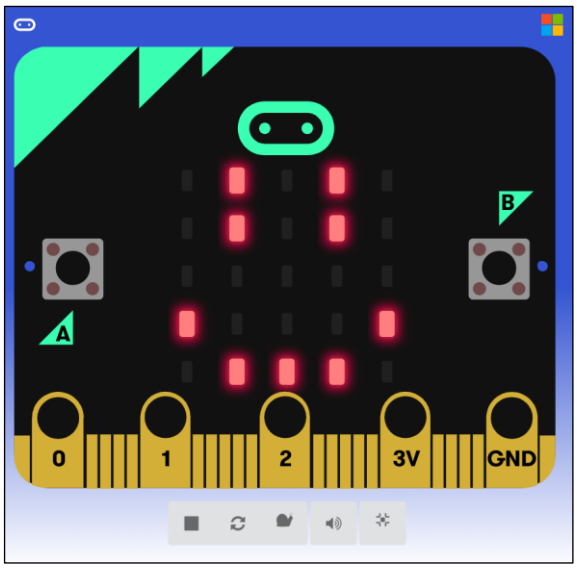
Jeder Block muss sich in einem Startblock befinden. Die beiden Standard-Blöcke, die bereits in der Programmierfläche zur Verfügung stehen, sind:



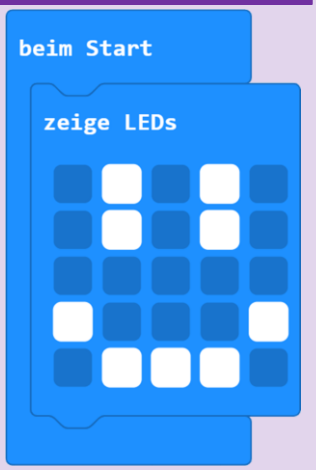


Fast jedes Programm eines Mikrocontrollers besteht aus diesen beiden Blöcken.

- „**beim Start**“: Blöcke hierin werden ausgeführt, sobald der Mikrocontroller startet, d.h. beim Anschließen des Stromes oder nach dem Drücken der Reset-Taste.
- „**dauerhaft**“: Blöcke hierin werden immer wieder ausgeführt. Das heißt, wenn der Inhalt „abgearbeitet“ ist, wird damit wieder von vorne begonnen.

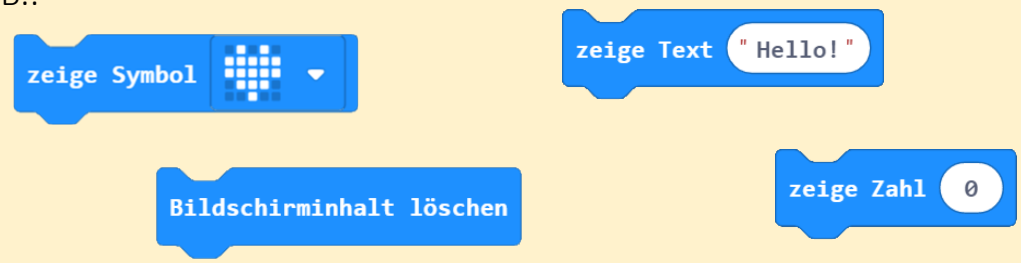


Lösung zu Experiment 1



Tipp

Es gibt noch mehr Blöcke, mit denen die LED-Matrix gesteuert werden kann, z.B.:



Wir verwenden hier den Block „**beim Start**“, weil die Matrix nur einmal verändert werden muss.

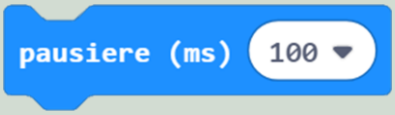
Soll sich darauf immer wieder etwas ändern, braucht man den Block „**dauerhaft**“. So wie im folgenden Experiment.

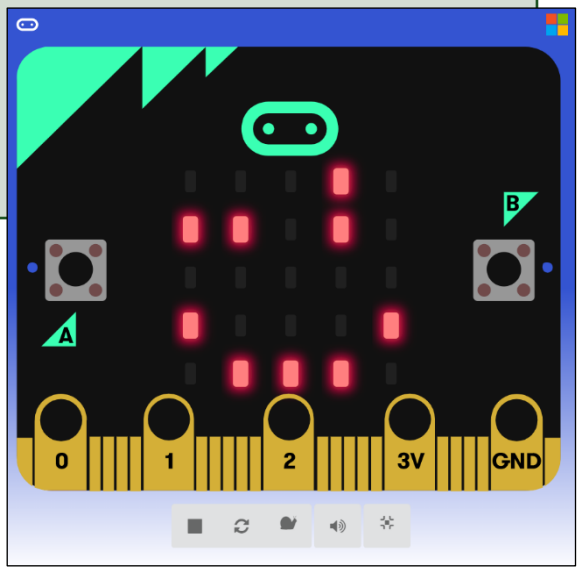
Experiment 2: Zwinkernder Smiley

Der Smiley auf der LED-Matrix soll nun dauerhaft zwinkern. Ziehe hierfür passende Blöcke aus der Kategorie „Grundlagen“ (blau) in die Programmierfläche.

- Es soll ein Smiley angezeigt werden.
- Dann warten wir kurz.
- Dann soll ein Smiley angezeigt werden, der zwinkert.
- Anschließend muss die „Zwinkerzeit“ gewartet werden.
- Danach soll das ganzer von vorne begonnen werden.

Beim Warten hilft der folgende Block:





Weiter gedacht…

Was würde passieren, wenn du die Blöcke für den zwinkernden Smiley nur in den Block „beim Start“ einfügst? Erkläre deine Beobachtung!

Programme für den realen micro:bit

Experiment 3: Schlagendes Herz

Auf der (realen!) LED-Matrix deines micro:bit soll nun ein schlagendes Herz angezeigt werden.

Verändere hierzu dein Programm des zwinkernden Smileys, um ein Herz kurz anzuzeigen und dann wieder auszublenden.

Wie du das Programm an deinen micro:bit überträgst, findest du auf der nächsten Seite.

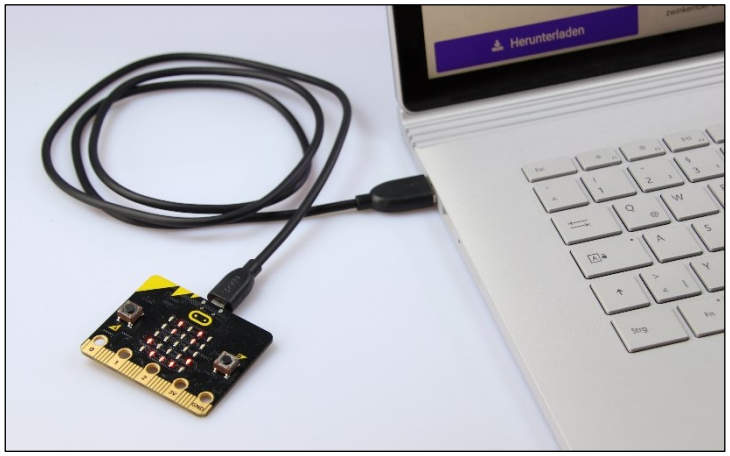
Programme an den micro:bit übertragen

Schritt 1:

Achtung!

Beachte auf Seite 4: „2. Stromversorgung“

Schließe den micro:bit mit dem USB-Kabel an deinen PC an.

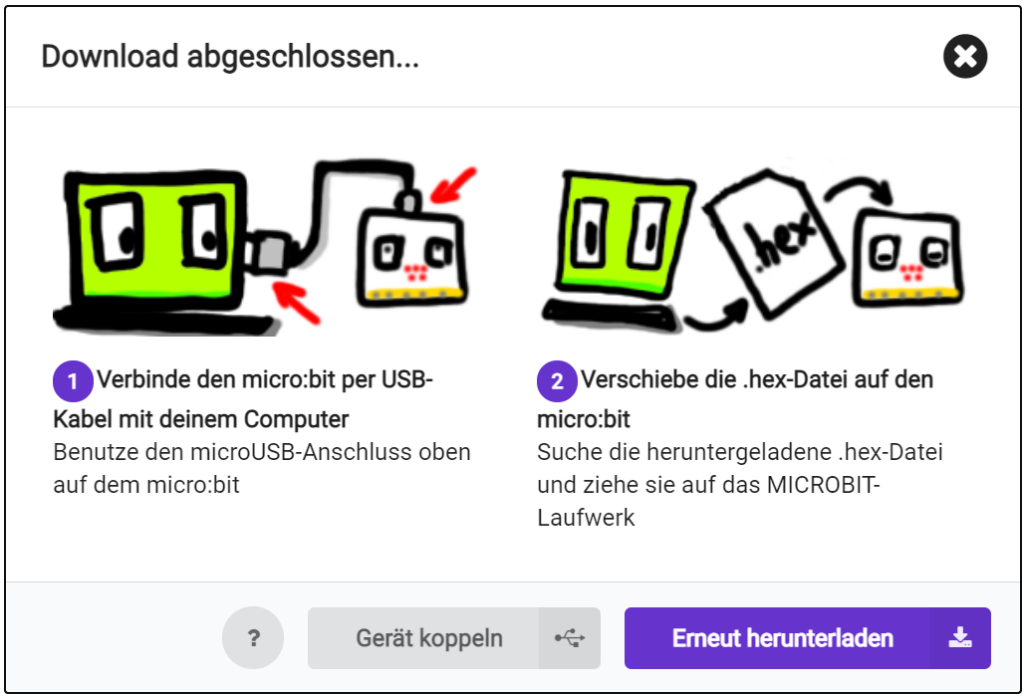


Schritt 2:

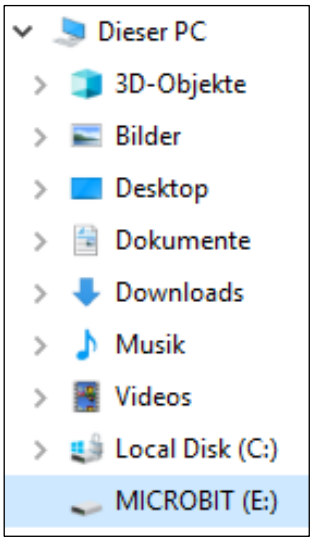
Klicke auf den Button 

.

Befolge die Anleitung zum Speichern der Programmdatei auf das micro:bit-Laufwerk:



Sollte sich ein weiteres Popup öffnen, wähle das Laufwerk des micro:bit aus und speichere die Datei dort.



Microsoft Windows

Mac OS

Hinweis

Manchmal wird die Datei auch direkt automatisch im „Download“-Ordner gespeichert. Verschiebe diese dann einfach auf das micro:bit-Laufwerk.

Falls du Google Chrome benutzt, kannst du das Gerät auch koppeln. Beachte hier die Anleitung auf Seite 38.

Lösung zu Experiment 3

Eine mögliche Lösung ist:

Weiter gedacht…

Fühle deinen Puls!

Versuche dein Programm so anzupassen, dass es deinem aktuellen Herzrhythmus entspricht.

Welchen Wert musst du verändern?

micro:bit und Krokodil-Klemmen – Lautsprecher

Jedes Mal, wenn das Herz „schlägt“ soll ein Ton erklingen. Hierfür brauchen wir den Lautsprecher aus dem Paket.

Um externe Aktoren und Sensoren anzuschließen, können die Pins an der Unterseite des micro:bit verwendet werden.

Information

Strom fließt nur in einem geschlossenen Stromkreis.

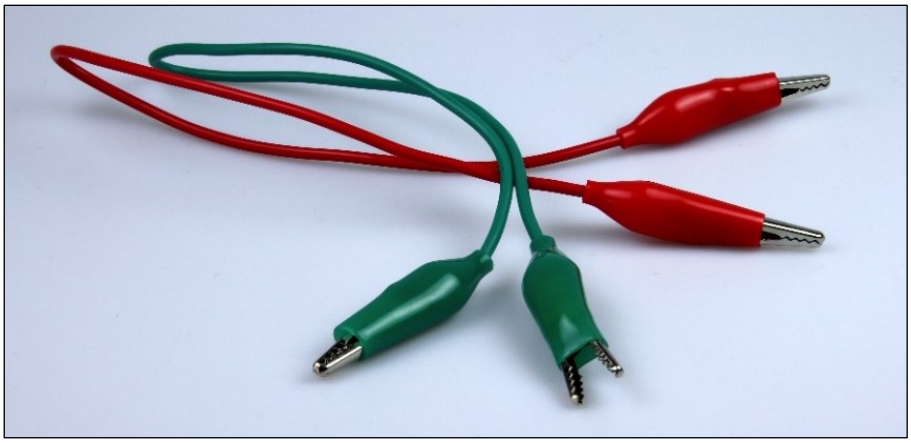
Die Pins 0, 1 und 2 sind programmierbar und geben Strom ab (Plus-Pol). Der Pin 3 V gibt dauerhaft Strom ab (Plus-Pol).

Der Ground (GND) ist der Minuspol.

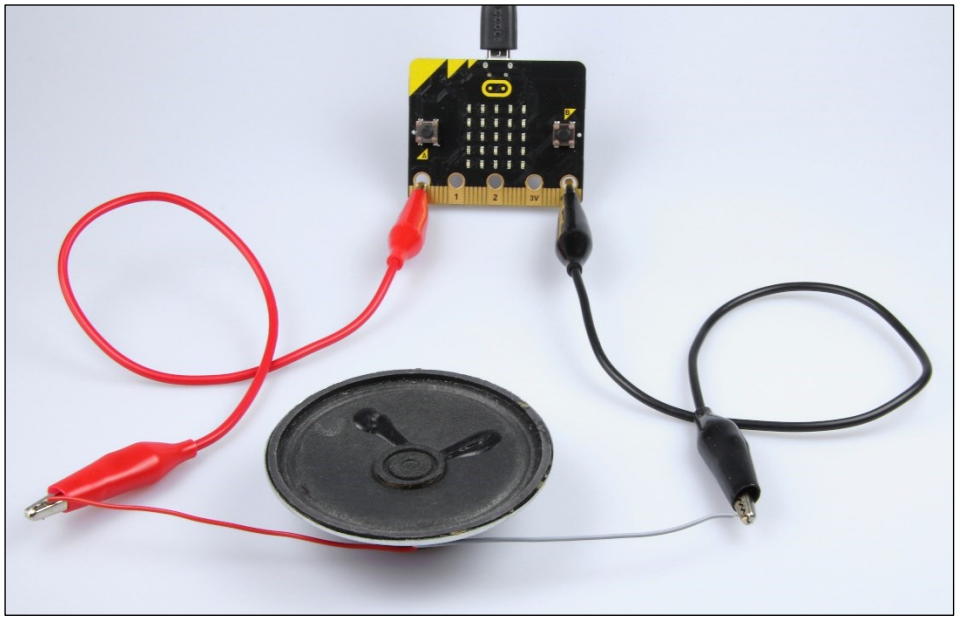
Achtung!

Beachte auf Seite 4:
„9. Vermeide Kurzschlüsse“

Um zum Beispiel den Lautsprecher anzuschließen, können *Krokodil-Klemmen* verwendet werden:

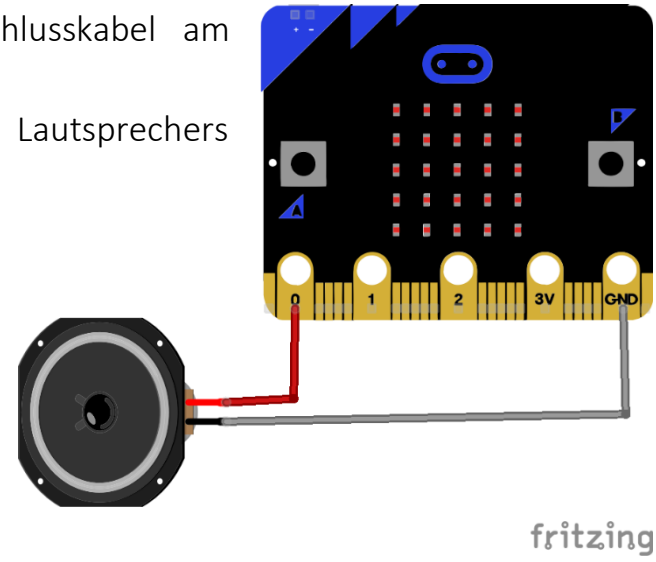


Die Farbe der Krokodil-Klemmen spielt eigentlich keine Rolle. Es ist aber üblich, an den *Ground schwarze* und an den *Plus-Pol* (Pin 0, 1, 2 oder 3 V) *rote Kabel* anzuschließen.



Schließe den Lautsprecher mithilfe von Krokodil-Klemmen an.

- Achte auf die Farbe der Anschlusskabel am Lautsprecher!
- Verbinde den grauen Draht des Lautsprechers GND, den roten mit Pin 0.



Achtung!

Beachte auf Seite 4:
„4. Schaltungen stromfrei aufbauen“

Experiment 4: Herzmonitor

Wir wollen nun mit dem Arduino einen kleinen Herzmonitor simulieren. Jedes Mal, wenn das Herz angezeigt wird, soll ein kurzer Ton abgespielt werden. Dies ist möglich mit dem Musik-Block:

Erweitere dein Programm und übertrage es auf deinen micro:bit!

Externe LED – eine Taschenlampe programmieren

Mithilfe von Krokodil-Klemmen lassen sich noch mehr Sensoren und Aktoren anschließen. Zum Beispiel eine weiße LED für eine Taschenlampe.

Information

Eine LED hat immer ein kürzeres und ein längeres Beinchen. Das kürzere Beinchen wird immer näher zum Ground ausgerichtet.

Biege die beiden Beinchen bei der Verwendung mit Krokodil-Klemmen leicht auseinander, um keinen Kurzschluss zu verursachen! (siehe „9. Vermeide Kurzschlüsse“ auf Seite 4)

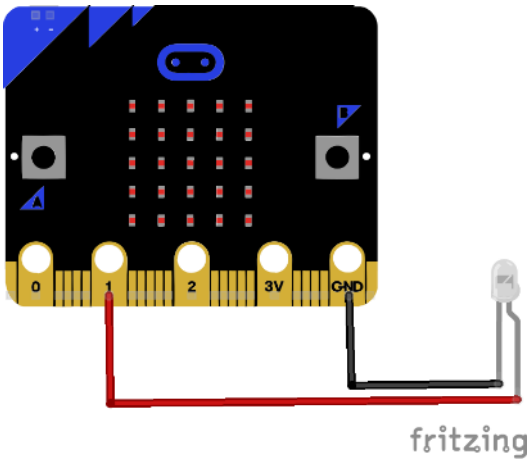
Eine LED kann über Pins gesteuert werden und benötigt ein *digitales* Signal (0 oder 1). Hierbei ist der Baustein

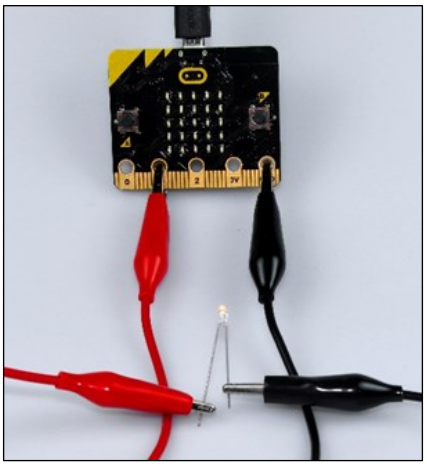
aus der Kategorie „Fortgeschritten“ → „Pins“ hilfreich.

Experiment 5a: Blinkende Taschenlampe

Schließe eine weiße LED mit Krokodil-Klemmen an Pin 1 an.

Lass deine Taschenlampe fürs erste regelmäßig blinken.





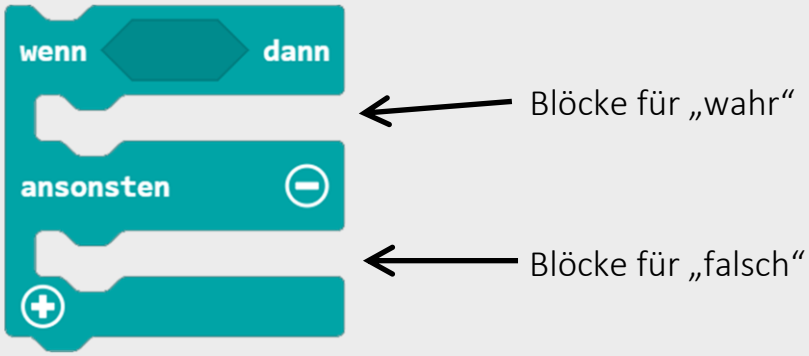
Wenn – Dann – Sonst

Die Taschenlampe soll nun um folgende Funktion erweitert werden:

> Wenn die Taste A gedrückt ist,
>
> dann wird die externe weiße LED angeschaltet,
>
> sonst wird die externe weiße LED ausgeschaltet.

Information

Falls Teile des Programmes von einer *Bedingung* abhängig sind, spricht man von einer *bedingten Anweisung*. Der passende Block aus „Logik“ ist:



Die Bedingung im Feld  kann *wahr (w)* oder *falsch (f)* sein. Je nachdem wird entweder der erste oder der zweite Teil ausgeführt und der Rest übersprungen.

Bedingungen haben in MakeCode immer die gleiche sechseckige Form. Die Bedingung „Knopf A ist gedrückt“ findet sich in der Kategorie „Eingabe“:

Experiment 5b: Taschenlampe mit Taster

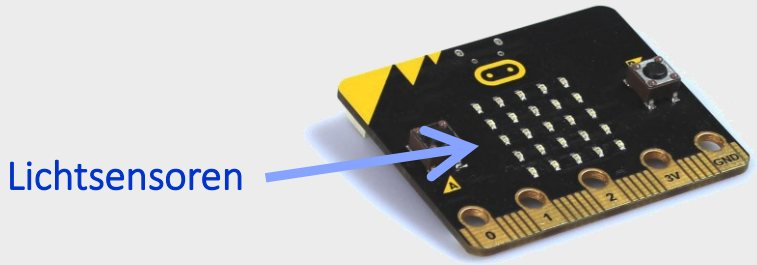
Programmiere eine Taschenlampe: Die weiße LED soll, wenn A gedrückt ist, eingeschaltet sonst ausgeschaltet werden.

Automatisches Nachtlicht

Wir wollen nun ein automatisches Nachtlicht bauen, das immer dann angeht, wenn es dunkel wird.

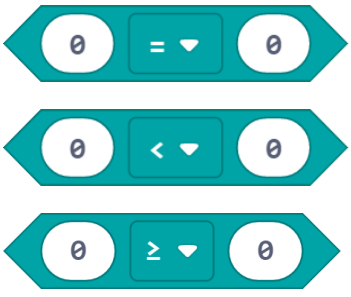
Information

Das Umgebungslicht wird vom micro:bit mit **Lichtsensoren** auf der Vorderseite gemessen. Sie ist in `Lichtstärke` als Zahl zwischen 0 und 255 gespeichert.



Je kleiner die Zahl ist, desto dunkler ist die Umgebung.

Vergleiche liefern Wahrheitswerte. Die Lichtstärke kann mit einer Zahl mithilfe der folgenden Blöcke verglichen werden:



usw.

Experiment 6a: Automatisches Nachtlicht

Programmiere ein Nachtlicht mit der angeschlossenen externen LED mithilfe der Blöcke oben. Ziehe die Blöcke dafür sinnvoll „ineinander“.

Weiter gedacht…

Experimentiere mit dem Vergleichs-Zahl für den Lichtwert.

Ab welchem Wert sollte deiner Meinung nach das Licht der externen LED angeschaltet werden?

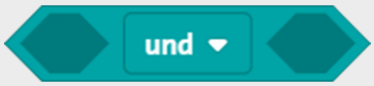
Simuliere eine dunkle Umgebung, indem du mit deiner Hand einen Schatten auf dem micro:bit produzierst.

Leider besitzt unser Nachtlicht noch einen entscheidenden Nachteil: Wenn der micro:bit versehentlich falsch herum liegt, geht das Licht an, obwohl es eigentlich noch nicht dunkel genug ist. Wir wollen das mit dem **Lagesensor** des micro:bit beheben. Er liefert uns einige Eigenschaften des Mikrocontrollers.

Der folgende Bedingungs-Block benutzt den Lagesensor, um die Position des micro:bit zu ermitteln:

Information

Mehrere Bedingungen können mit dem folgenden Block aus der Kategorie „Logik“ verknüpft werden:

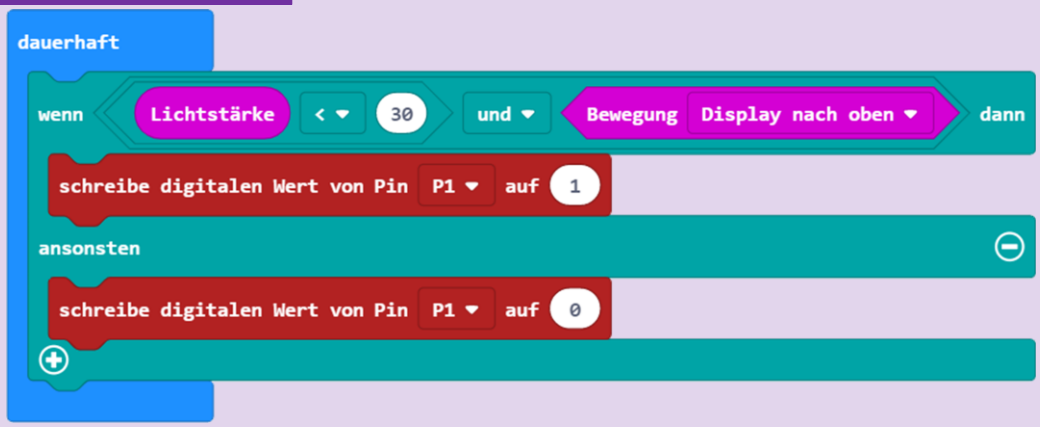


Er liefert der Wert *wahr*, wenn beide Bedingungen wahr sind, sonst *falsch*.

Experiment 6b: Automatisches Nachtlicht (verbessert)

Verbessere dein Nachtlicht so, dass es nur dann angeht, wenn das Umgebungslicht unter einem Schwellwert liegt **und** das Display nach oben zeigt.

Lösung zu Experiment 6b



Experiment 6c: Automatisches Nachtlicht (Variation)

Benutze den Baustein 

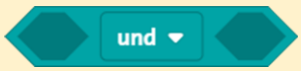
, um dein Programm so abzuändern, dass das Licht angeht, wenn es dunkel ist und das Display **nicht** nach oben zeigt.

Der logische Block 

liefert *wahr*, wenn **eine** Bedingung wahr ist, sonst *falsch*.

Weiter gedacht…

Experimentiere mit den Bausteinen „und“, „oder“ und „nicht“. Fülle die folgenden Tabellen aus. Die horizontale und vertikale Richtung stellt jeweils eine Eingabe dar. Z. B.: *Welchen Wert hat der gesamte Block, wenn die linke und die rechte Bedingung bei „und“ wahr sind?*



	w	f
w		
f		

	w	f
w		
f		

w	
f	

Eigene Taster

Anstelle der fest verbauten Taster, wollen wir uns nun selbst welche basteln. Du benötigst hierfür:

- Ein Paar Krokodil-Klemmen
- Alufolie
- Den micro:bit

Falte zwei etwa Hand-große Felder aus drei oder vier Lagen Alufolie. Verbinde das eine Feld durch eine Krokodil-Klemme mit dem Ground, das andere mit Pin 0.

Zur Erinnerung

Strom fließt nur in einem geschlossenen Stromkreis. Ein „Verbraucher“ muss sowohl an den Minus- als auch an den Pluspol angeschlossen sein.

In diesem Experiment wird durch deine Berührung der beiden Taster der Stromkreis zwischen Ground und Pin 0 geschlossen. **Keine Angst, der Strom ist so gering, dass man ihn überhaupt nicht spürt!**

Achtung!

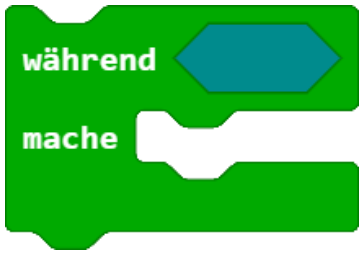
Beachte auf Seite 4:
„10. Berühren der Kontaktflächen“

Experiment 7: Eigene Taster

Schreibe ein Programm, das ein beliebiges Symbol auf der LED-Matrix anzeigt, wenn der Stromkreis geschlossen ist. Folgende Bedingung hilft dir dabei:

Wiederholung, Wiederholung, Wiederholung

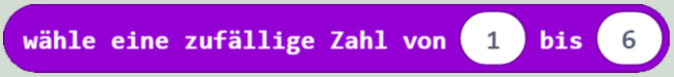
Eine Form der Wiederholung hast du bereits kennen gelernt: die *unendliche Wiederholung* im Block „dauerhaft“. Eine weitere Form ist die *bedingte Wiederholung* aus der Kategorie „Schleifen“:



Auch hier kann im Feld  eine Bedingung angegeben werden. Das Innere des Blockes wird solange ausgeführt, bis die Bedingung nicht mehr zutrifft.

Experiment 8: Ewiger Würfel

Formuliere ein Programm, das eine zufällige Zahl zwischen 1 und 6 auf der LED-Matrix anzeigt, solange die selbst gebauten Taster gedrückt sind. Eine Zufallszahl erhältst du mit:



Weiter gedacht…

In manchen Fällen muss gewartet werden, bis eine Bedingung nicht mehr zutrifft. Man benutzt dann den Block der bedingten Wiederholung mit leerem Inhalt. Gesprochen:

Warte bis die Bedingung **nicht mehr** zutrifft.

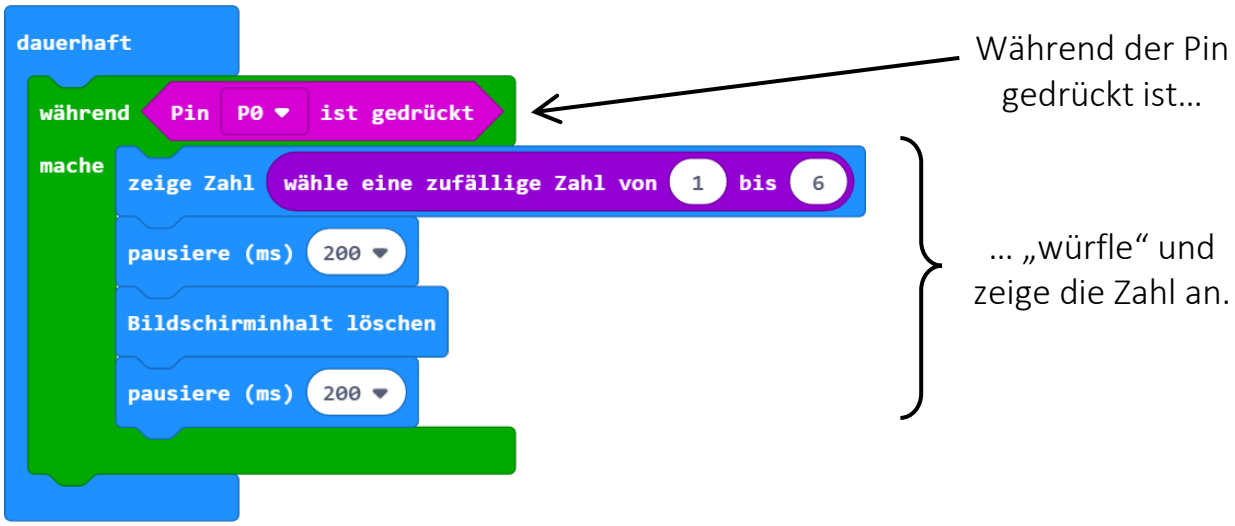
oder

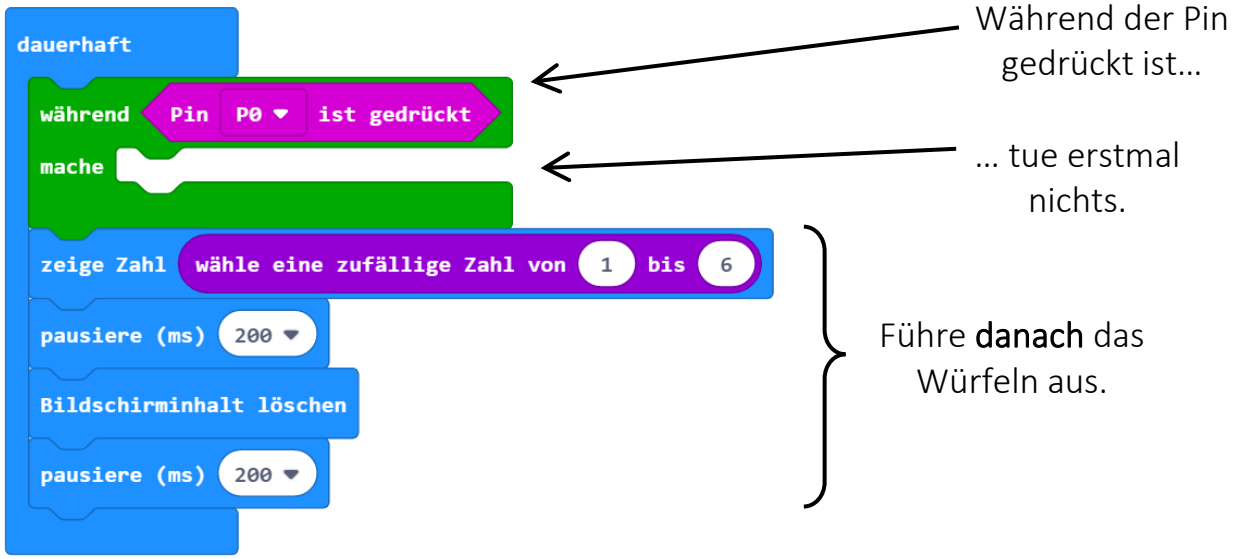
Während die Bedingung zutrifft,
mache nichts.

Ändere dein Experiment so ab, dass immer dann eine neue Zufallszahl angezeigt wird, wenn die Taster losgelassen wurden.

Lösung zu Experiment 8 und seiner Variante

Wir vergleichen die beiden Lösungen der vorherigen Seite. Sie unterscheiden sich eigentlich nur in der Position des „während“-Blocks:





Wie schnell reagierst du?

Wie schnell ist deine Reaktionszeit? Wer reagiert schneller? Bastle dir ein kleines Reaktionsspiel mit dem micro:bit!

Schließe hierfür deinen selbst gebastelten Taster an Pin 1 an. Das Spiel soll vorerst nur einen Benutzer unterstützen. Wir werden es später erweitern. Es funktioniert wie folgt:

Reaktionsspiel für einen Benutzer

- Der Spieler legt eine Hand auf die Ground-Fläche.
- Irgendwann (zufällig) leuchtet auf den LEDs des micro:bit etwas auf.
- Jetzt muss der Spieler so schnell es geht die Pin 1-Fläche berühren.

Der micro:bit tut dabei immer wieder folgendes:

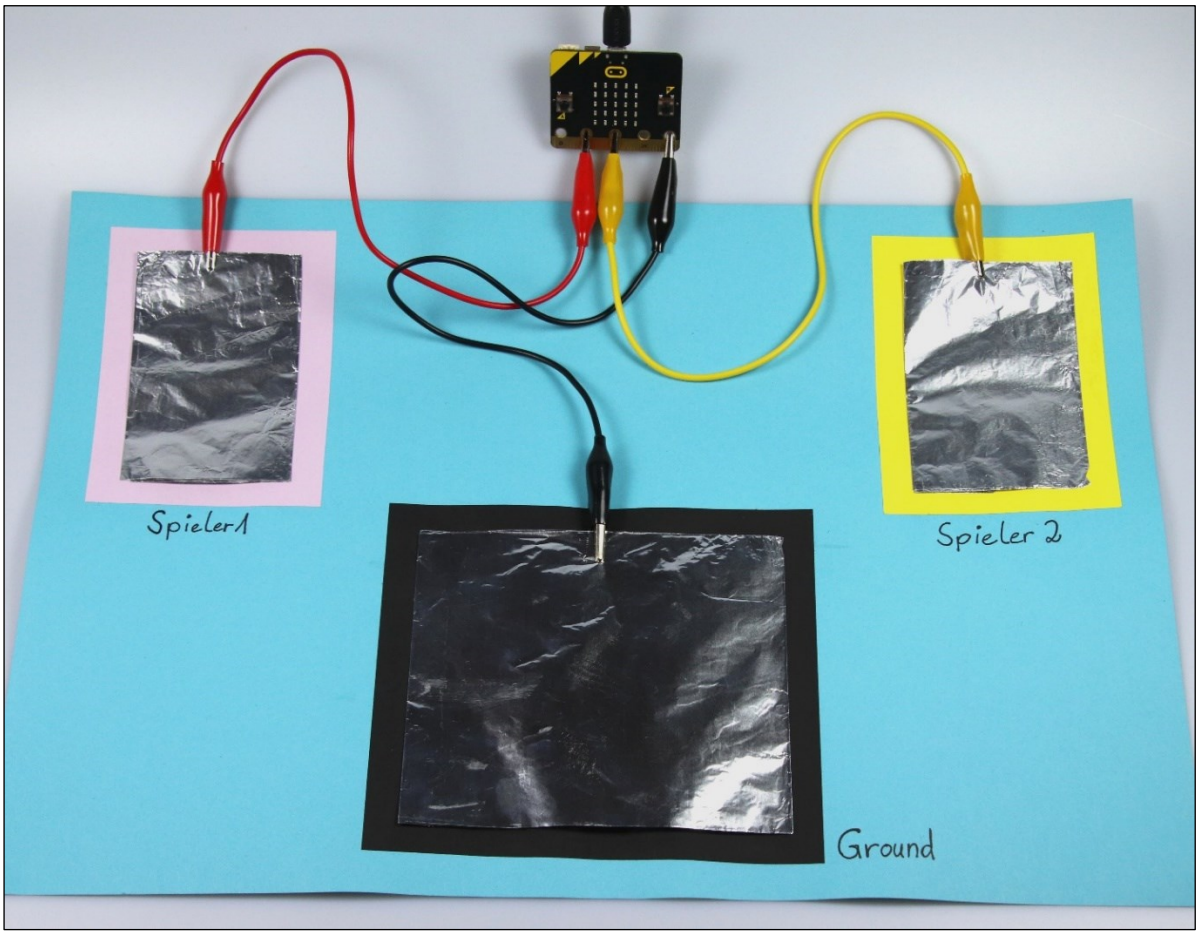
- Pausiere zufällig zwischen 3 und 6 Sekunden
- Zeige ein Symbol auf der LED-Matrix an
- Warte bis Pin 1 gedrückt wird
- In dieser ersten Variante des Spiels kann nur Spieler 1 gewinnen, deshalb zeige die Zahl 1 auf der Matrix an
- Warte 2 Sekunden, leere die Matrix und starte von vorne.

Experiment 9a: Reaktionsspiel für einen Benutzer

Programmiere das Reaktionsspiel für einen Benutzer. Tipps:

- Eine Sekunde hat 1000 Millisekunden (ms)
- „Warte bis Bedingung“ ≈ „Tue solange Bedingung nicht erfüllt nichts“

Erweitere dein Spiel nun um einen zweiten Benutzer. Bastle dazu eine weitere Fläche, die du mit Pin 2 verbindest. Die Ground-Fläche können beide Spieler gemeinsam benutzen. Du kannst die Flächen auch auf einen Stück Papier festkleben.



Experiment 9b: Reaktionsspiel für zwei Benutzer

Passe dein Programm von oben an!

- Es muss jetzt gewartet werden, bis Pin 1 oder Pin 2 gedrückt werden.
- Danach muss entschieden werden: Wenn Pin 1 gedrückt wurde, wird eine 1 angezeigt, wenn Pin 2 gedrückt wurde, eine 2.

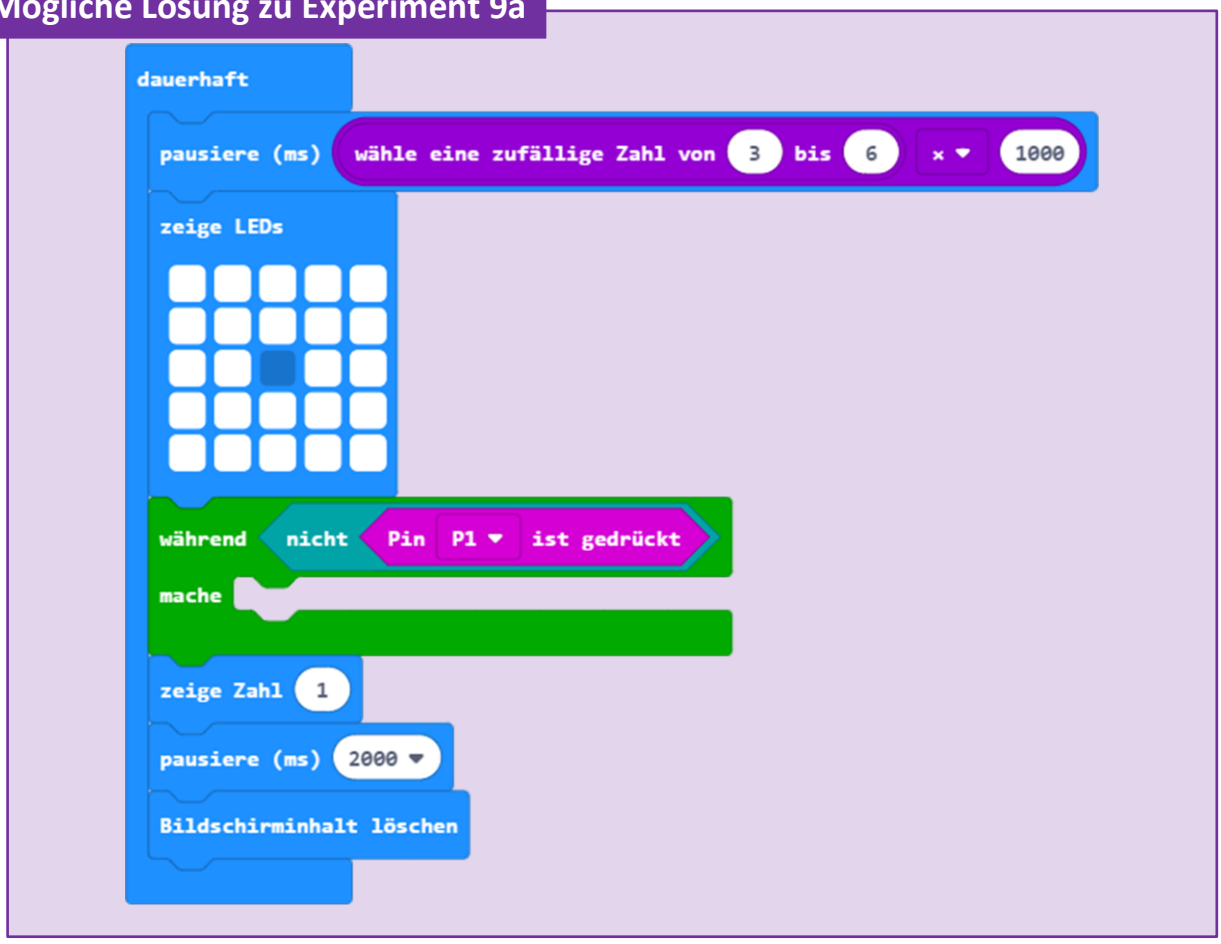
Weiter gedacht…

Wer aus deiner Familie hat die beste Reaktionszeit?

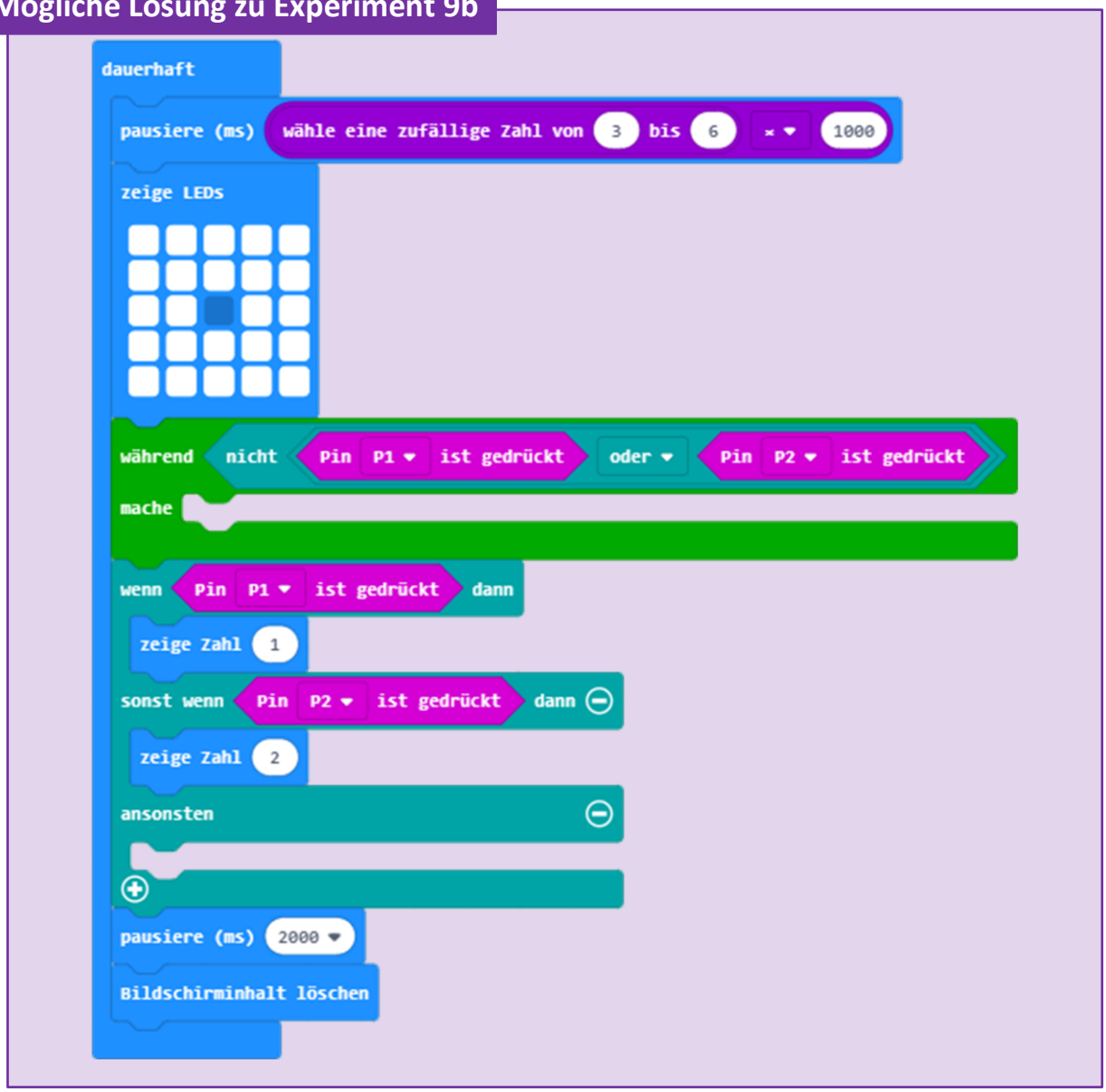
Achtung!

Beachte auf Seite 4:
„10. Berühren der Kontaktflächen“

Mögliche Lösung zu Experiment 9a



Mögliche Lösung zu Experiment 9b



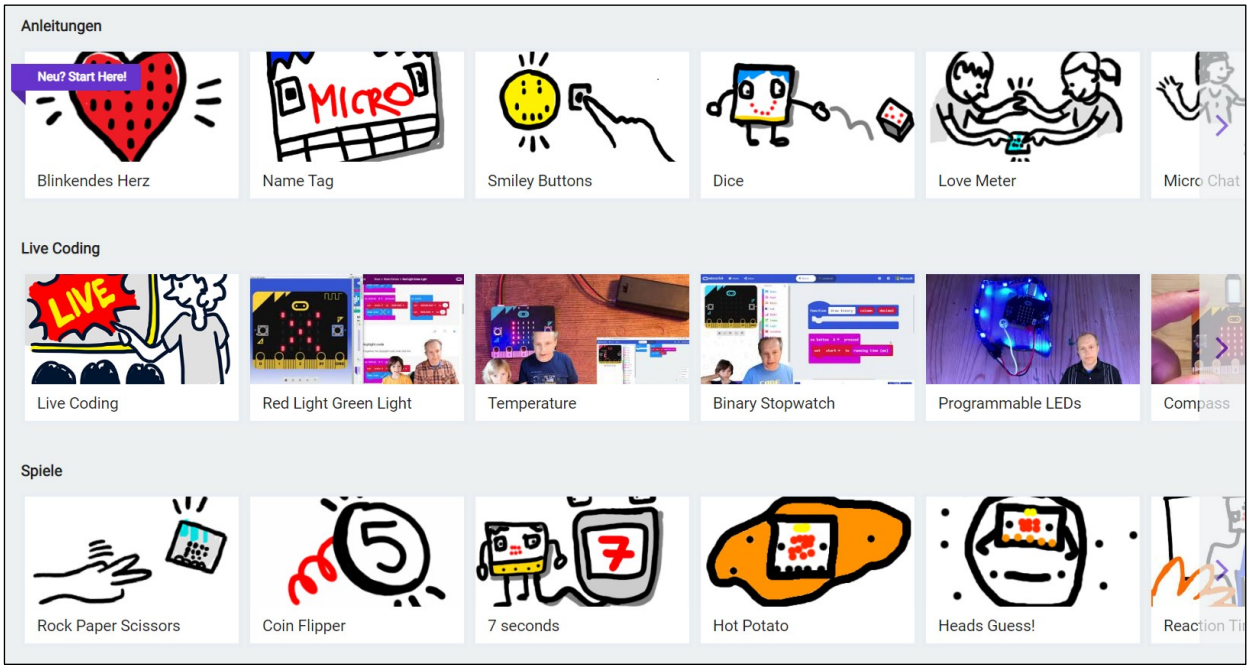
Herzlichen Glückwunsch!

Du hast nun die grundlegende Funktionsweise eines Microcontrollers kennen gelernt und bist bereit, ein paar komplexere Beispiele umzusetzen. Schau dir dazu einfach die nächsten Seiten an.

Weitere Beispiele findest du auf der Seite

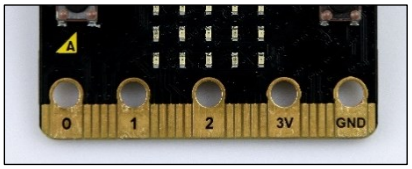
makecode.microbit.org

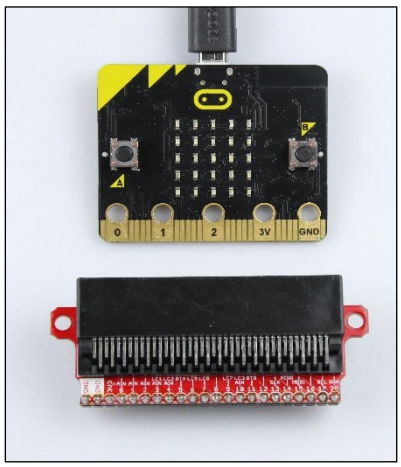
wenn du etwas weiter nach unten scrollst. Dort sind auch Anleitungen und andere Anregungen für Programme zu finden.

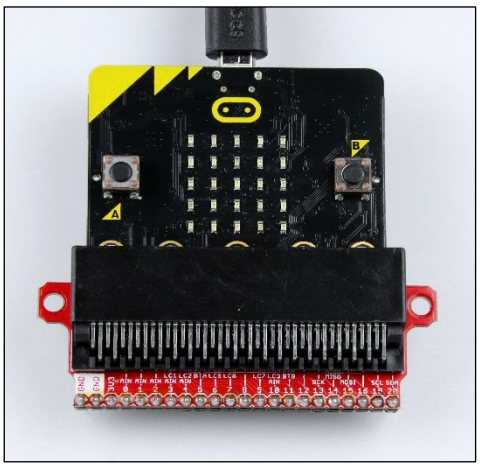


Anhang: Die Verwendung des Breadboard

In vielen Fällen reichen die drei programmierbaren Pins des micro:bit nicht aus. Aus diesem Grund besitzt er noch weitere schmale Pins, die mithilfe des „Breakout“-Steckers benutzt werden können:



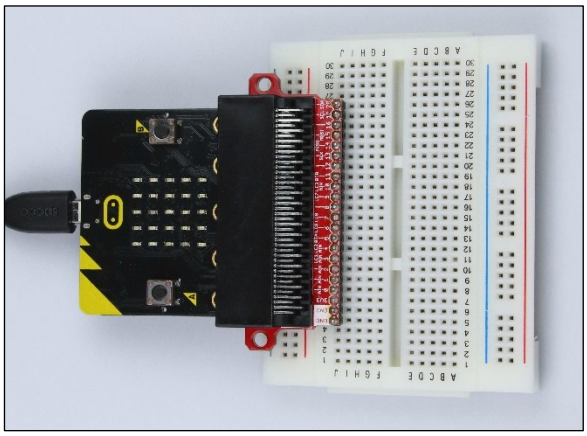




Mithilfe dieses Erweiterung-Steckers kann der Microbit auf ein Steckbrett, ein sogenanntes *Breadboard*, gesteckt werden:

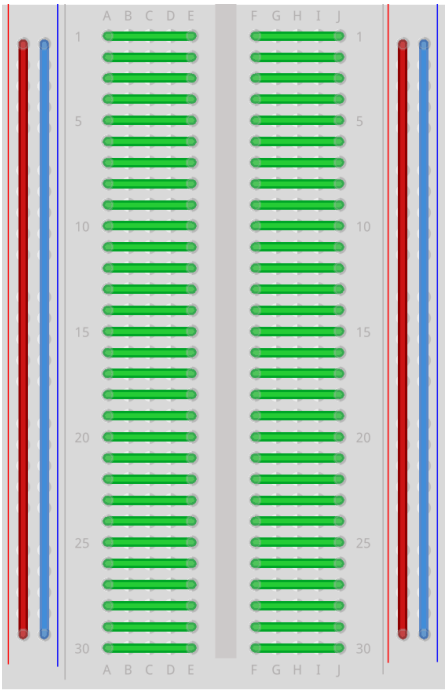
Achtung!

Beachte auf Seite 4: „4. Schaltungen stromfrei aufbauen“



In das Breadboard können elektronische Bauteile und Kabel einfach eingesteckt werden. Die Löcher auf dem Steckbrett sind untereinander so verbunden:

- Links und rechts außen sind die „Spalten“ durchverbunden (im Bild rechts rot und blau).
- Im Inneren Bereich sind jeweils die „Zeilen“ links bzw. rechts untereinander verbunden (im Bild grün).

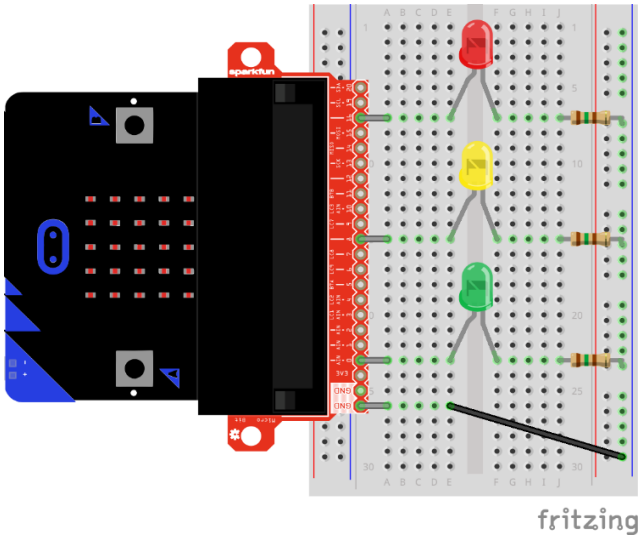


Ampel-LEDs auf dem Steckbrett

Stecke eine rote, gelbe und grüne LED wie abgebildet auf dein Breadboard. Denke wieder daran, dass das kurze Beinchen einer LED in Richtung des Minus-Pols (Ground) zeigen muss.

Der Widerstand (150 Ohm) ist notwendig, damit die LEDs mit der korrekten Spannung versorgt werden. Baue ihn entsprechend in deinen Aufbau ein.

Um Verbindungen auf dem Steckbrett herzustellen, verwende die dünnen sogenannten *Jumper-Kabel*.



Die Pins sind auf dem Breakout-Stecker durchnummeriert und lassen sich mit dem folgenden Baustein steuern:

Experiment 10 (Zusatz): Ampel

Programmiere den Ablauf einer Ampelschaltung.

Information

Eine normale Ampel schaltet rot – rotgelb – grün – gelb – rot.

Weiter gedacht…

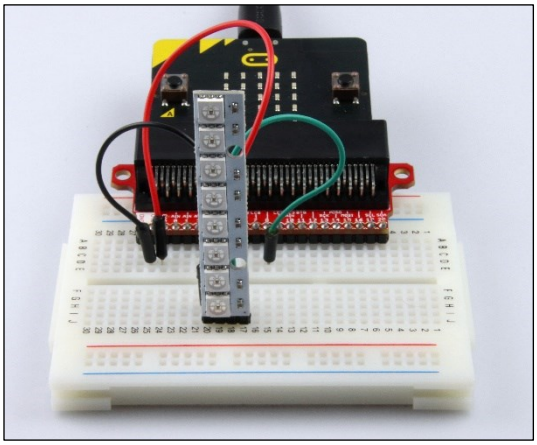
Verwende einen Taster des micro:bit als Fußgänger-Taster. Wenn er gedrückt wurde, soll die Ampel für eine gewisse Zeit auf rot umschalten.

Man könnte auch ein akustisches Signal für blinde Menschen einbauen.

Welche Anwendungen fallen dir noch mit den LEDs ein?

RGB-LED-Stick

Betrachten wir einen weiteren externen Aktor: Einen RGB-LED-Stick.

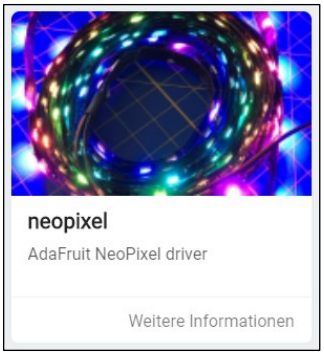


Information

Bei einem RGB-Stick kann jede einzelne LED eine beliebige Farbe annehmen. Sie wird aus den drei Grundfarben Rot, Grün und Blau gemischt.

Um LED-Sticks in MakeCode programmieren zu können, musst du das Erweiterungspaket *Neopixel* installieren.

Klicke hierzu ganz unten in den Block-Kategorien auf „Fortgeschritten“ → „Erweiterungen“. Wähle hier Neopixel aus.

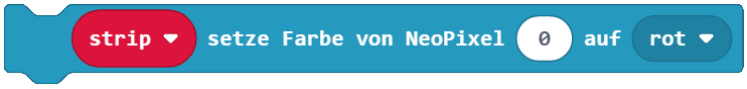


Anschließend steht dir eine neue Block-Kategorie „Neopixel“ zur Verfügung.

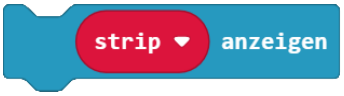
Um den LED-Stick zu verwenden, platziere die folgenden Blöcke in „beim Start“:

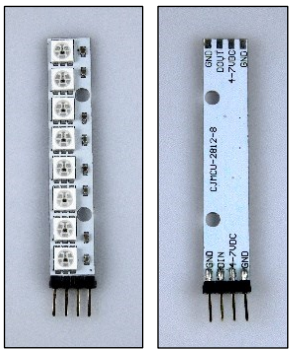
Hier wird die Länge des Strips (bei uns 8) und der Pin festgelegt.

Anschließend können die einzelnen Pixel mithilfe von RGB-Mischungen beliebig eingefärbt werden. Kombiniere dafür unter „mehr“ die Blöcke:



Um die Änderungen wirksam zu machen, ist der Aufruf des folgenden Blockes erforderlich:

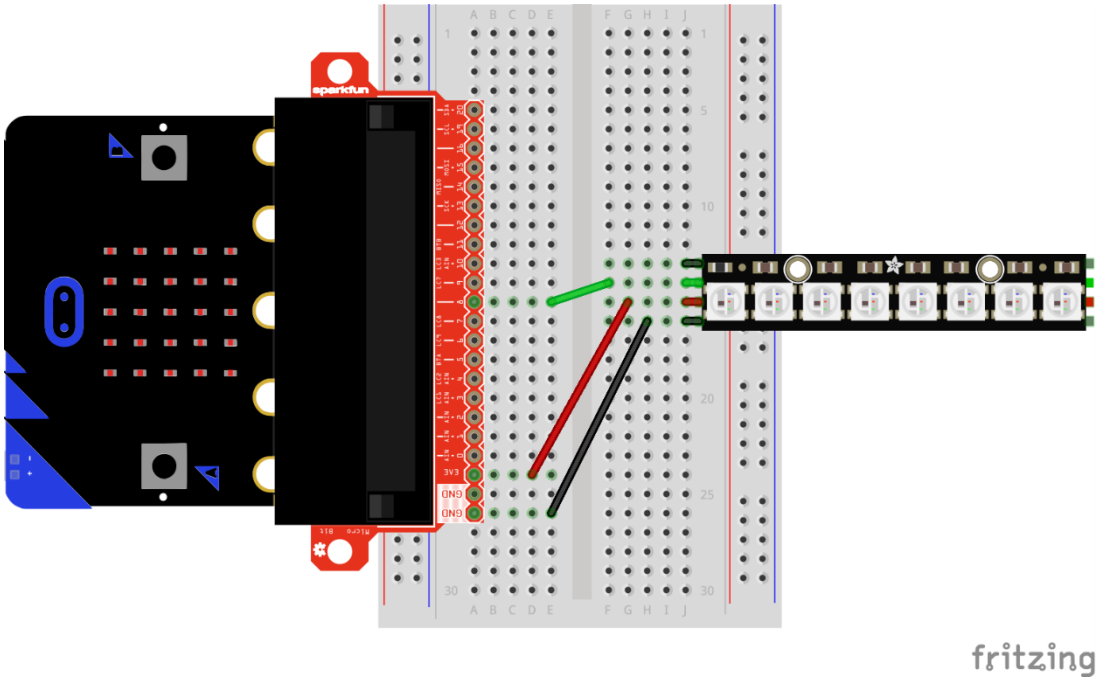




Der LED-Stick hat vier Anschlussmöglichkeiten, die auf der Rückseite benannt sind: 2x **GND**, **VDC** und **DIN**.

Verbinde einen der beiden Grounds mit dem Ground (Minuspol) und VDC mit dem 3V3 (Pluspol). Bei DIN wird eine digitale Eingabe erwartet. Diesen Anschluss musst du mit dem Pin verbinden, der im Startblock benannt wurde (z. B. Pin 8).

Möglicher Schaltplan:



Experiment 11 (Zusatz): LED-Stick

a) Färbe alle 8 LEDs in derselben Farbe.

b) Lass einen andersfarbigen Punkt von unten nach oben wandern.

c) Baue erneut eine Ampel, die die obersten drei Pixel für rot, gelb und grün verwendet.

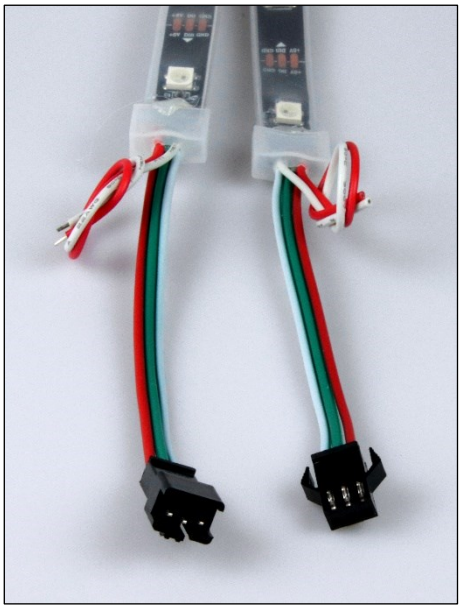
d) Überlege dir weitere Anwendungen für den LED-Stick und probiere sie aus.

Information

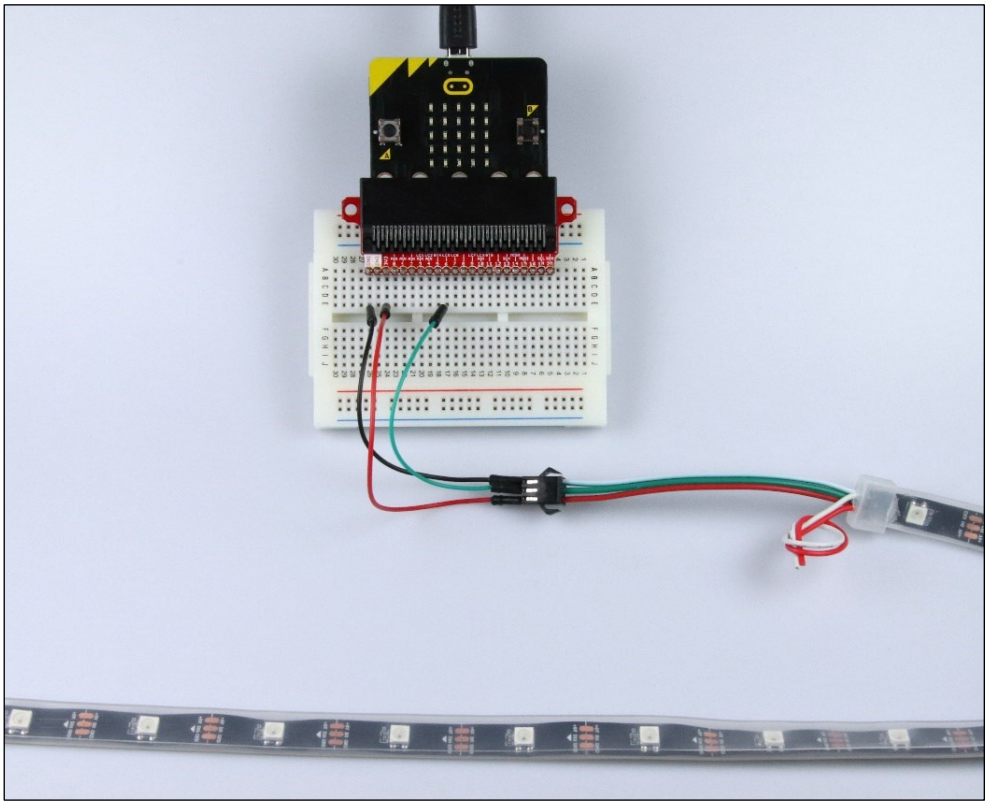
Wie fast immer in der Informatik werden die LEDs beginnend bei 0 durchnummeriert. Die unterste LED ist also die 0te, darüber die 1te usw.

LED-Schlauch

Auf die gleiche Art und Weise kann man einen LED-Schlauch mit 15 LEDs anschließen und programmieren. Er hat zwei verschiedene Anschlussseiten:



Benutze zum Anschließen Jumper-Kabel, die du in die rechte Anschlussseite im Bild oben steckst. Hier wird der Ground ausnahmsweise durch ein weißes Kabel dargestellt. Die beiden zusätzlichen Kabel werden nicht benötigt.

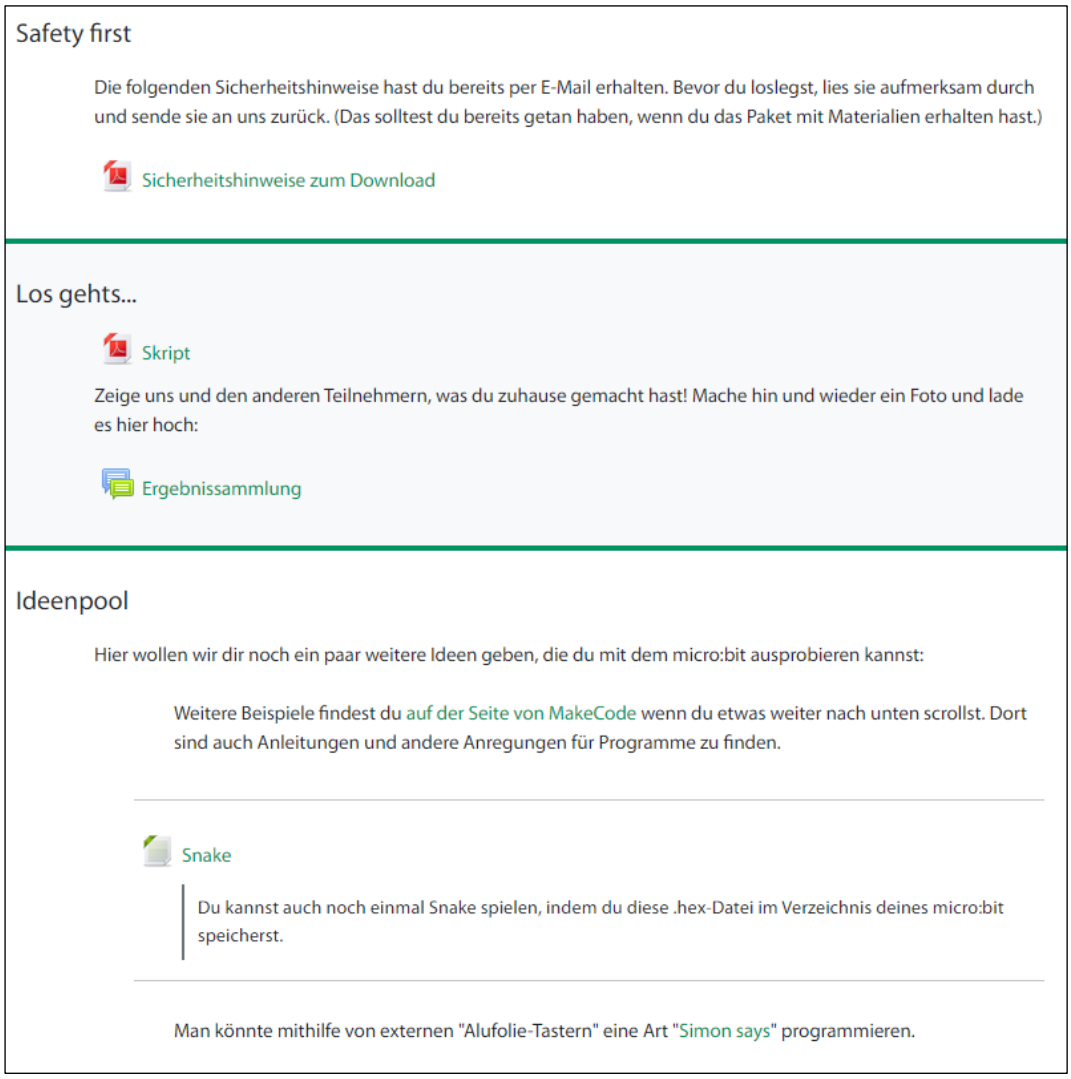


Experiment 12 (Zusatz): LED-Schlauch

Überlege dir Anwendungen für den LED-Schlauch und setze sie um.

Poste davon Fotos im E-Learning-Kurs (siehe Seite 5).

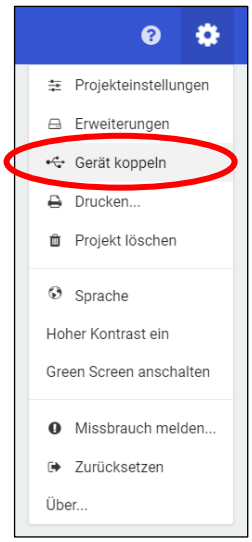
Links und Anweisungen zu weiteren Ideen findest du auch im E-Learning-Kurs. Schau einfach mal vorbei!

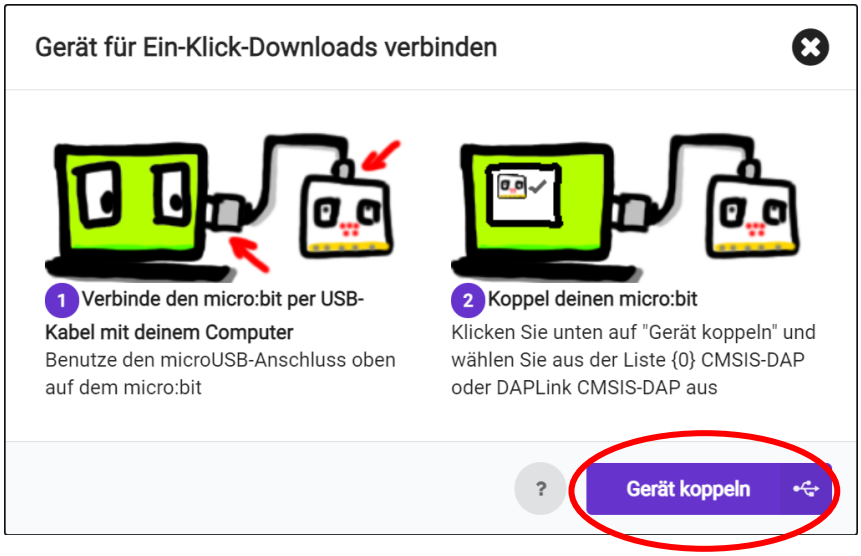


Anhang: micro:bit mit Google Chrome koppeln

Schritt 1:

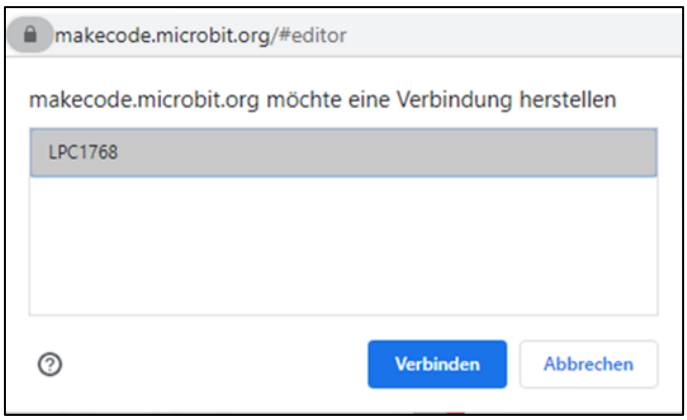
Wähle unter den **Einstellungen** die Option „**Gerät koppeln**“. Es öffnet sich ein Fenster. Klicke dort erneut auf „**Gerät koppeln**“:





Schritt 2:

Wähle deinen micro:bit aus der Liste aus (es sollte nur einer verfügbar sein) und klicke auf „Verbinden“:



Dein micro:bit ist nun mit deinem Browser gekoppelt. Um ein Programm auf den Mikrocontroller zu übertragen, klicke auf Herunterladen. Das Programm wird dann nach kurzer Zeit starten.

Gefördert durch BayernMINT
kompetent. vernetzt. erfolgreich

Dieses Skript entstand im Rahmen des
TAO-Schülerforschungszentrum Oberfranken
TechnologieAllianzOberfranken
tao-oberfranken.de/sfz

Digitales Lehren und Lernen &
Didaktik der Informatik
Universität Bayreuth
dlldi.uni-bayreuth.de