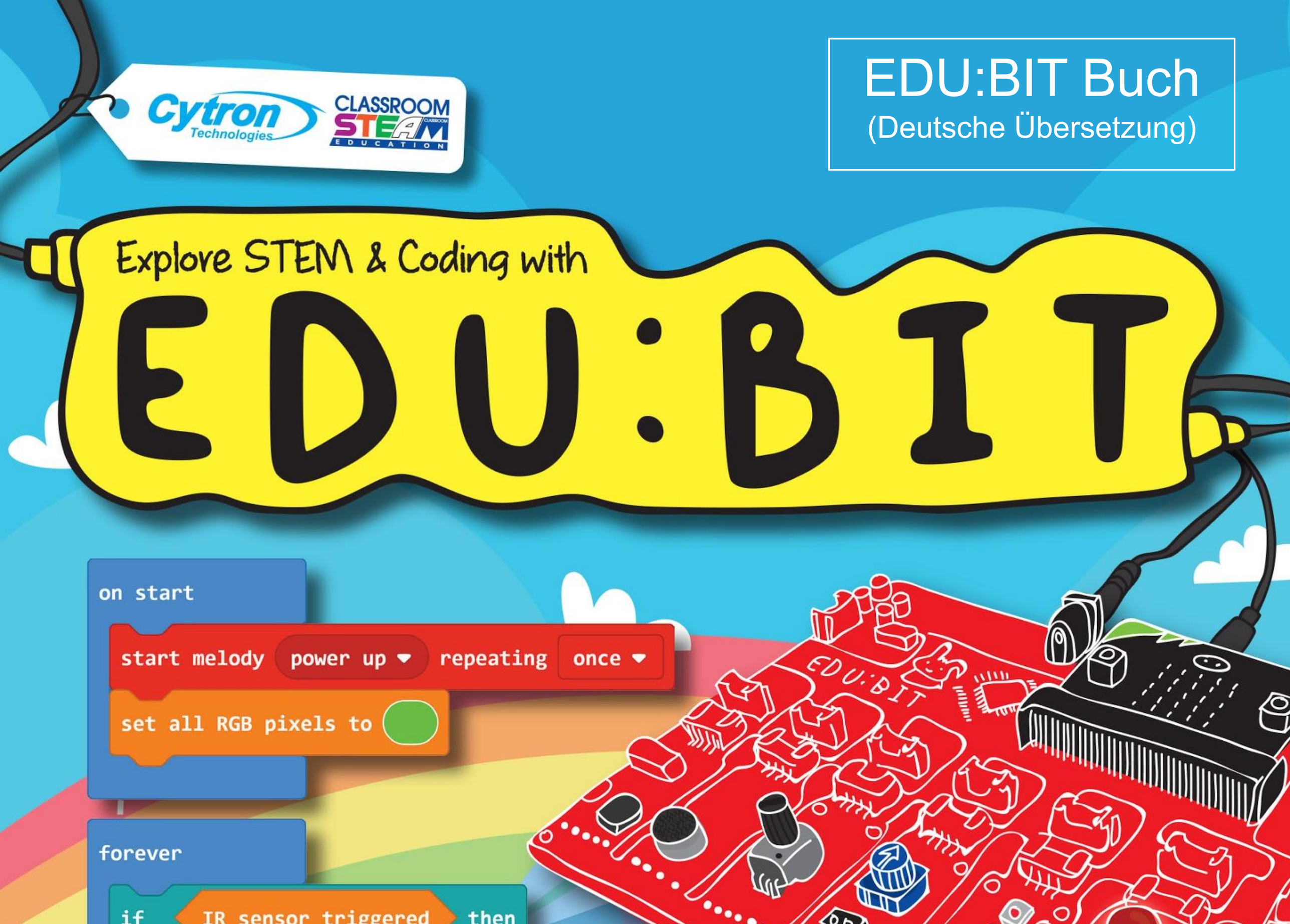
EDU:BIT Buch

(Deutsche Übersetzung)

Explore STEM & Coding with

EDU:BIT



NACHRICHT VON EDUTEAM @ CYTRON

Hallo _____ !
(Name des Kindes)

Hast du schon einmal von micro:bit gehört? Das ist eine kleine programmierbare Platte, auch Einplatinencomputer genannt, die in Großbritannien erfunden und in der ganzen Welt verbreitet wurde, damit Kinder auf einfache Art und Weise programmieren lernen können.

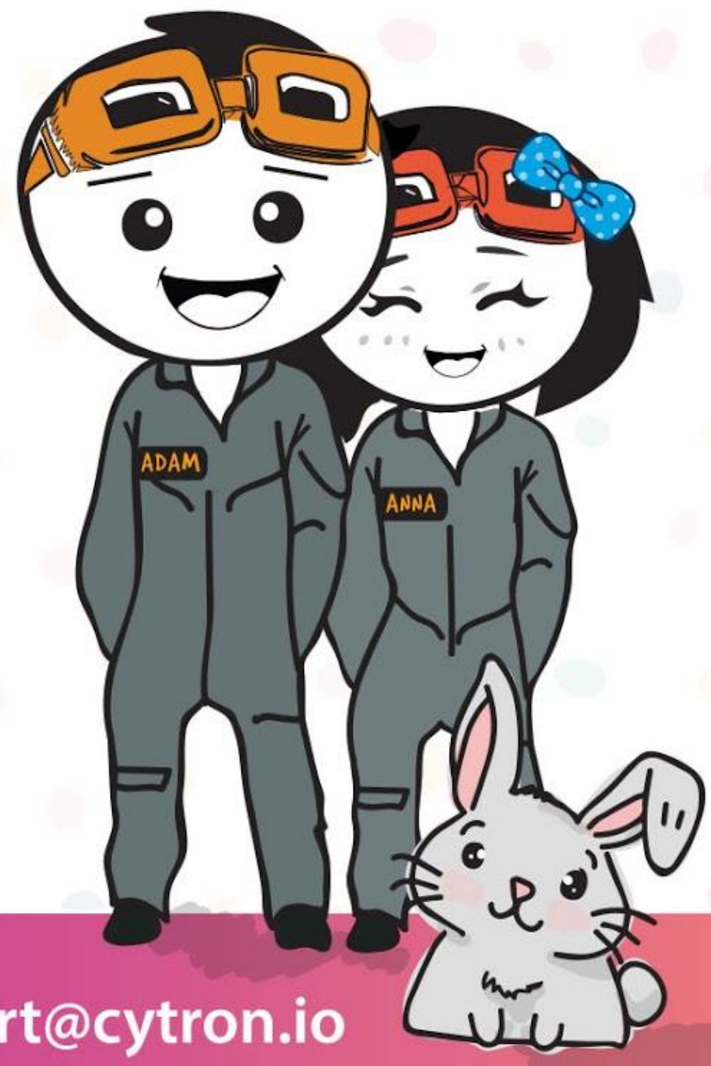
Technische Fachleute bei Cytron haben micro:bit noch weiter entwickelt, indem sie EDU:BIT gebaut haben, mit dem du Schritt für Schritt lernen kannst, Code zu schreiben. EDU:BIT besteht aus Music Bit, mit dessen Piezo-Summer und Kopfhöreranschluss du Musik spielen kannst, Sound Bit zum Messen von Geräuschen, Potentio Bit zur analogen Eingabe, IR Bit mit einem Infrarotsensor, RGB Bit für farbenfrohe Lichtspiele, Ampel Bit mit roten, gelben und grünen LEDs und schließlich Button Bit, eine größere Version der Knöpfe auf der micro:bit-Platine. In diesem Baukasten findest du auch einen Gleichstrommotor und einen Servomotor. Das ist richtig cool!

Lass uns loslegen! In diesem Buch werden wir einige klassische Kinderspiele wie “Schere, Stein, Papier”, das Leiterspiel, Abfangen, Talentshow, Twister, “Simon sagt” und viele andere lustige Spiele entdecken und programmieren. Folge einfach Schritt für Schritt unseren Anleitungen um die Spiele zu entwickeln. Zeige sie deinen Freundinnen und Freunden und habt Spaß beim Spielen! Verändere den Code der Spiele ruhig nach deinen eigenen Vorstellungen.

Am Ende eines jeden Kapitels findest du einen kleinen Test, bei dem du zeigen kannst, was du gelernt hast, indem du eine App für deine Klasse schreibst. Probiere es aus und wenn du irgendwo hängen bleibst, sind wir hier um dir zu helfen.

Bist du bereit? Lass uns diese spannende Reise beginnen!
Viel Spaß beim Lernen und Entdecken!

Wir sehen uns!
Adam & Anna



Erforsche MINT und Programmieren mit dem EDU:BIT Training & Project Kit

Geschrieben von

Cheryl Ng, SC Lim & Adrian Teo

Der Inhalt wurde getestet von

Joshayne D. Lim (7 Jahre alt)

Illustrationen von

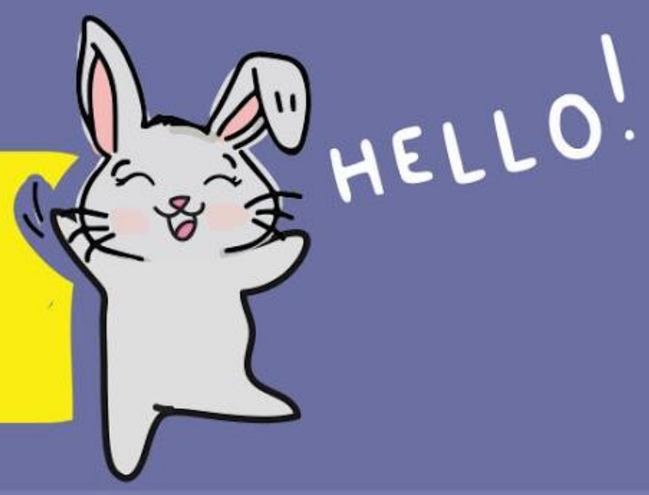
Suhana Oazmi

Übersetzung von Michael Steinkellner

2020

Publiziert von Cytron Technologies

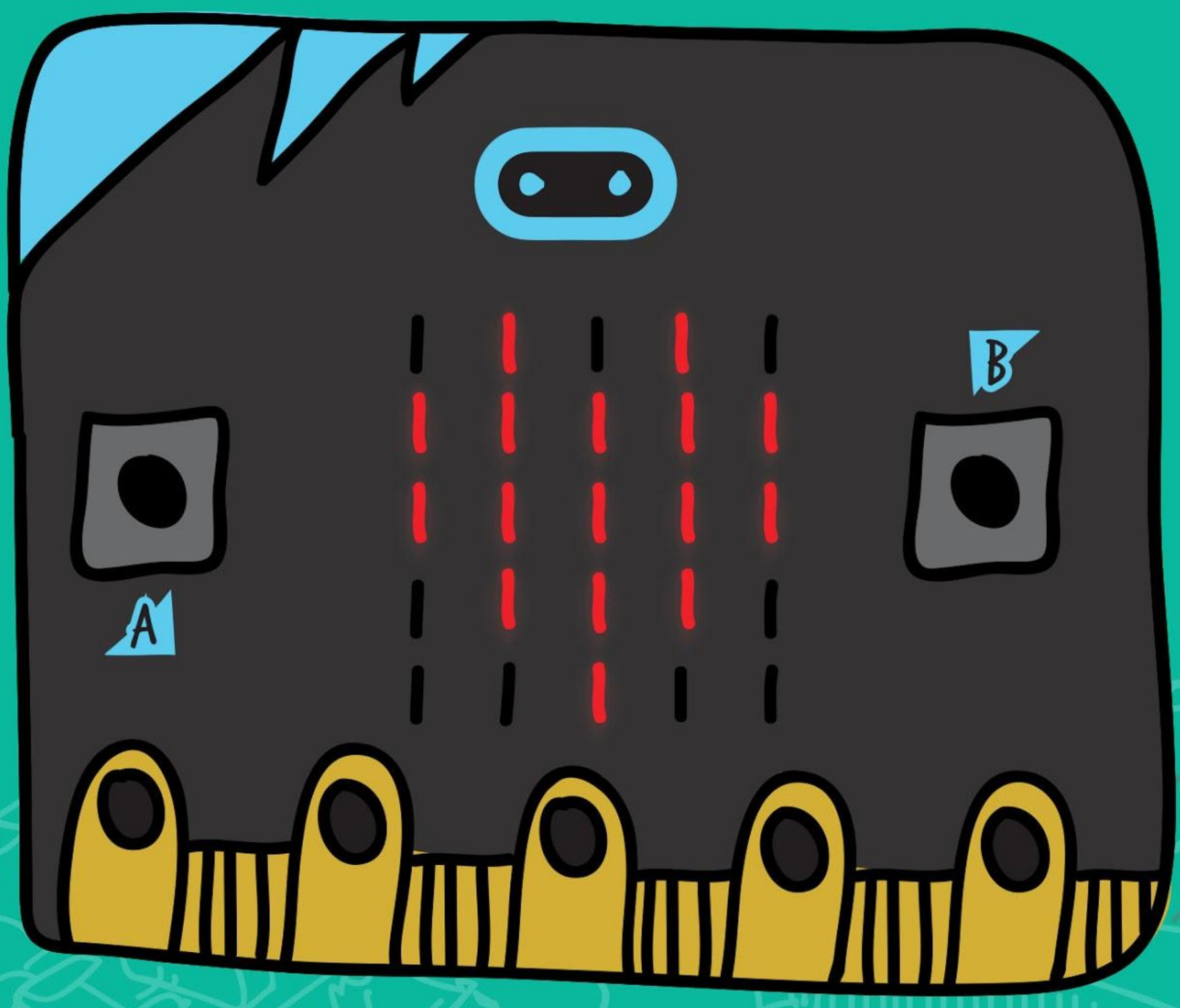
Inhalt



Kapitel 1

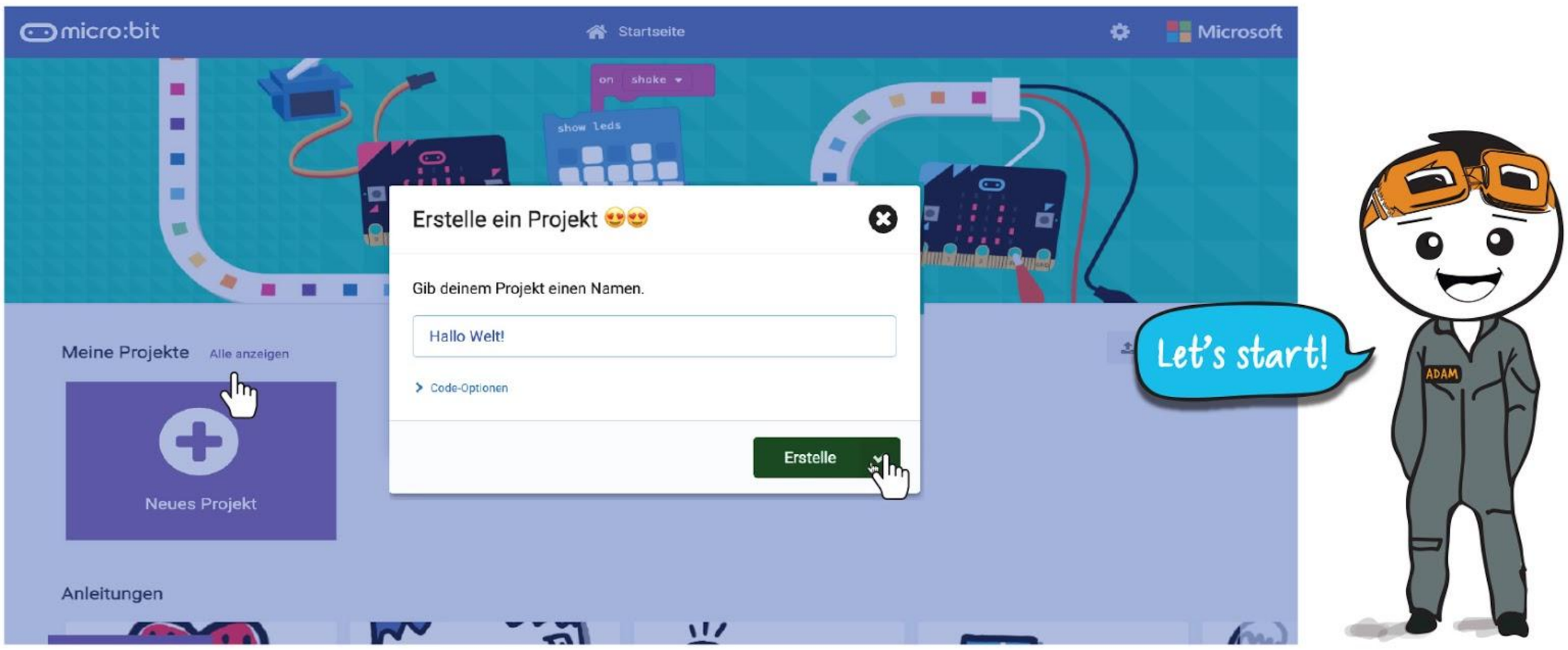
> Hallo Welt!_

Die LED-Matrix auf dem micro:bit



LASS UNS PROGRAMMIEREN!

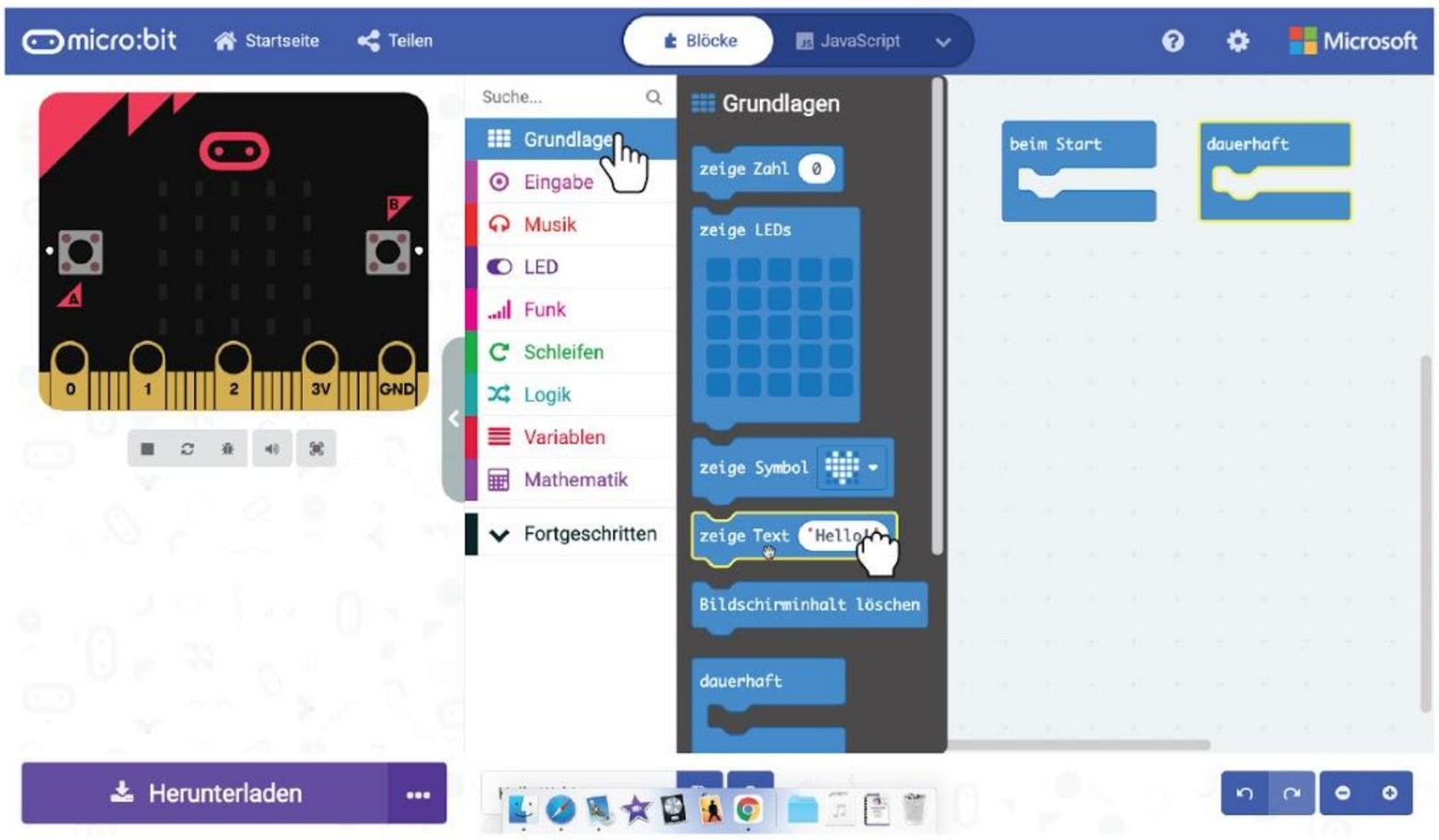
SCHRITT 1 Öffne deinen Browser und gehe zu **https://makecode.microbit.org/**
Klicke auf '**Neues Projekt**'. Tippe einen Projektnamen ein und klicke auf '**Erstelle**'.



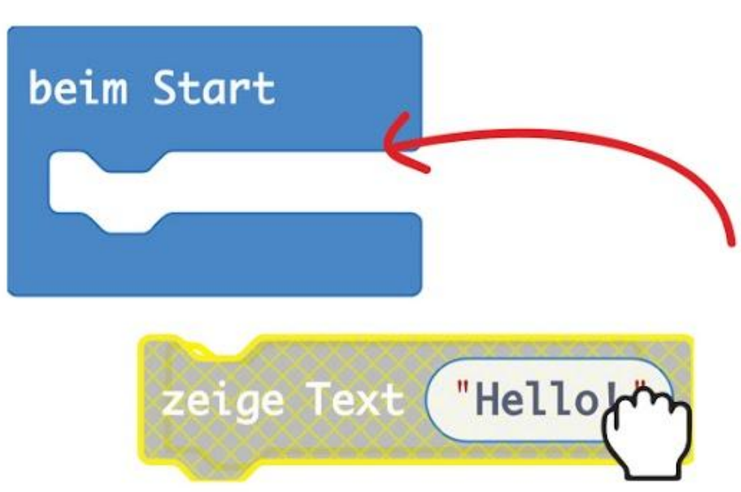
Du siehst jetzt den Block-Editor von Microsoft MakeCode, mit dem du Programme erstellen kannst, indem du farbige Blöcke zusammenbaust. Die Blöcke kannst du ganz einfach anklicken, ziehen und wieder ablegen.

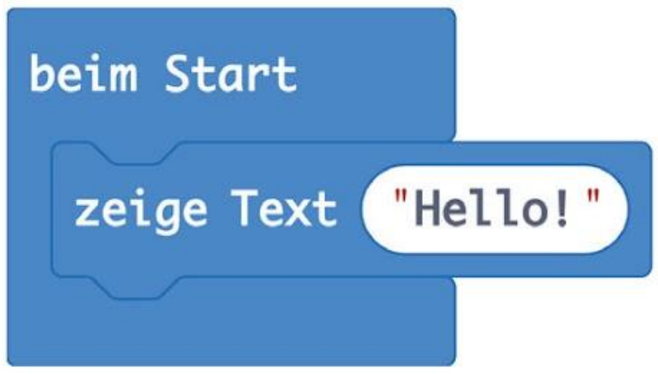
1) Veröffentliche und teile dein Projekt.
2) Programmiere mit Blöcken, Javascript oder Python.
3) Öffne das Hilfemenü.
4) Ändere Einstellungen, füge Erweiterungen hinzu und kopple Geräte.
5) **SIMULATOR** - Zeigt in einer Vorschau, wie dein Programm auf dem micro:bit aussehen wird.
6) **WERKZEUG / VERZEICHNIS** - Klicke um dir alle verfügbaren Code-Blöcke in jeder Gruppe anzusehen. Die Blöcke derselben Gruppe haben alle die gleiche Farbe.
7) **ARBEITSBEREICH** - Hier kannst du Programme erstellen, indem du Blöcke zusammen baust.
8) Übertrage dein Programm auf deinen micro:bit.
9) Hier kannst du dein Programm umbenennen und auf deinen Computer herunterladen.
10) Damit kannst du ein GitHub-Repository erstellen.
11) Rückgängig machen und wiederholen.
12) Arbeitsfläche vergrößern/verkleinern.

Schritt 2 Klicke auf **[Grundlagen]** und dann auf den Block **[zeige Text]**.

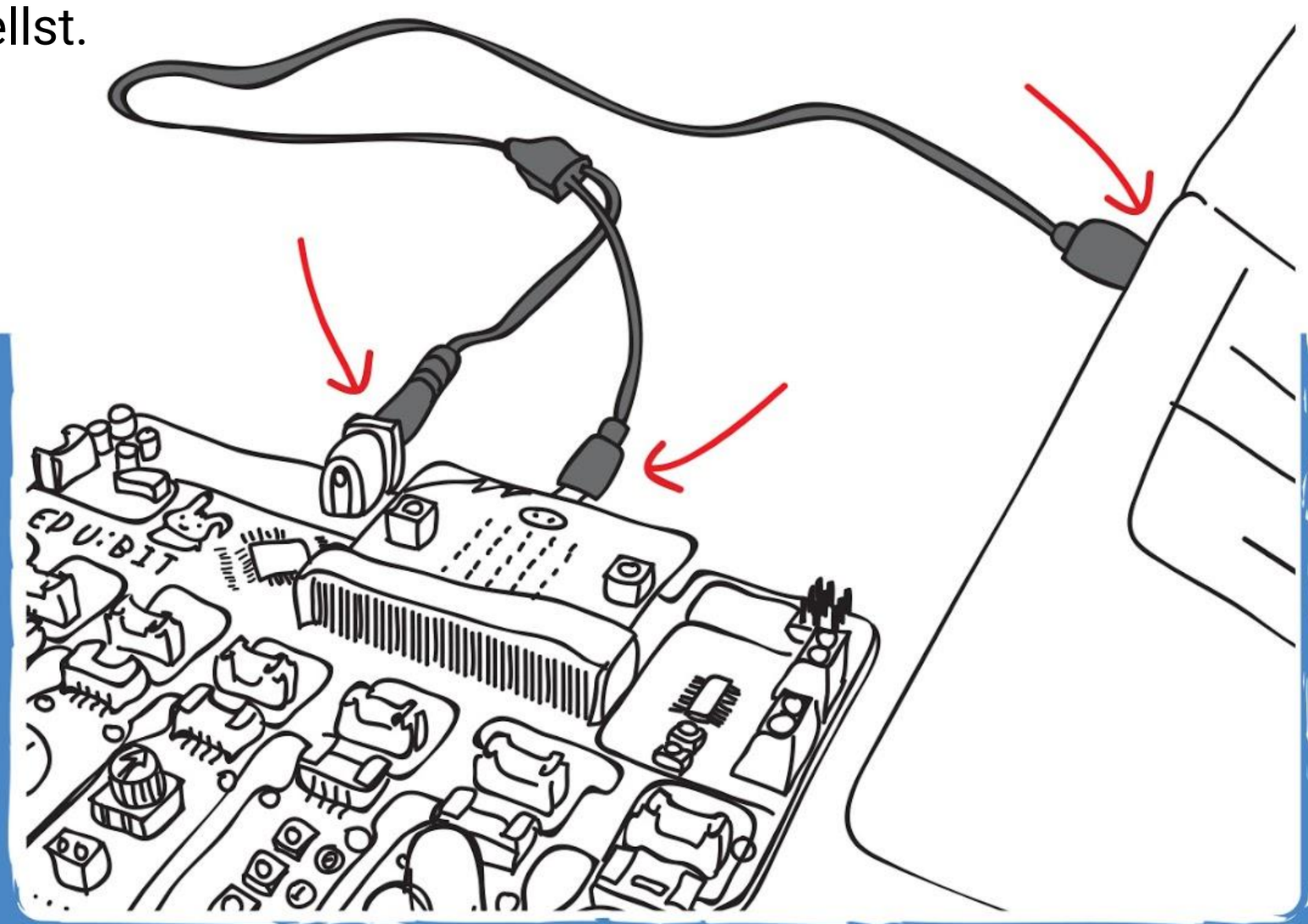


Schritt 3 Klicke auf den Block **[zeige Text]**, ziehe ihn in den Block **[beim Start]** und lasse ihn dort los.

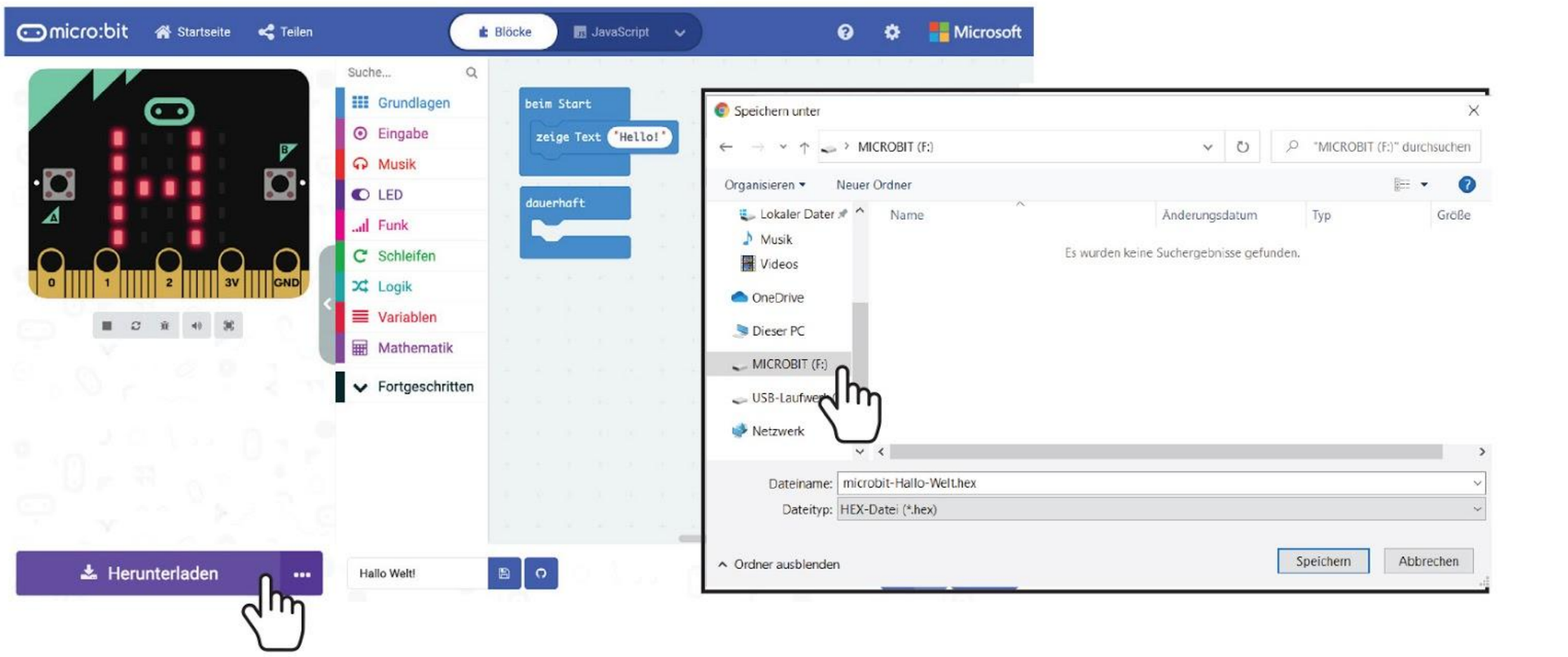




Schritt 4 Verbinde EDU:BIT und deinen Computer mithilfe des USB-Kabels miteinander wie in dem Bild gezeigt. Schalte EDU:BIT ein, indem du den Schalter auf ON stellst.



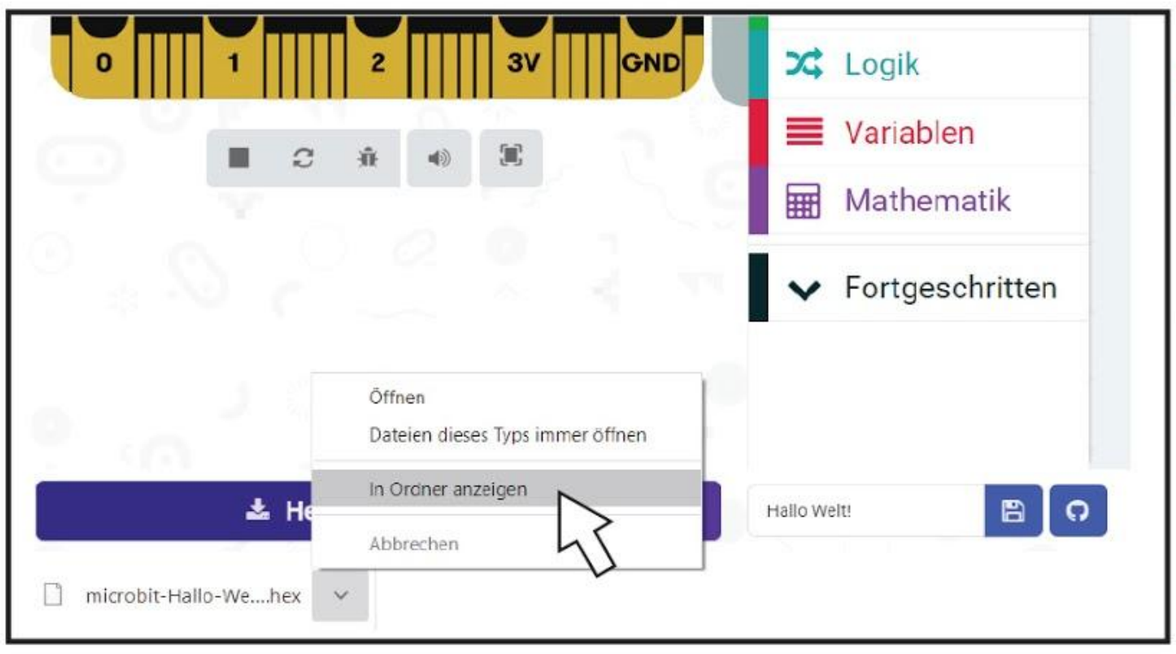
Schritt 5 Klicke auf die Schaltfläche **[Herunterladen]**. Wähle im Dialogfenster das Laufwerk „MICROBIT“ und klicke auf „Speichern“. Schließe das Fenster mit dem Text „Download abgeschlossen“.

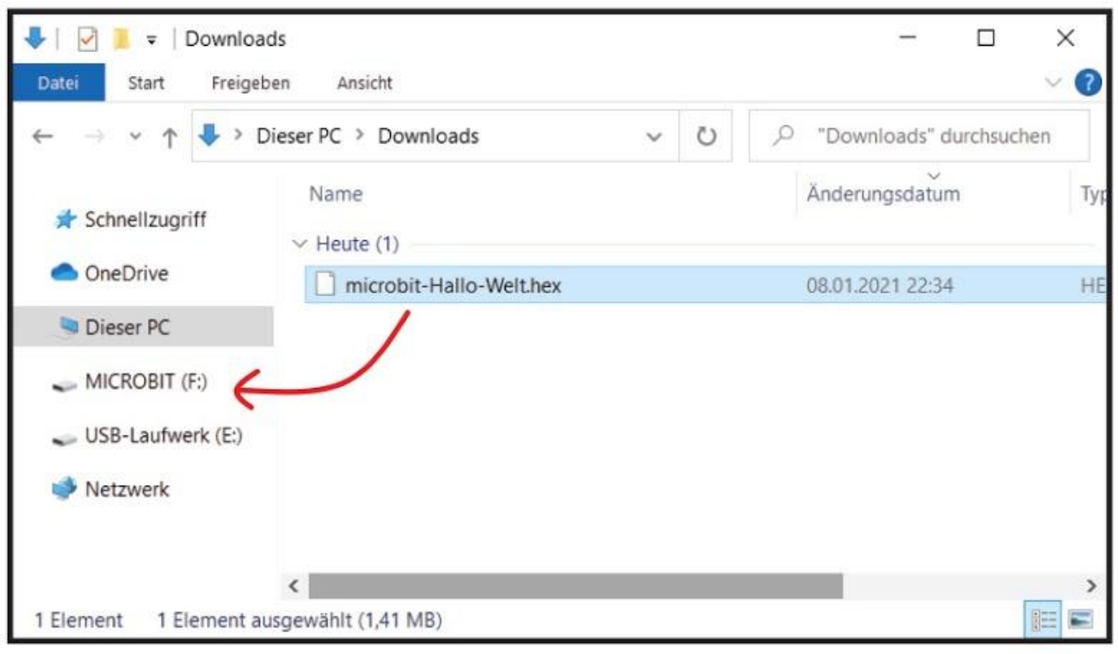


Dieser Vorgang, bei dem du dein Programm auf den micro:bit überträgst, wird Flashen genannt. Die orange LED auf der Rückseite des micro:bit blinkt während des Flashens und sobald das Programm fertig übertragen wurde, wird es automatisch gestartet.

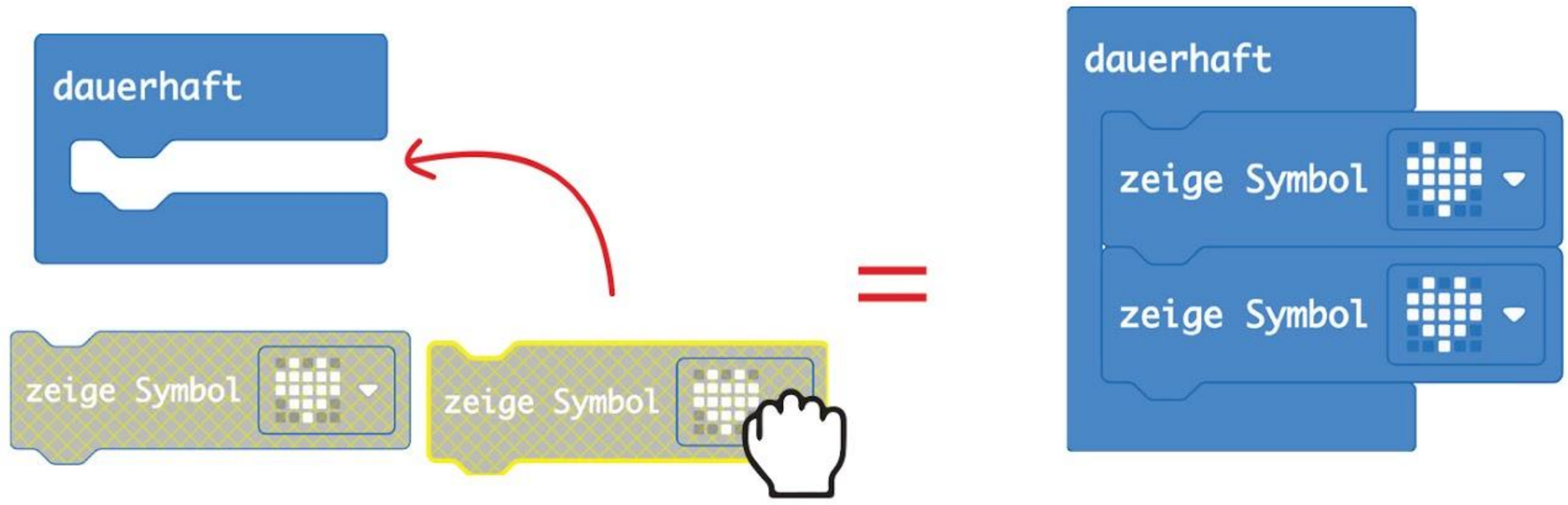
HINWEIS!

Wenn kein Dialogfeld erscheint, bedeutet das, dass deine Datei automatisch im Download-Ordner deines Internet-Browsers abgespeichert wurde. Klicke mit der rechten Maustaste auf die .hex-Datei am unteren Rand deines Browsers und klicke auf „Zeige in Ordner“. Ziehe die heruntergeladene Datei „microbit-xxxx.hex“ auf das MICROBIT-Laufwerk, so als würdest du die Datei auf einen USB-Stick kopieren.

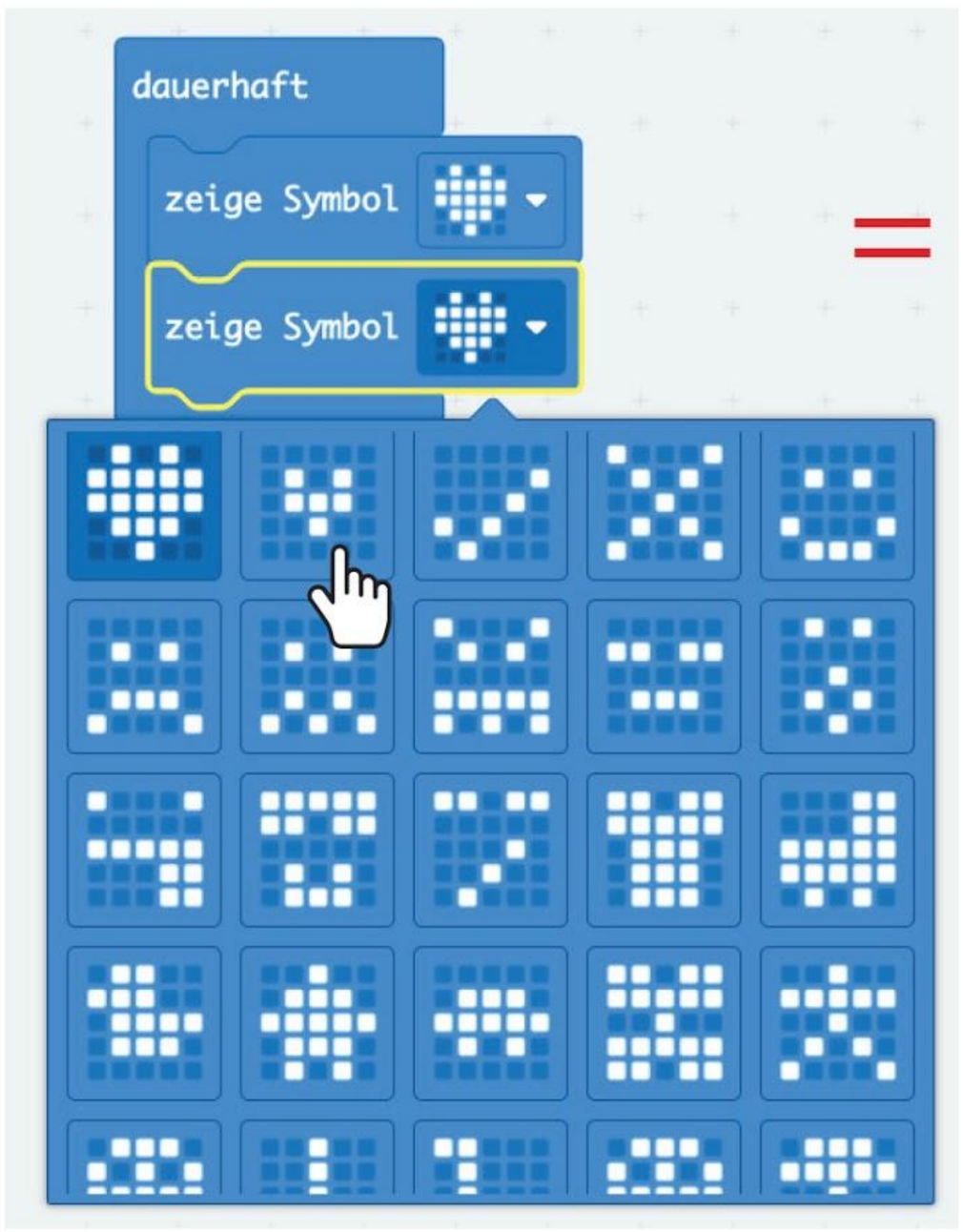


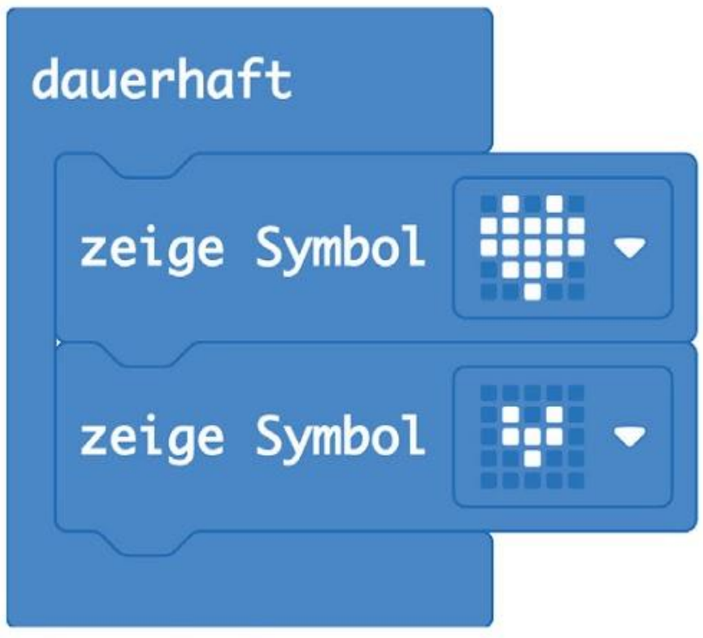


Schritt 6 Klicke auf **[Grundlagen]** und dann auf **[zeige Symbol]**. Wiederhole diesen Schritt um einen zweiten **[zeige Symbol]**-Block hinzuzufügen. Ziehe beide Blöcke in den Block **[dauerhaft].**



Schitt 7 Klicke auf das Symbol im zweiten **[zeige Symbol]**-Block und wähle das kleine Herz im Pop-up-Fenster aus. Flashe den Code auf deinen EDU:BIT.

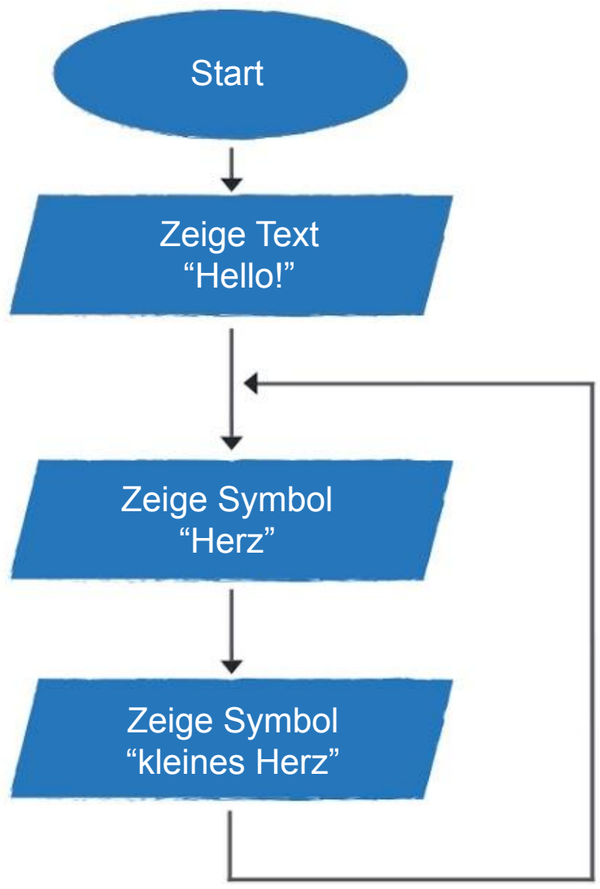




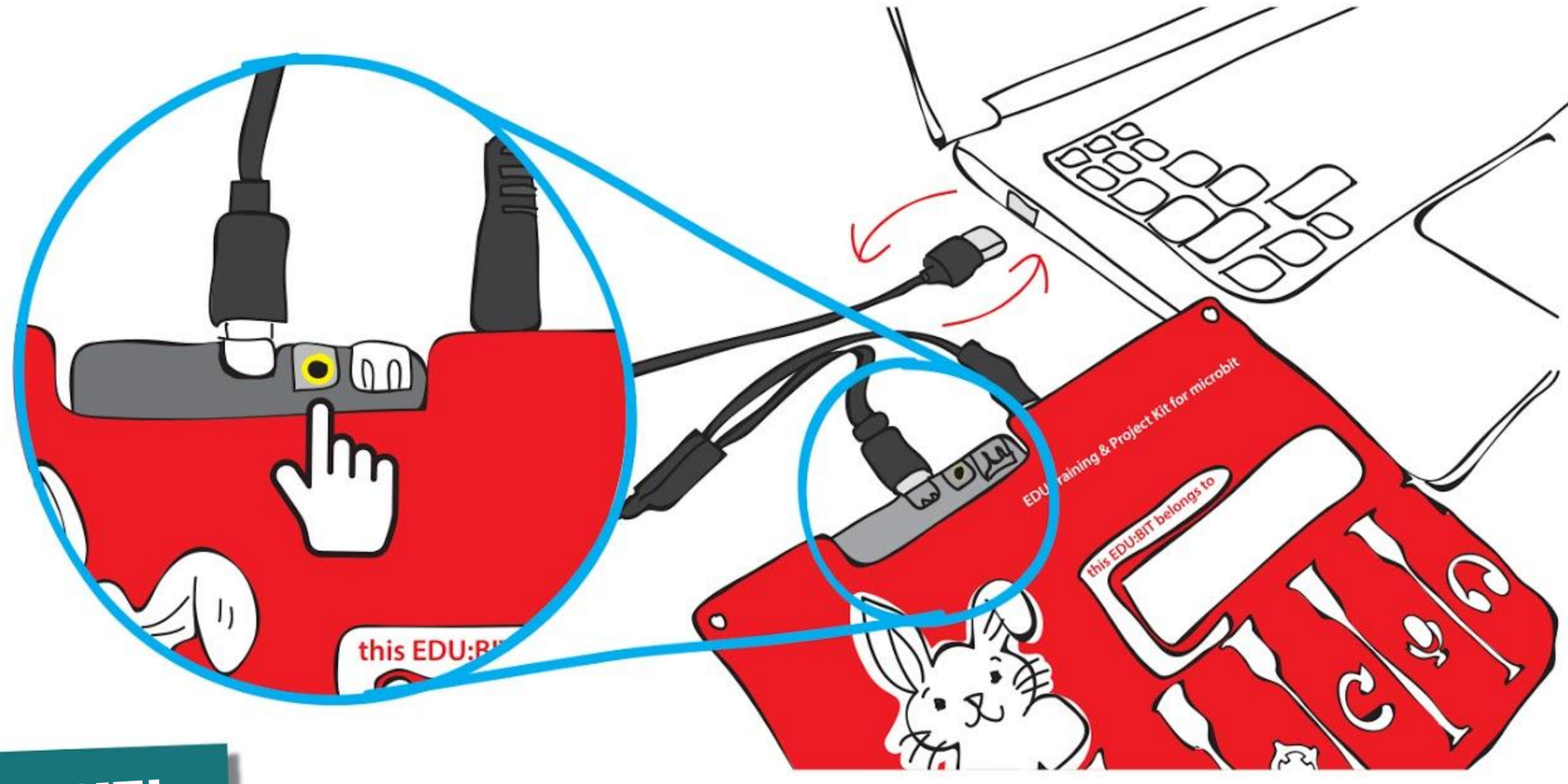
Es ist Liebe auf den ersten Blick. Siehst du die Animation mit dem schlagenden Herz?

KNACKE DEN CODE

Ist dir aufgefallen, dass der Text „Hello!“ nur einmal angezeigt wird, aber das Herz weiter und weiter blinkt? Warum?



Der Block **[beim Start]** führt den Code nur einmal aus (beim Start). Der Block **[dauerhaft]** führt den Code immer wieder aus (dauerhaft).

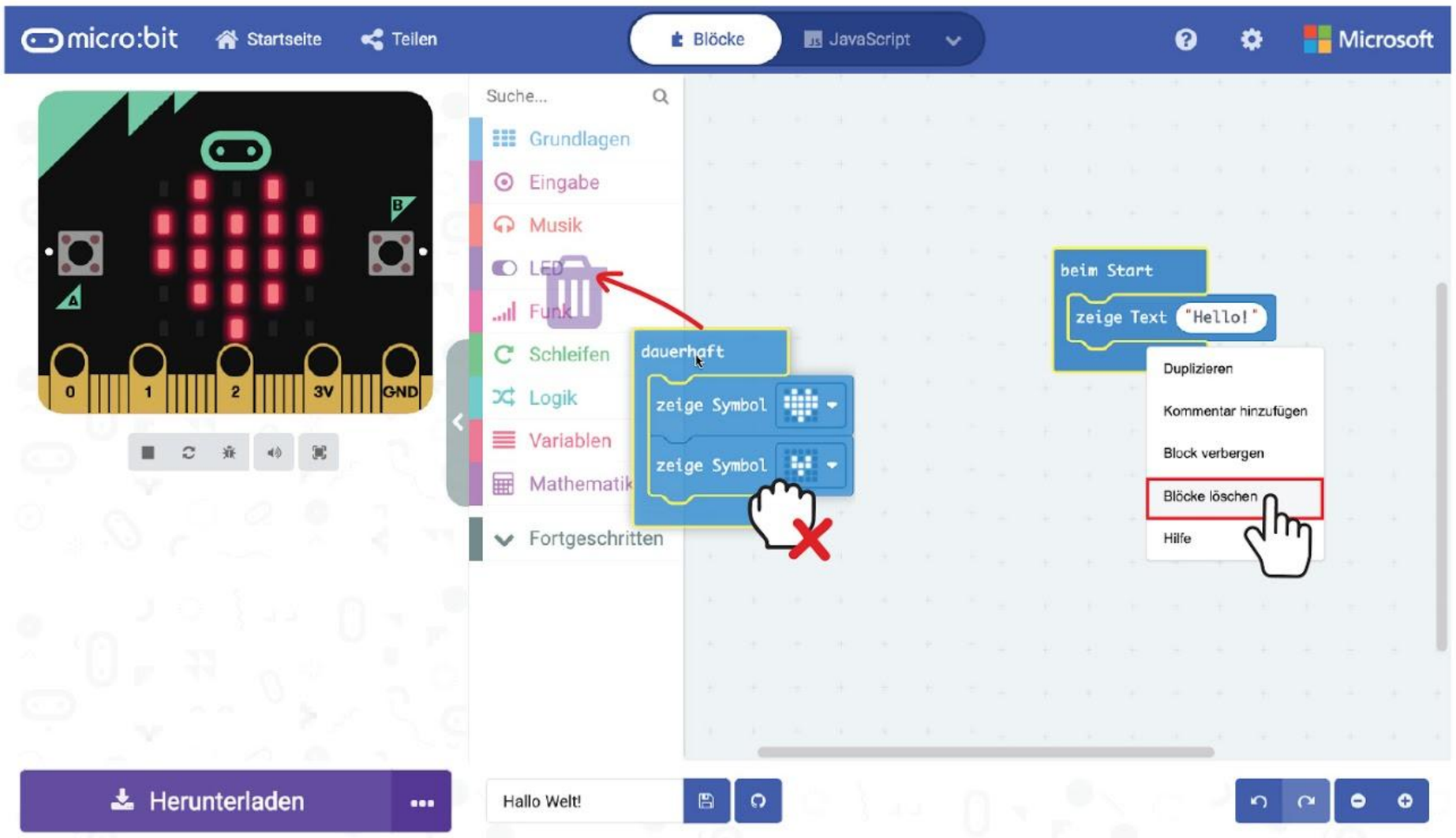


MERKE!

Wenn du dein Programm neu starten willst, drücke einfach den RESET-Knopf oder stecke das USB-Kabel aus und wieder ein.

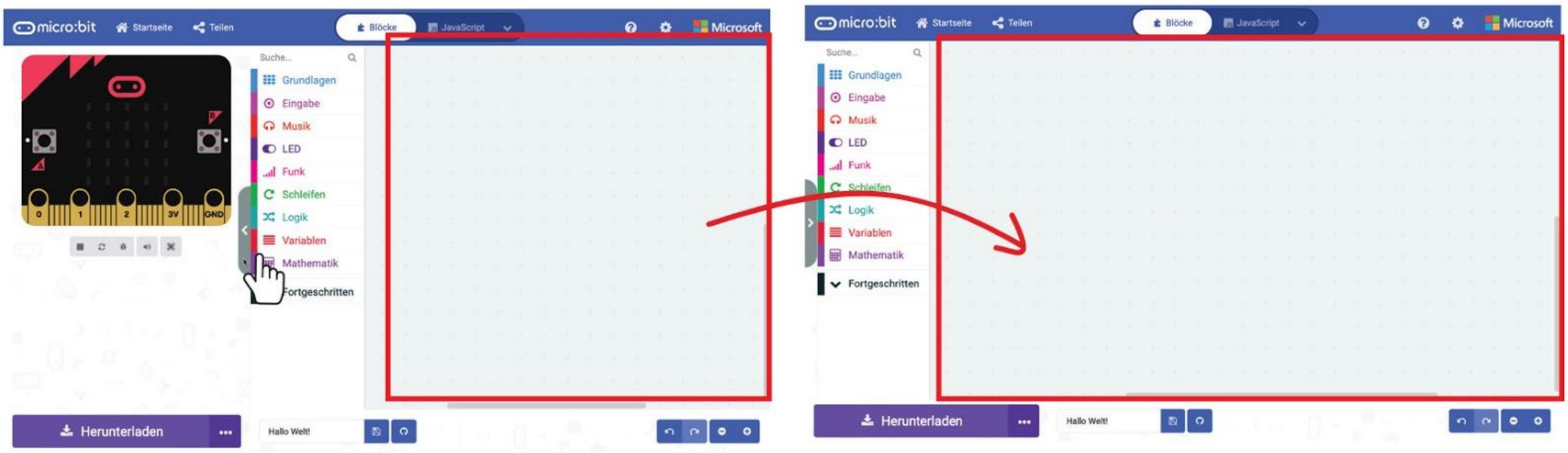
Tipp #1!

Du kannst Blöcke löschen, indem du sie in den Bereich „Werkzeug“ ziehst. Lass die Maustaste los, wenn das Papierkorb-Symbol erscheint. Du kannst auch mit der rechten Maustaste auf den Block klicken und „Block löschen“ auswählen.



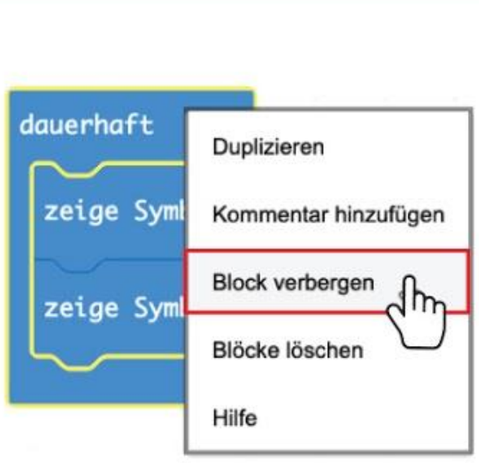
Tipp #2!

Wenn du den Simulator nicht benutzt, kannst du ihn ausblenden, indem du auf das Symbol rechts daneben klickst. So hast du mehr Platz für deine Code-Blöcke.



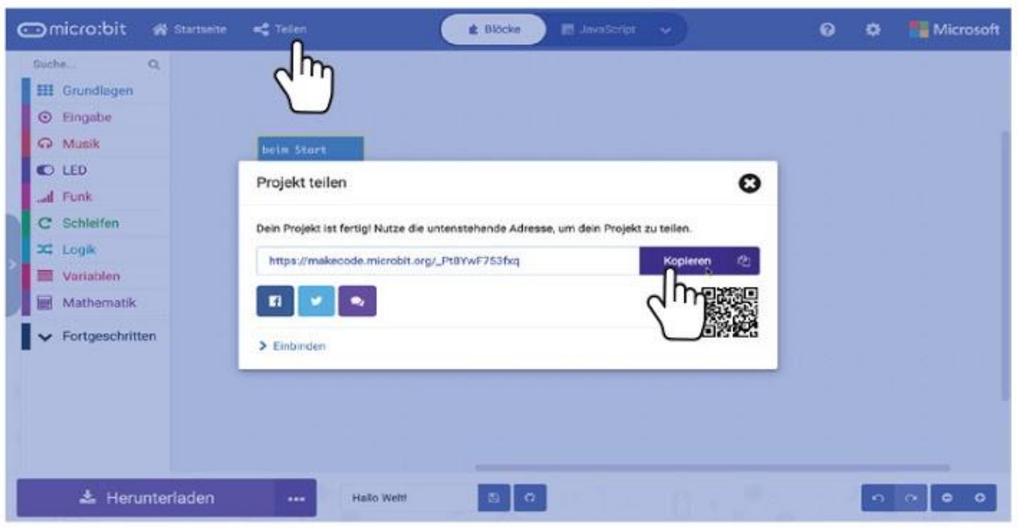
Tipp #3!

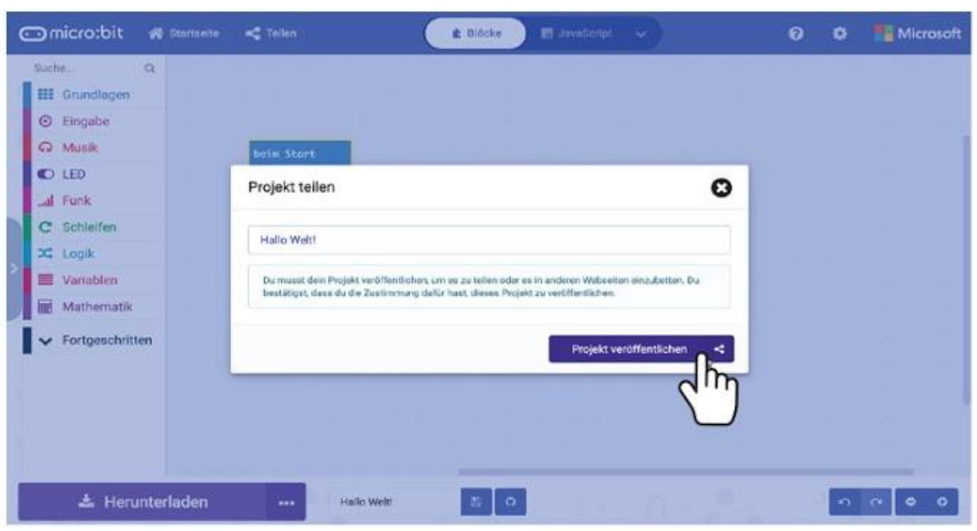
Du kannst eine Gruppe von Blöcken einklappen, indem du mit der rechten Maustaste auf den äußersten Block klickst und „Block verbergen“ auswählst. Um sie wieder auszuklappen, klicke einfach auf das kleine Symbol rechts.



Tipp #4!

Du kannst deinen Code mit deiner Lehrer/in oder deinen Freund/innen teilen, indem du das Projekt veröffentlichst und ihnen den Link zum Projekt schickst. Klicke auf [Teilen] und dann auf [Projekt veröffentlichen]. Du siehst dann ein neues Dialogfenster mit einem Link zu deinem Projekt. Klicke auf [Kopieren] um den Link in die Zwischenablage zu kopieren.



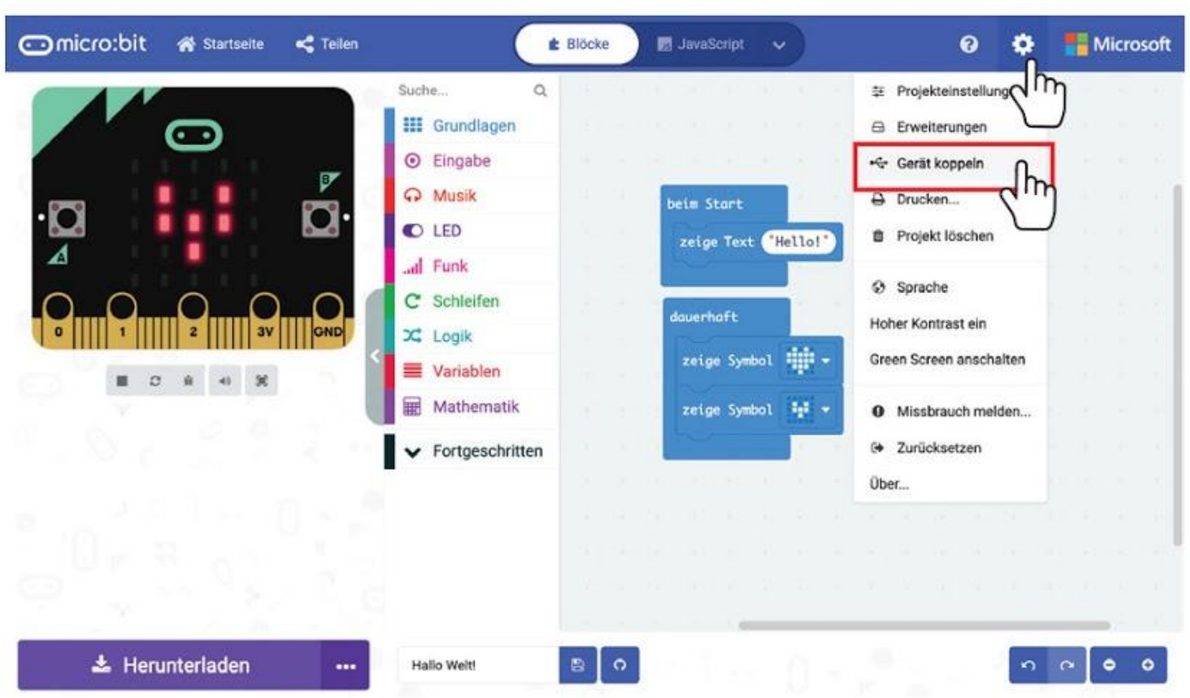


Tipp #5!

Deine Lehrer/innen und deine Freund/innen sehen die folgende Seite, wenn sie dein Projekt öffnen. Sie können sich deinen Code anschauen und auf **[Edit Code]** klicken, um ihn zu bearbeiten.

Tipp #6!

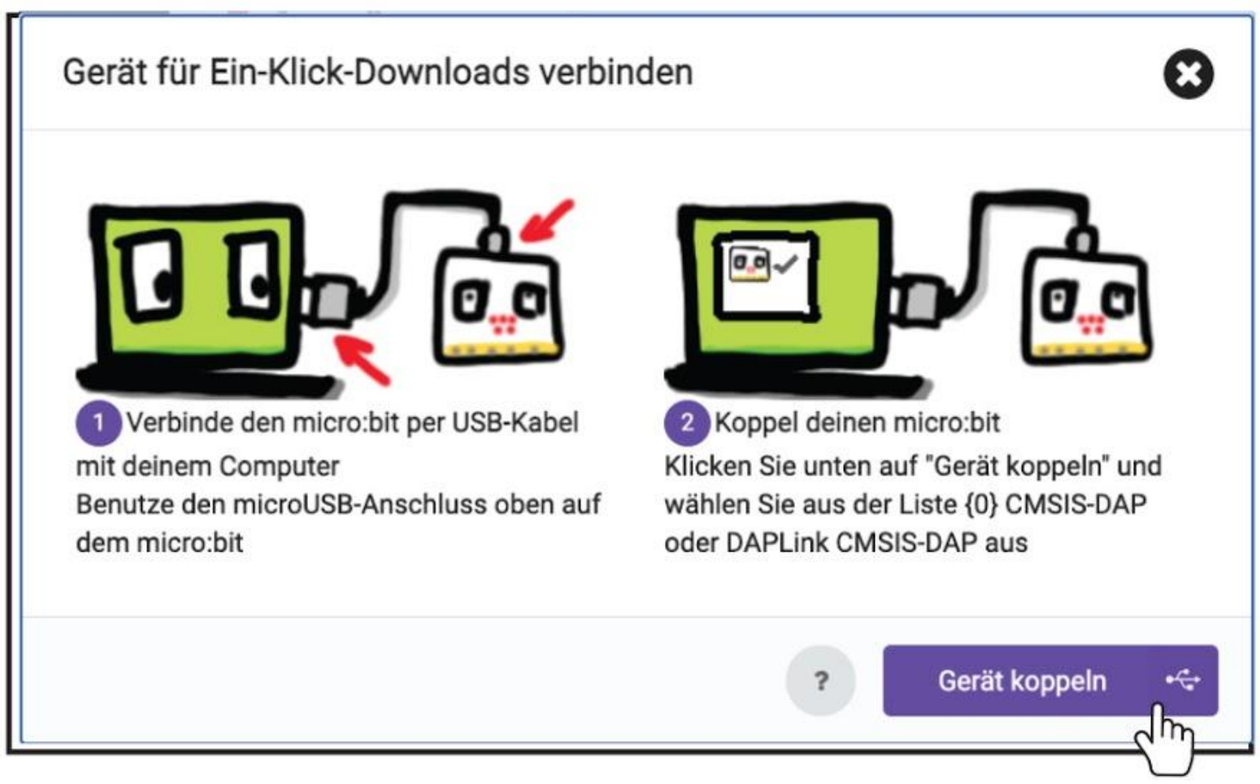
Wusstest du, dass du dein Gerät mit dem Editor koppeln und es dann mit einem einzigen Klick flashen kannst? Klicke einfach auf „Gerät koppeln“.



ACHTUNG!

Du musst die neueste Firmware auf deinem micro:bit haben und entweder Edge oder Chrome als Browser nutzen. Verwende diese Anleitung, wenn du deine Firmware aktualisieren musst - https://microbit.org/get-started/user-guide/firmware/

Sorge dafür, dass dein EDU:BIT mit deinem PC verbunden ist und klicke auf **[Gerät koppeln]**. Wähle „BBC micro:bit CMSIS-DAP“ oder „DAPLink CMSIS-DAP“ aus der Liste aus und klicke auf **[Verbinden]**.

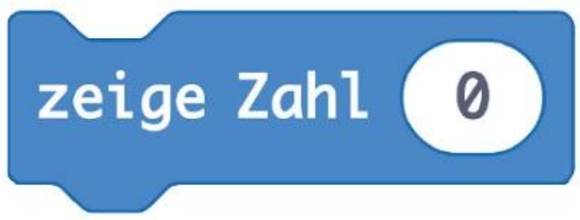


Nachdem du dein Gerät gekoppelt hast, kannst du deinen Code direkt auf deinen EDU:BIT übertragen, indem du auf **[Herunterladen]** klickst. Probiere es aus!

Wenn du Probleme damit hast, dein Gerät zu koppeln, findest du Hilfe unter https:/makecode.microbit.org/device/usb/webusb/troubleshoot

ENTDECKE NOCH MEHR

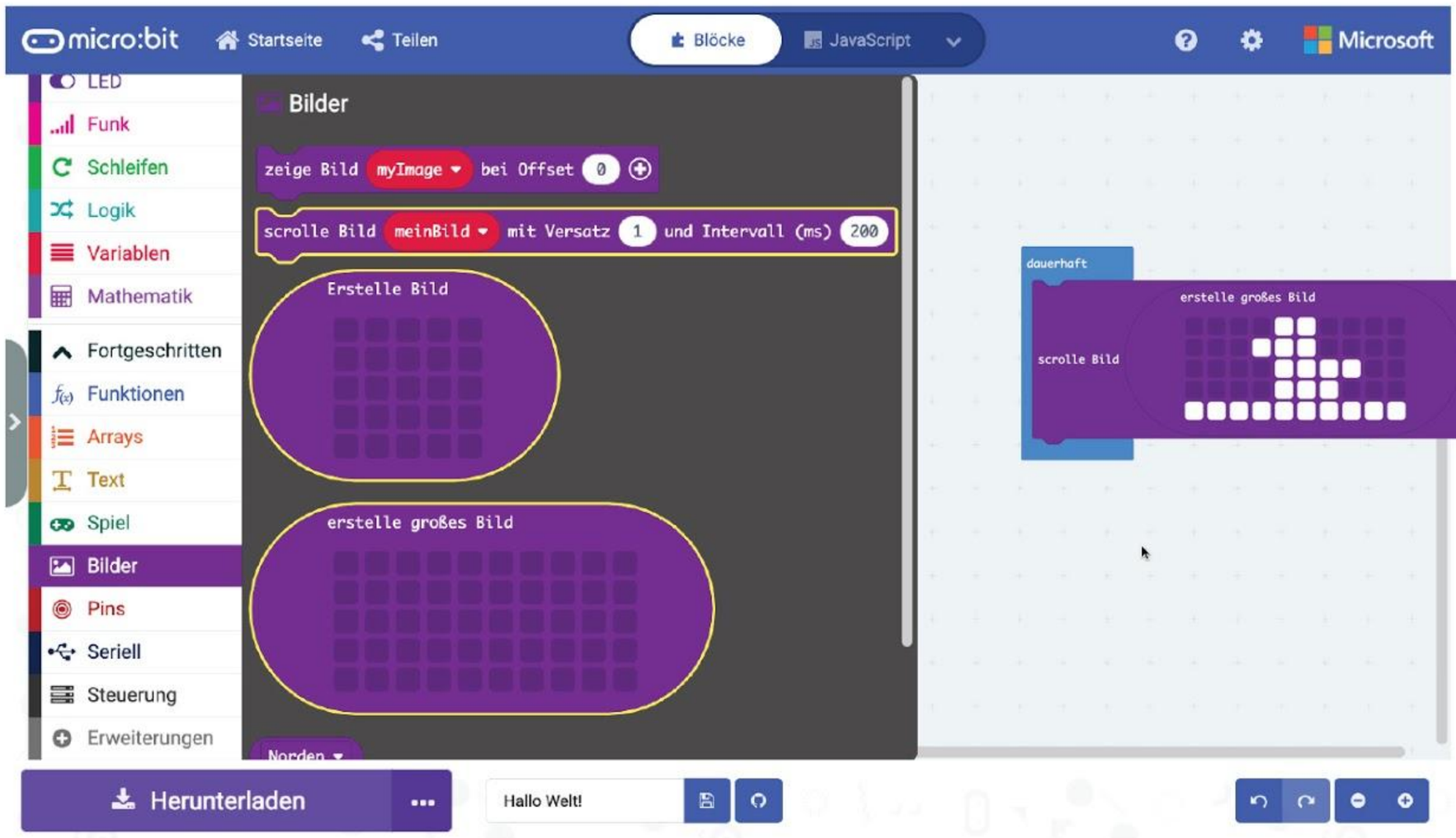
#1 Benutze den **[zeige LEDs]**-Block um deine eigenen Symbole zu zeichnen und den **[zeige Zahl]**-Block um Zahlen anzuzeigen.



Wusstest du?
1000 Millisekunden (ms) = 1 Sekunde

#2 Füge einen **[pausiere]**-Block hinzu um das Programm langsamer zu machen. Diese Funktion pausiert das Programm für die Anzahl an Millisekunden (ms), die du eingibst.

#3 Um ein Bild über die LED-Matrix laufen zu lassen, benutze den Block **[scrolle Bild _ mit Versatz _ und Intervall (ms) _]** zusammen mit **[erstelle Bild]** oder **[erstelle großes Bild]**. Du findest diese Blöcke im Bereich **Fortgeschritten** unter **[Bilder]**.



Wenn du das Programm ausführst, wirst du kleine Entenküken hintereinander über die LED-Matrix wandern sehen.

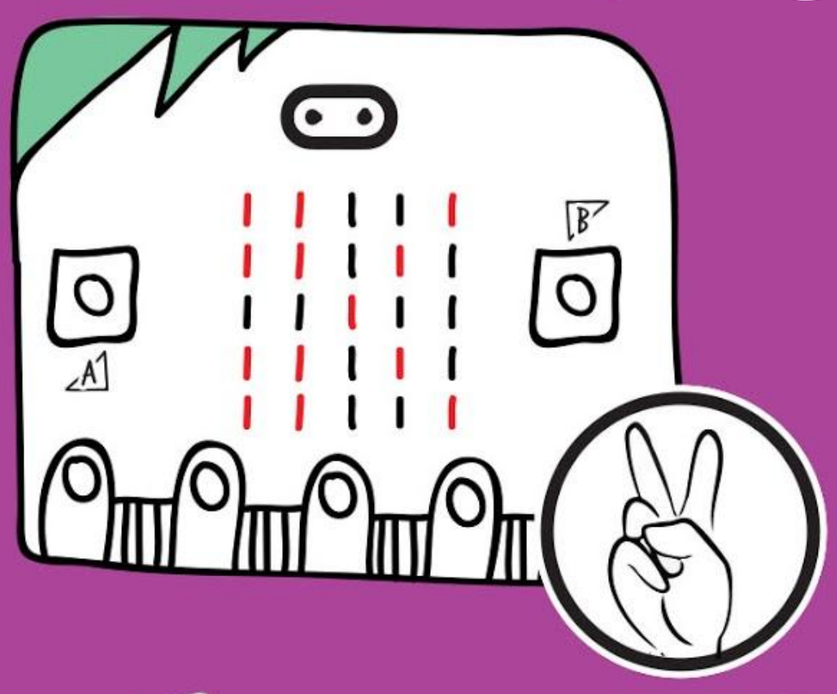
HERAUSFORDERUNG

Programmiere mit EDU:BIT ein digitales “Schwarzes Brett”.	
beim Start	Zeige eine einfache Animation als Intro und zeige dann den Namen deiner Klasse an.
dauerhaft	Zeige das heutige Datum und andere wichtige Informationen für deine Klasse an.

KAPITEL 2

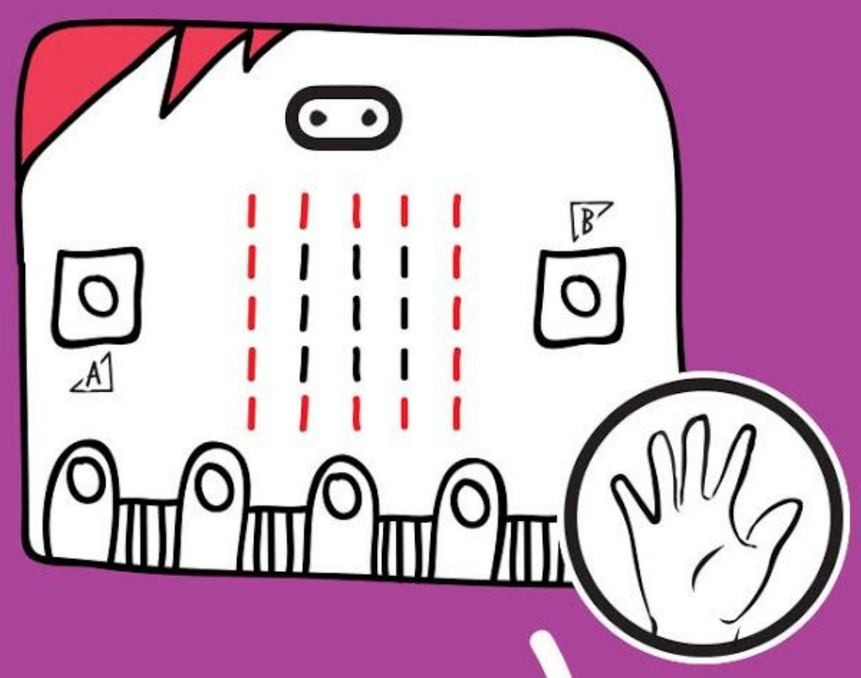
Spielen wir “Schere, Stein, Papier”!

Drück die Knöpfe auf micro:bit und Button Bit



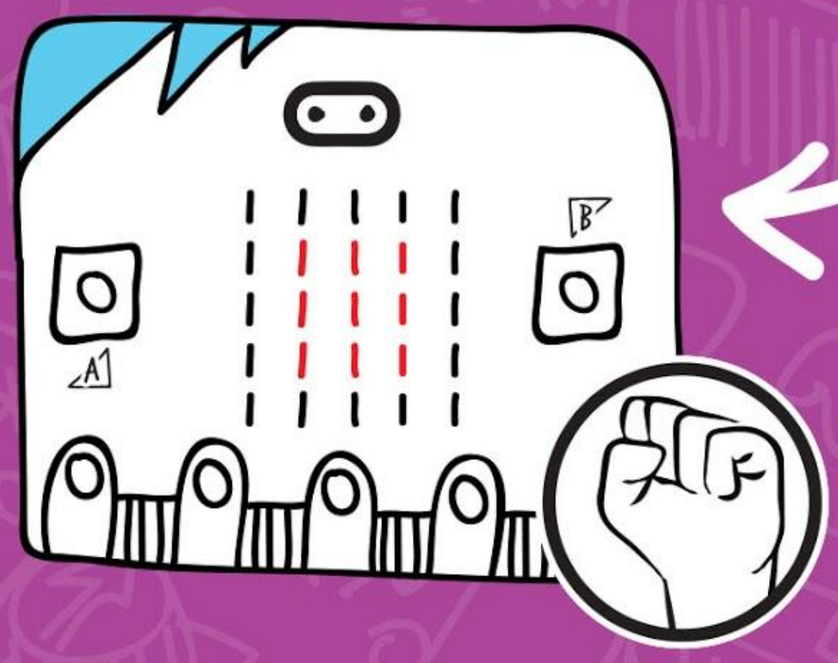
SCHERE

schlägt Papier



PAPIER

Schlägt Stein



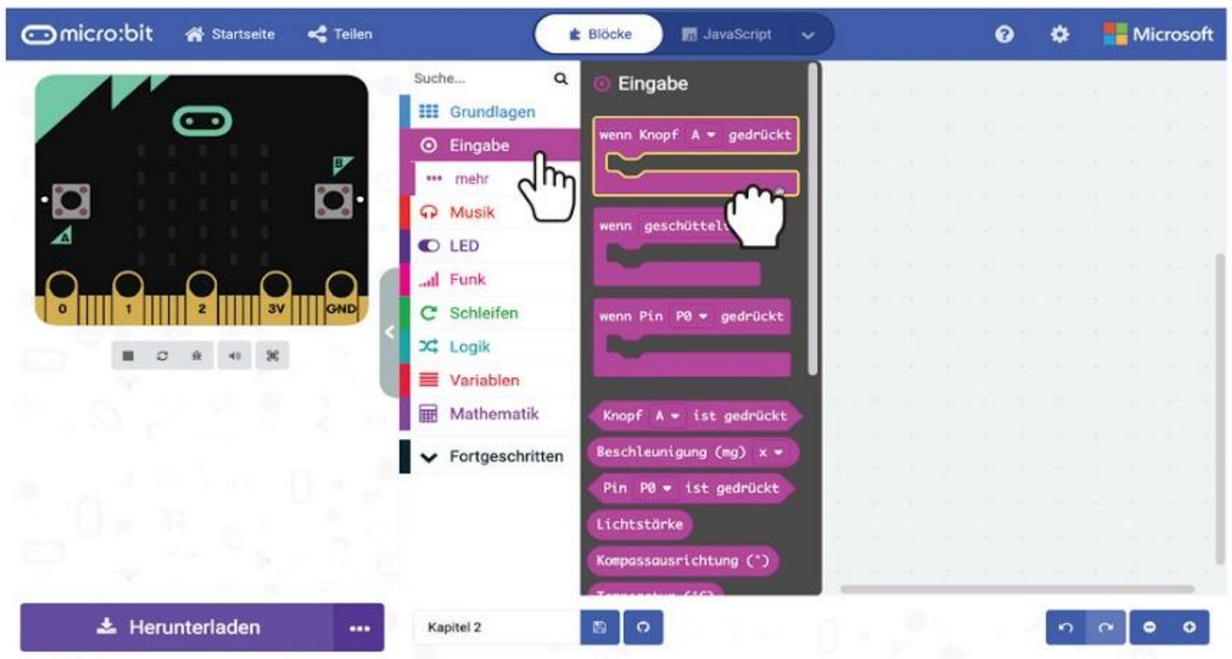
STEIN

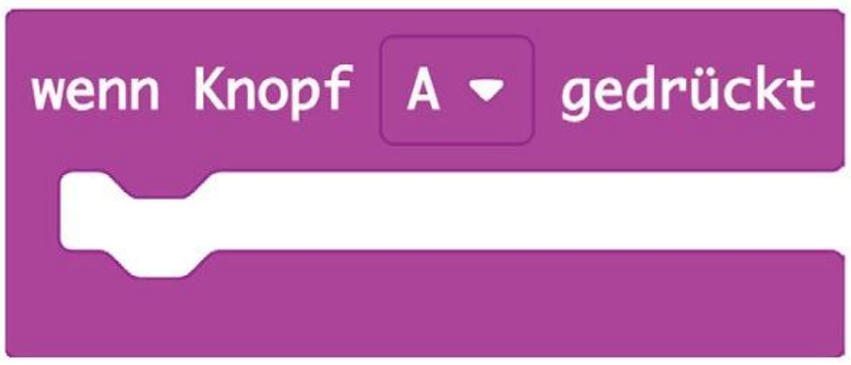
Schlägt Schere

link.cytron.io/edubit-chapter-2

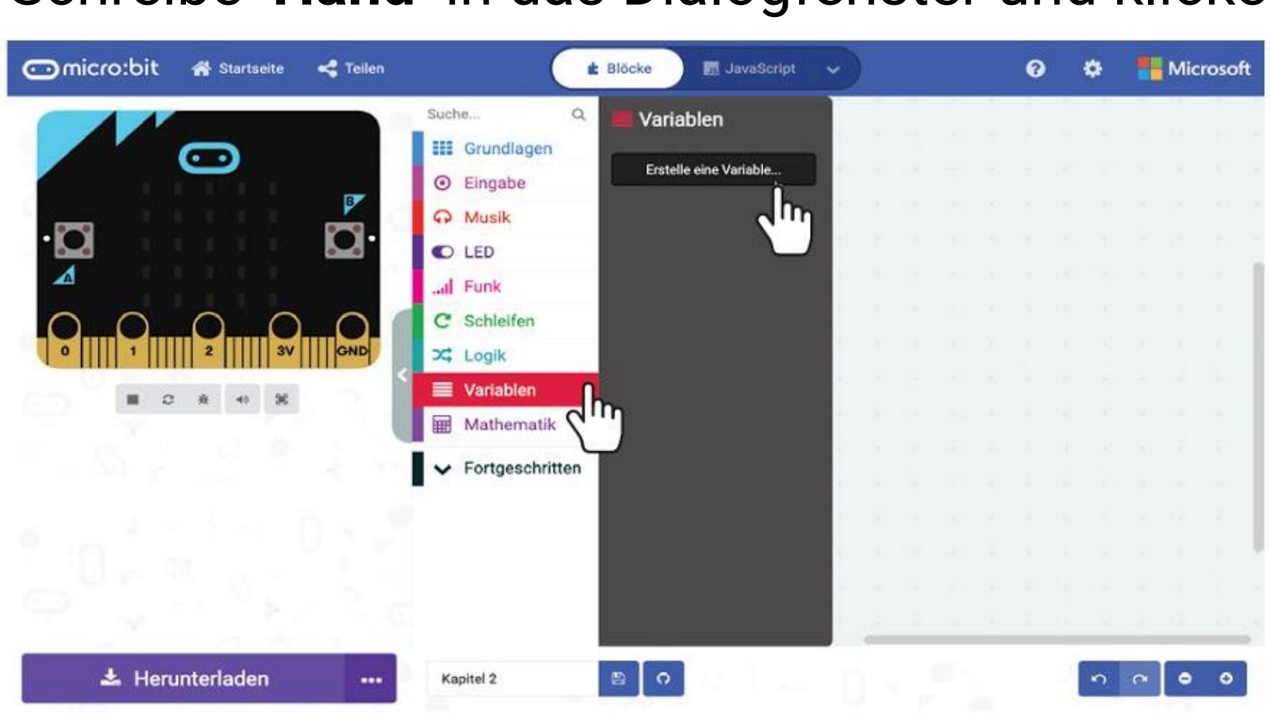
LASS UNS PROGRAMMIEREN!

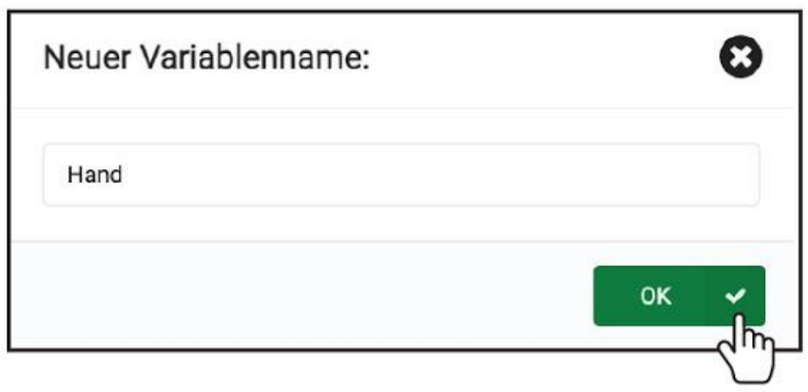
Schritt 1 Gehe zu **https://makecode.microbit.org/** (oder klicke einfach auf **Startseite**, wenn du den MakeCode Editor schon geöffnet hast). Erstelle ein neues Projekt. Klicke auf die Gruppe **[Eingabe]** und dann auf **[wenn Knopf _ gedrückt]**.



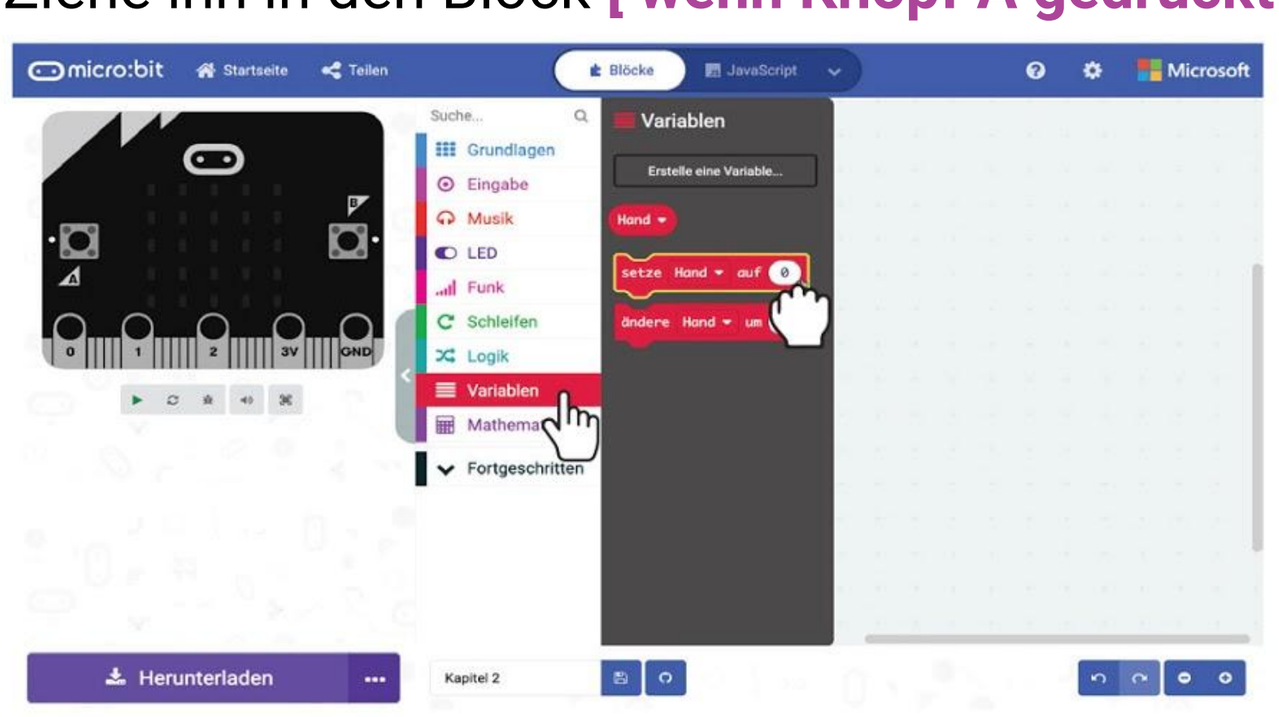


Schritt 2 Klicke auf die Gruppe **[Variablen]** und auf **[Erstelle eine Variable]**. Schreibe **'Hand'** in das Dialogfenster und klicke auf OK.

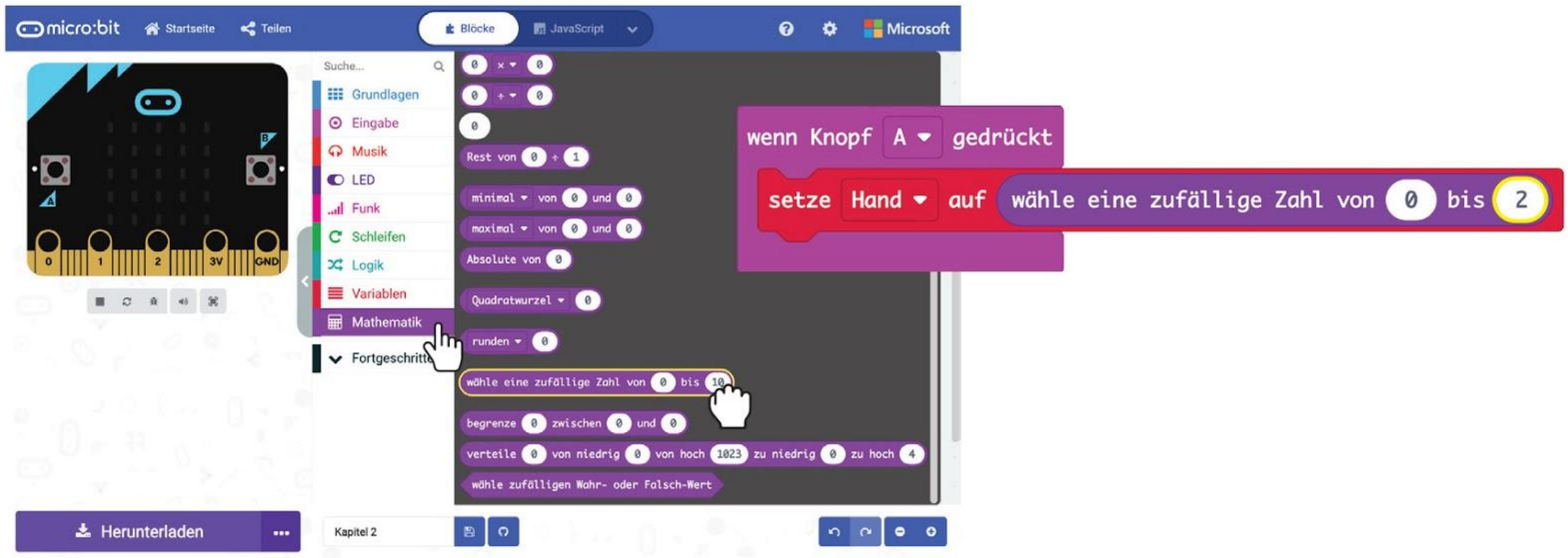




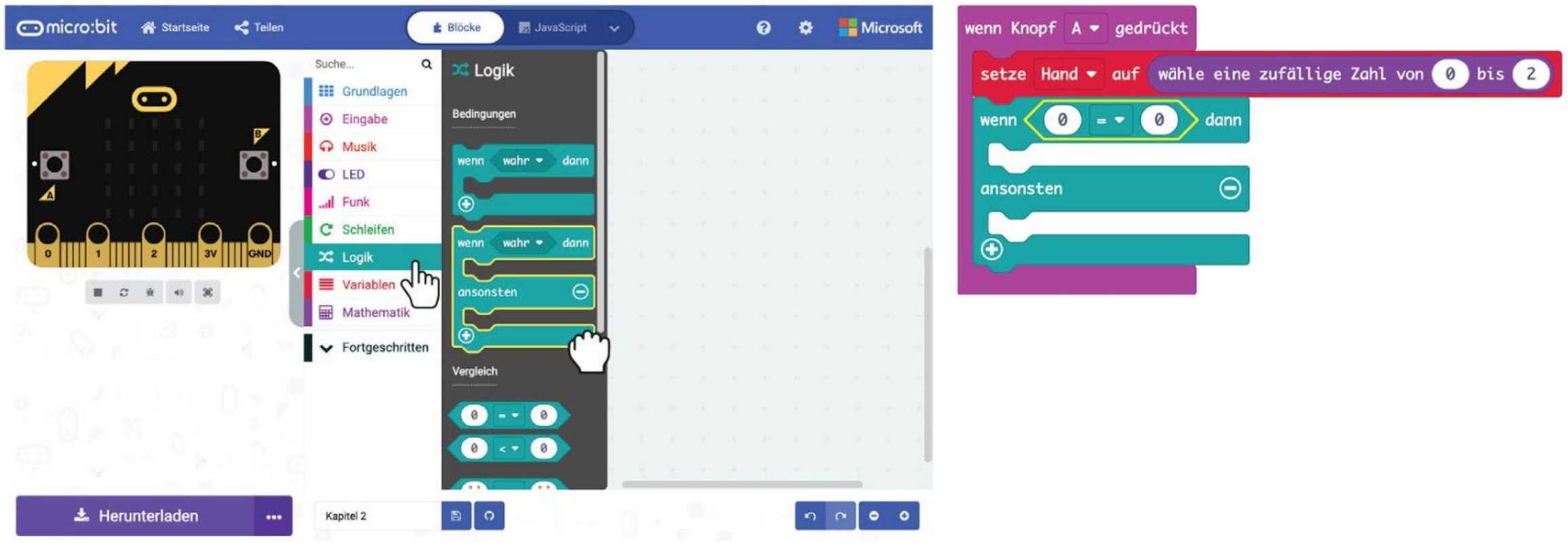
Schritt 3 Klicke auf die Gruppe **[Variablen]** und dann auf den Block **[setze _ auf _]**. Ziehe ihn in den Block **[wenn Knopf A gedrückt]**.



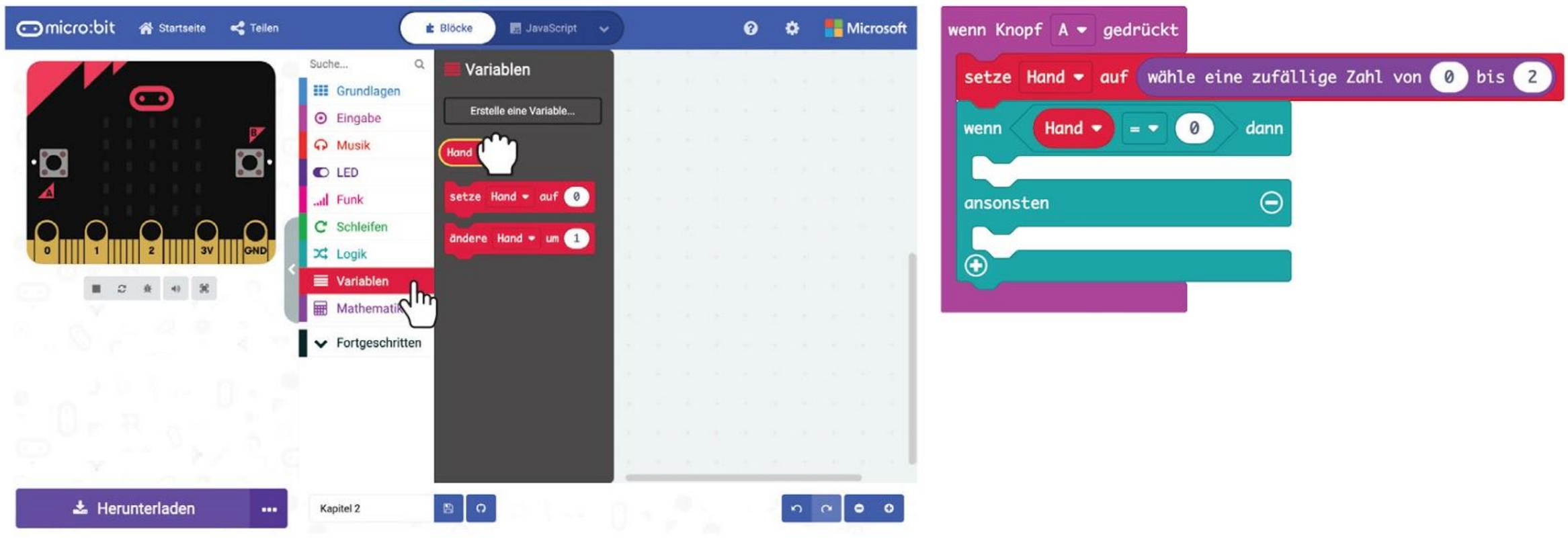
Schritt 4 Klicke auf die Gruppe **[Mathematik]** und wähle den Block **[wähle eine zufällige Zahl von _ bis _]**. Ändere die Zahl 10 auf den Wert **2**.



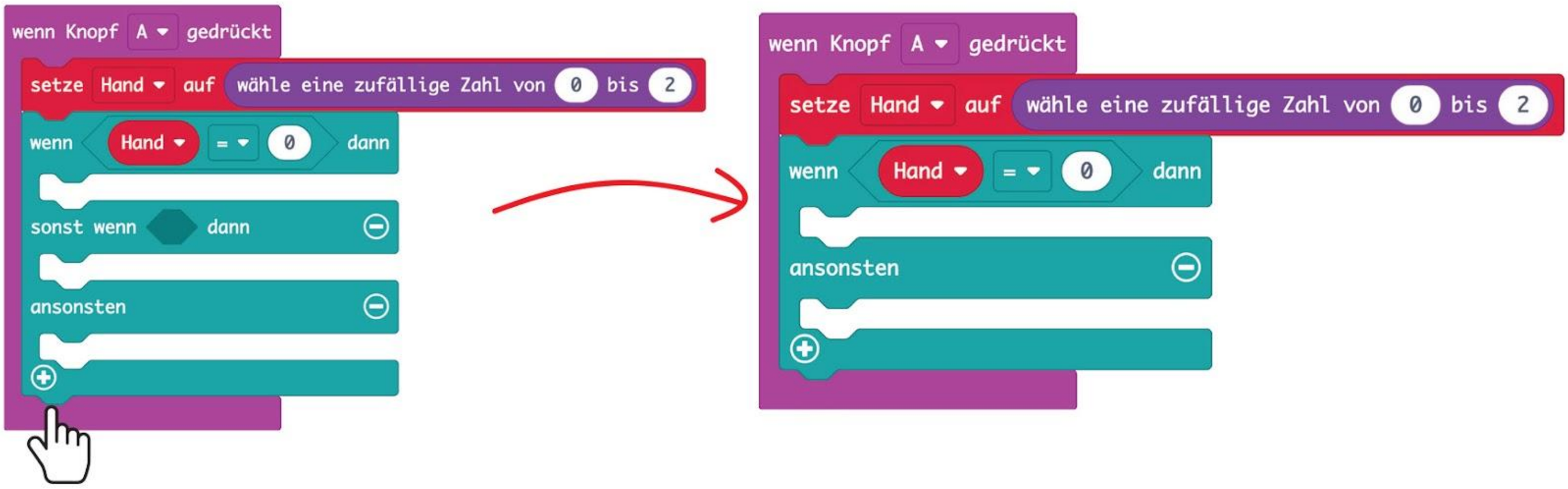
Schritt 5 Klicke auf die Gruppe **[Logik]** und dann auf den Block **[wenn-dann-ansonsten]**. Füge den Vergleichs-Block **[_ = _]** in den wenn-Block ein.



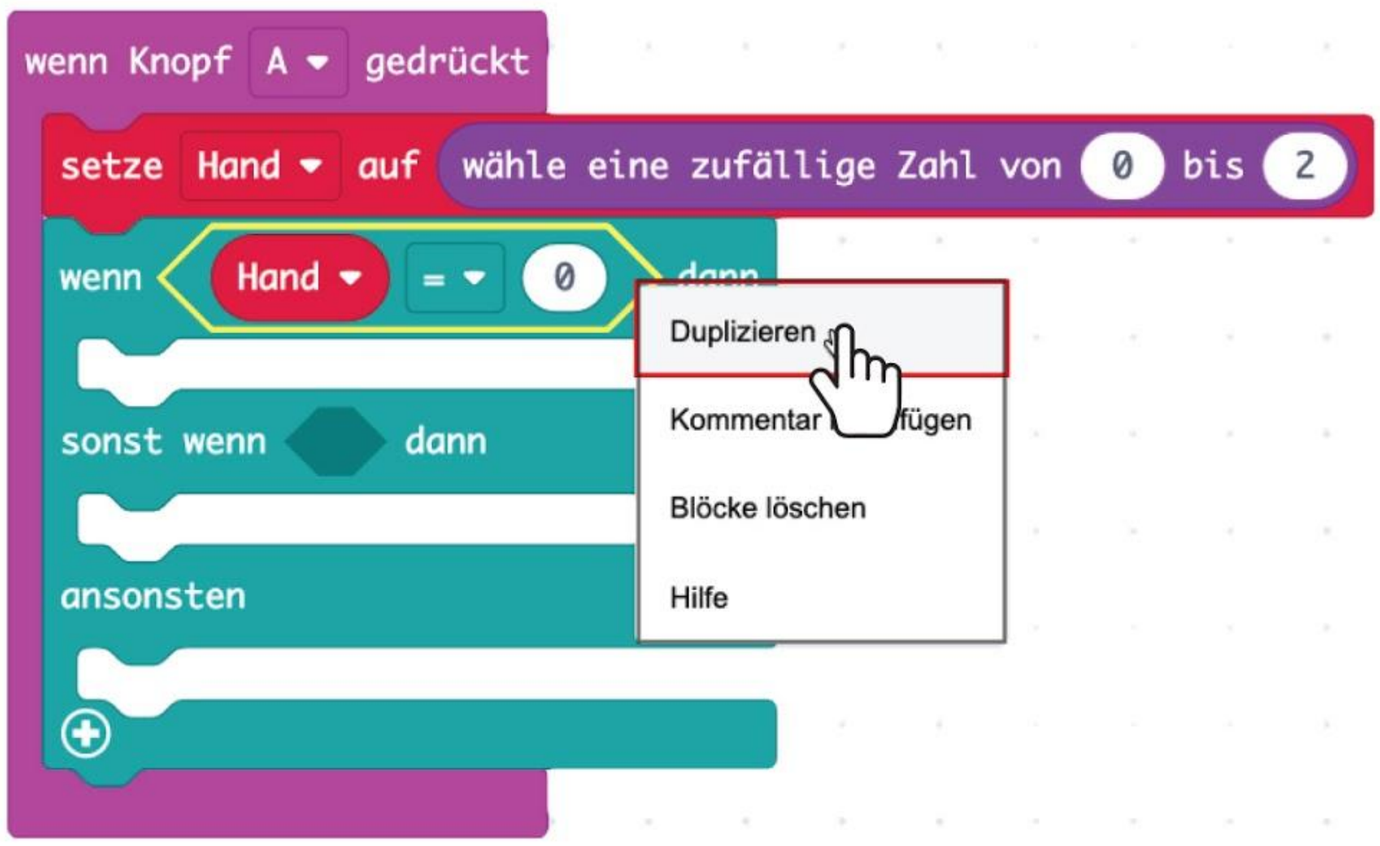
Schritt 6 Klicke auf die Gruppe **[Variablen]** und wähle **[Hand]** aus. Setze den Block in den Vergleichs-Block ein.



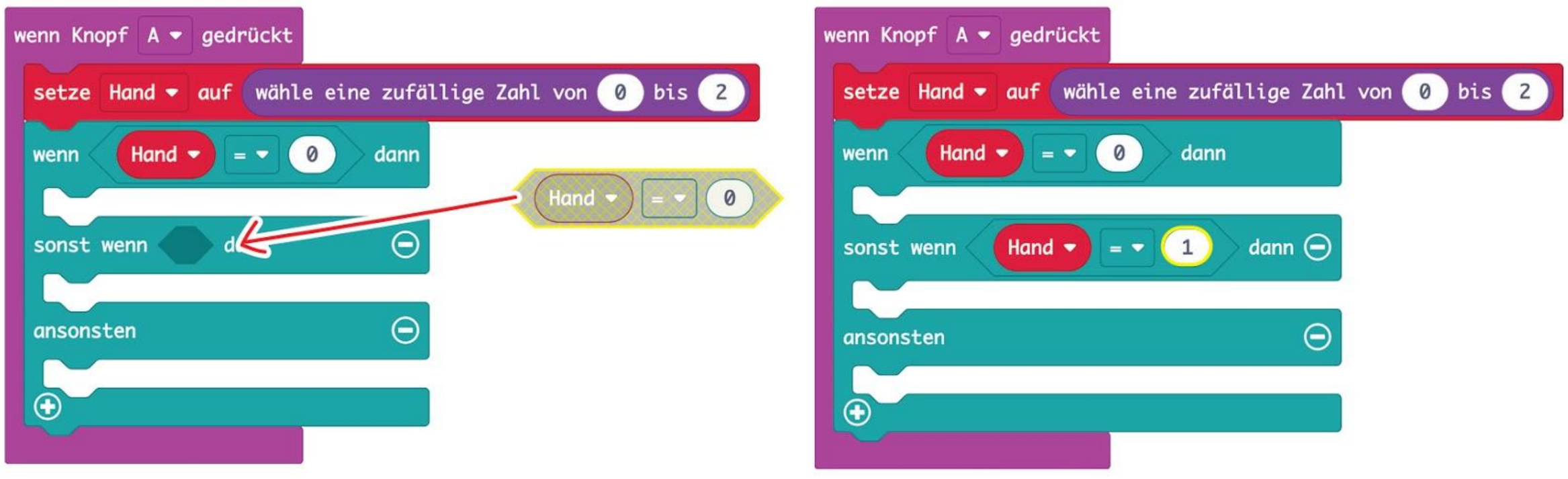
Schritt 7 Klicke auf das Symbol (**+**) um die Bedingung **[sonst wenn]** hinzuzufügen.



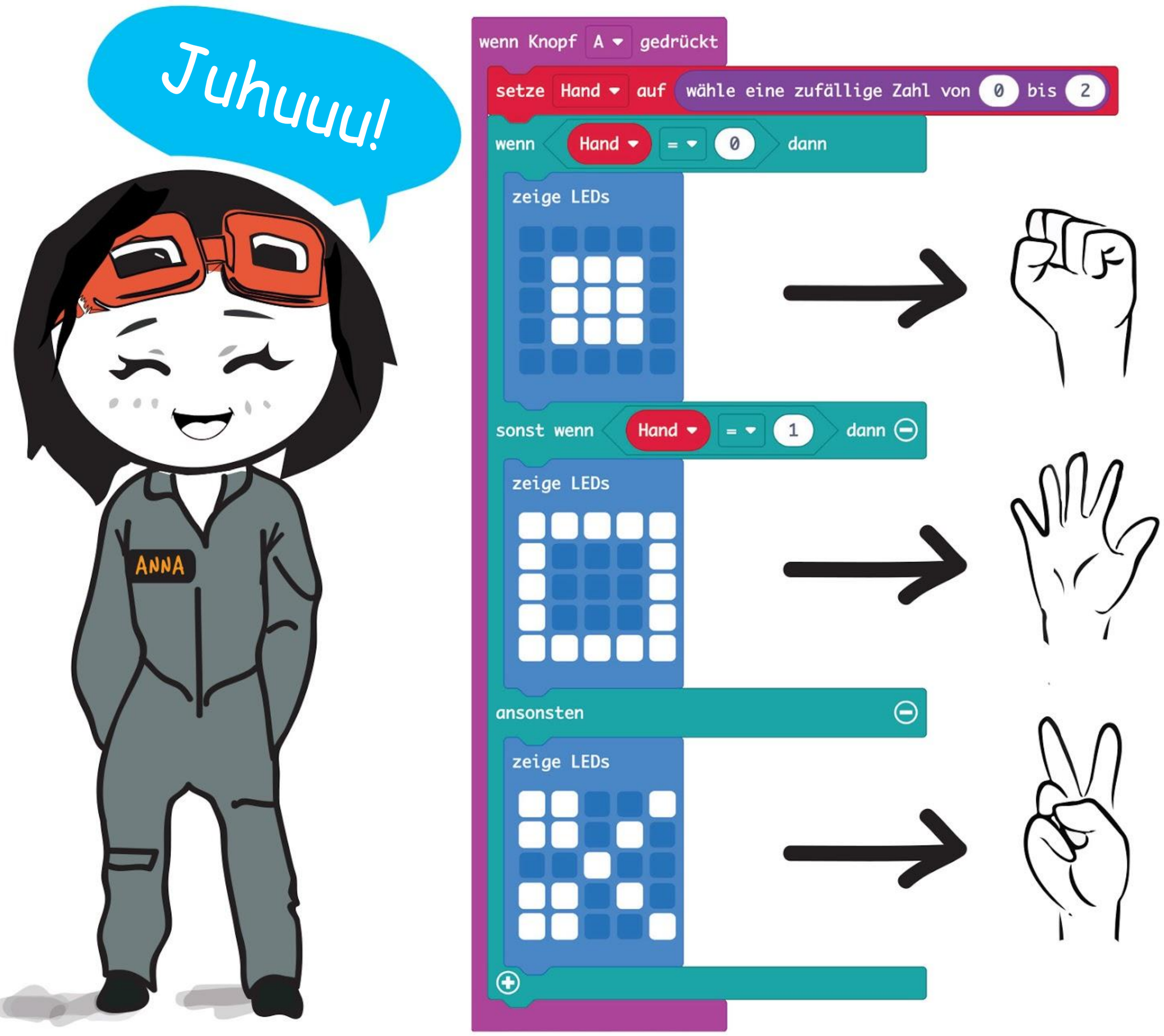
Schritt 8 Klicke mit der rechten Maustaste auf den Vergleichs-Block und auf '**Duplizieren**'



Schritt 9 Füge den neuen Block in den Bereich 'sonst wenn' ein und ändere die Zahl 0 auf **1**.



Schritt 10 Lege drei **[Grundlagen] : [zeige LEDs]**-Blöcke in die Bereiche ‘wenn’, ‘sonst wenn’ und ‘ansonsten’. Zeichne Symbole, indem du auf die Kästchen in den **[zeige LEDs]**-Blöcken klickst:

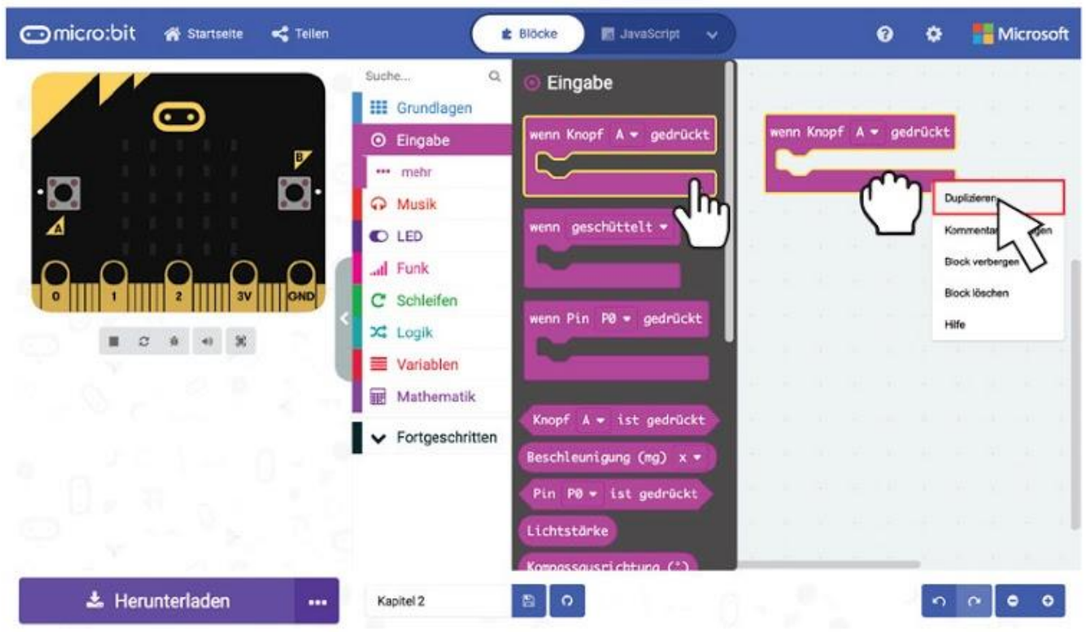


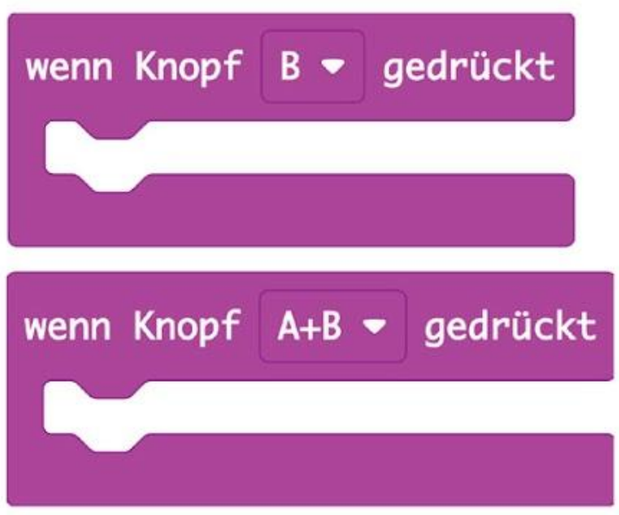
Lade den Code auf deinen EDU:BIT und spiele “Schere, Stein, Papier” mit deinen Freundinnen und Freunden. Immer wenn du Knopf A auf dem micro:bit oder den gelben Knopf drückst, zeigt die Anzeige zufällig Schere, Stein oder Papier an.

Merke: Wenn du dein Projekt behalten willst, speichere es in einem Ordner auf deinem Computer, indem du auf das Symbol “Speichern” klickst.

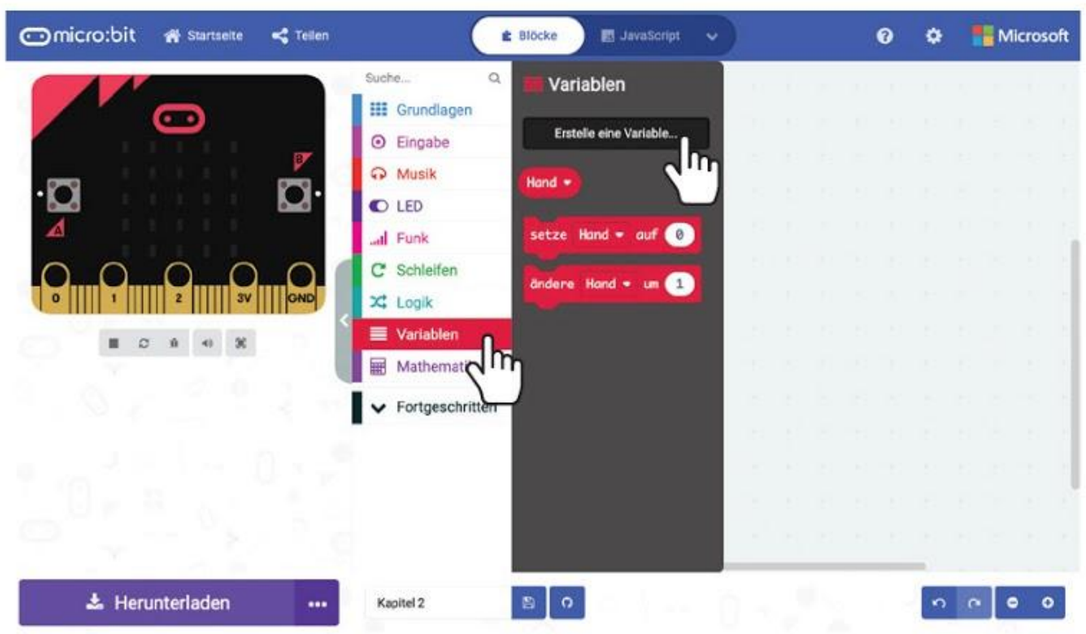
Lass uns dieses Programm erweitern, indem wir jeder Person, die mitspielt, 3 Leben geben. Dafür musst du eine Variable mit dem Namen "Leben" erstellen und die folgenden Code-Blöcke hinzufügen.

Schritt 11 Klicke auf die Gruppe **[Eingabe]** und wähle den Block **[wenn Knopf _ gedrückt]**. Dupliziere den Block und ändere die Einstellung jeweils zu '**Knopf B**' und '**Knopf A+B**'.

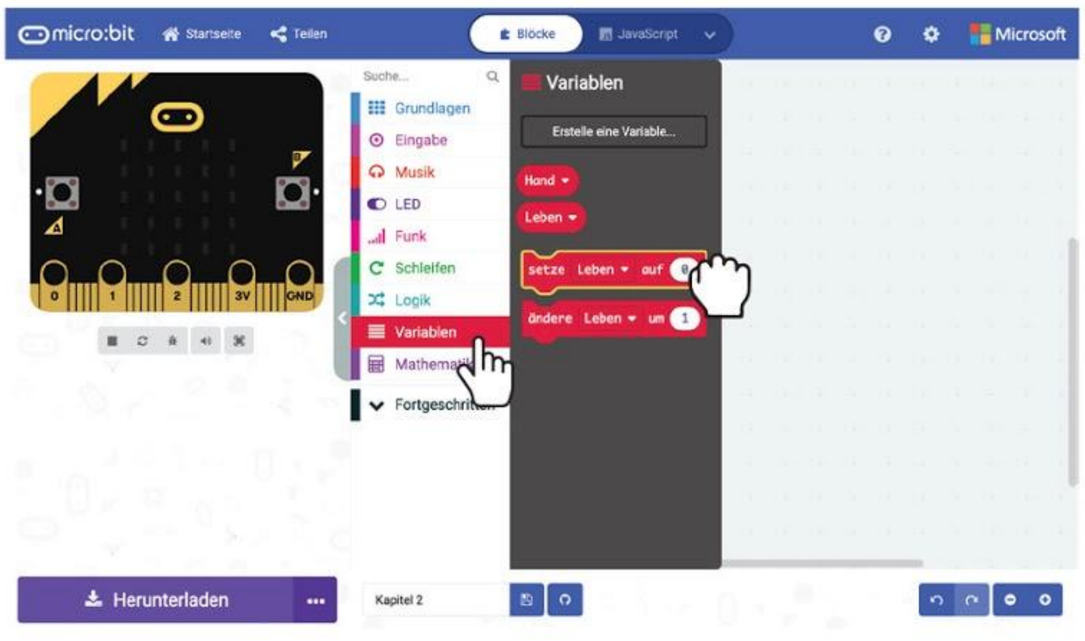




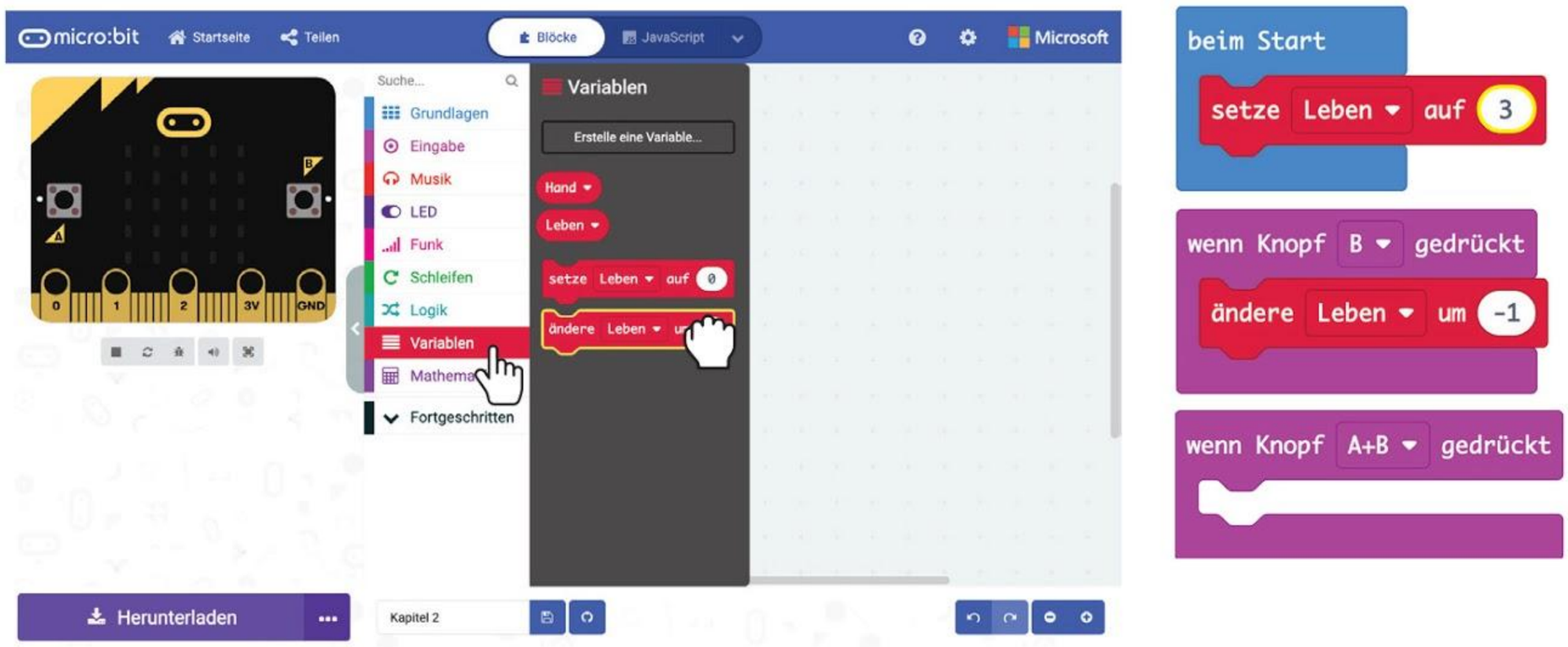
Schritt 12 Klicke auf die Gruppe **[Variablen]** und auf **[Erstelle eine Variable]**. Schreibe '**Leben**' in das Dialogfenster und klicke auf OK.



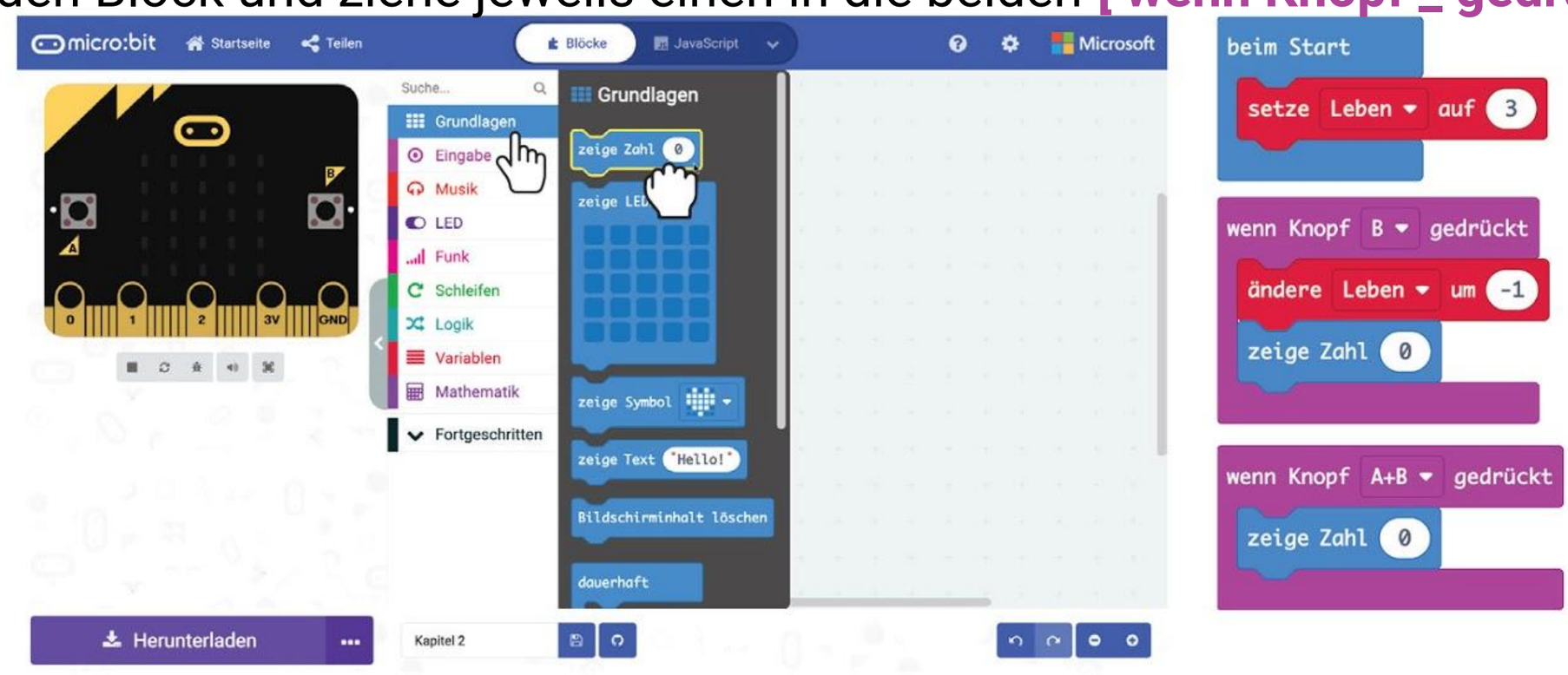
Schritt 13 Klicke auf **[Variablen]** und wähle den Block **[setze _ auf _]**. Ziehe diesen Block in den Block **[Grundlagen] : [beim Start]**. Ändere die Variable auf '**Leben**' und den Wert auf **3**.



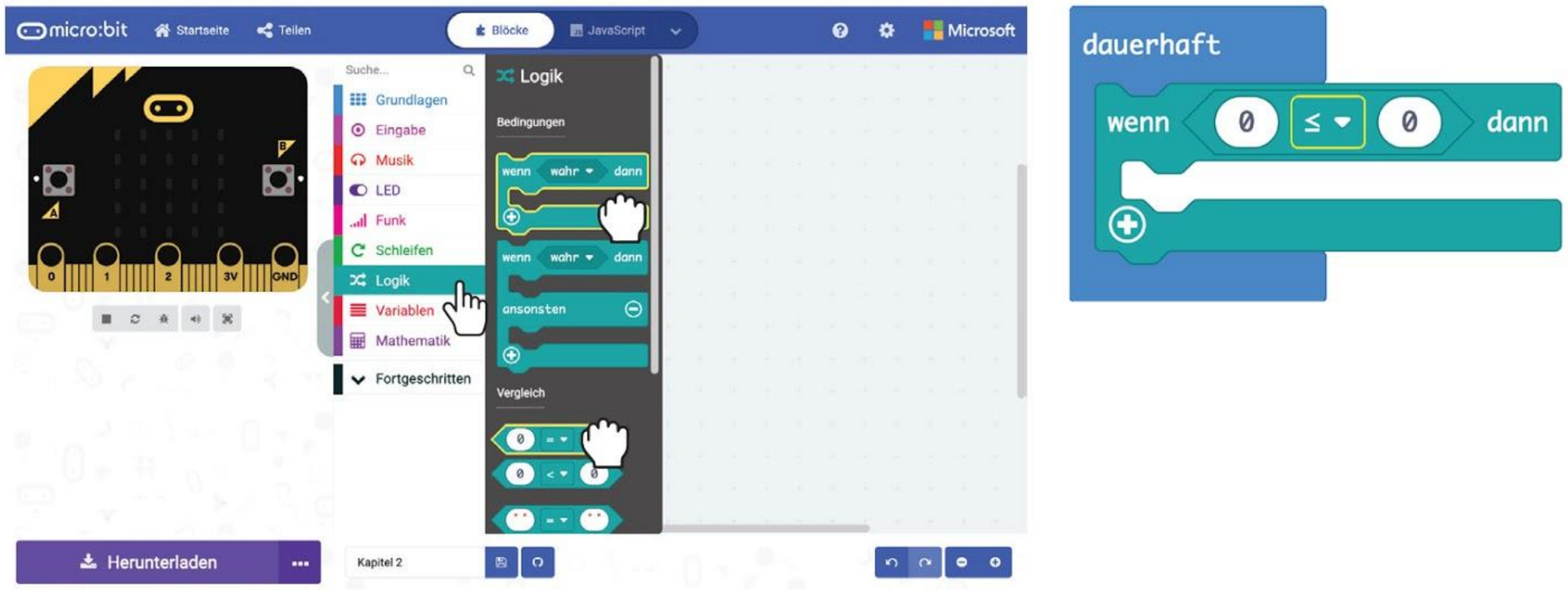
Schritt 14 Klicke wieder auf **[Variablen]** und dann auf den Block **[ändere _ um _]**. Ziehe den Block in **[wenn Knopf B gedrückt]**. Ändere die Variable zu '**Leben**' und den Wert zu **-1**.



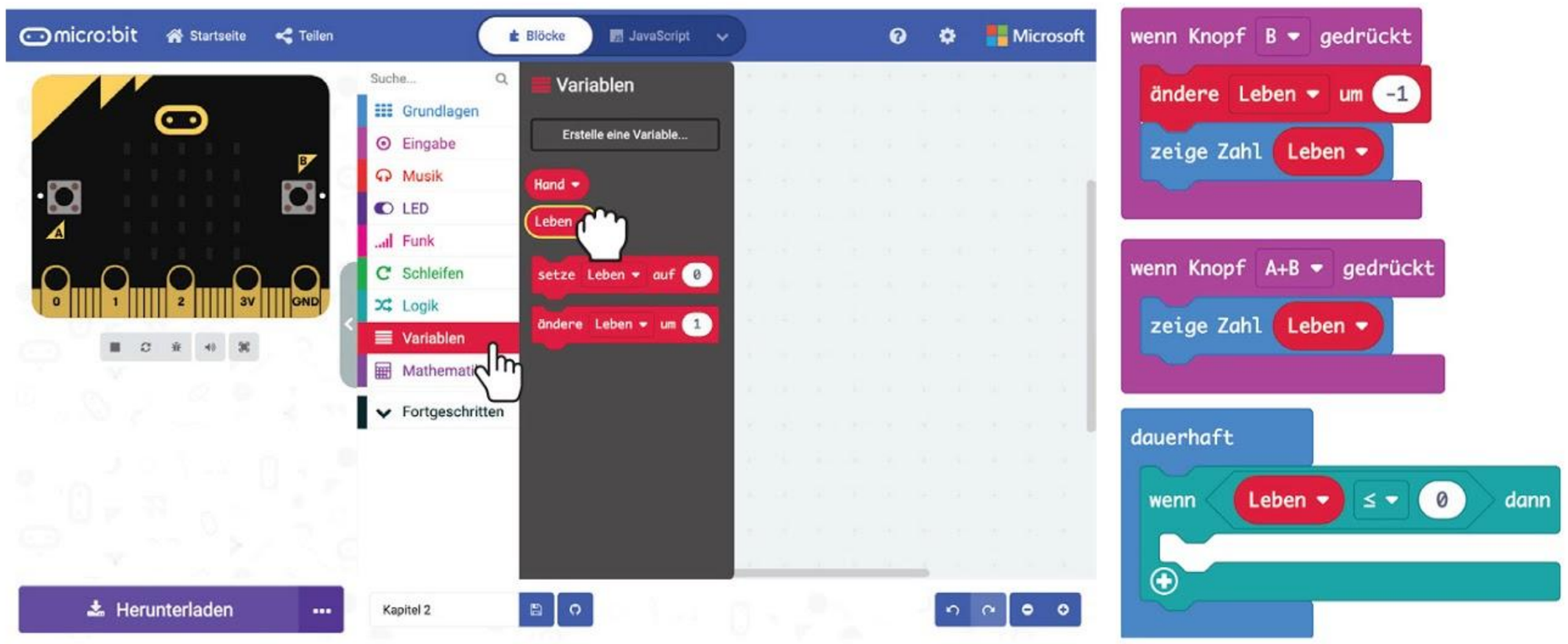
Schritt 15 Klicke auf die Gruppe **[Grundlagen]** und wähle **[zeige Zahl]**. Dupliziere den Block und ziehe jeweils einen in die beiden **[wenn Knopf _ gedrückt]**-Blöcke.



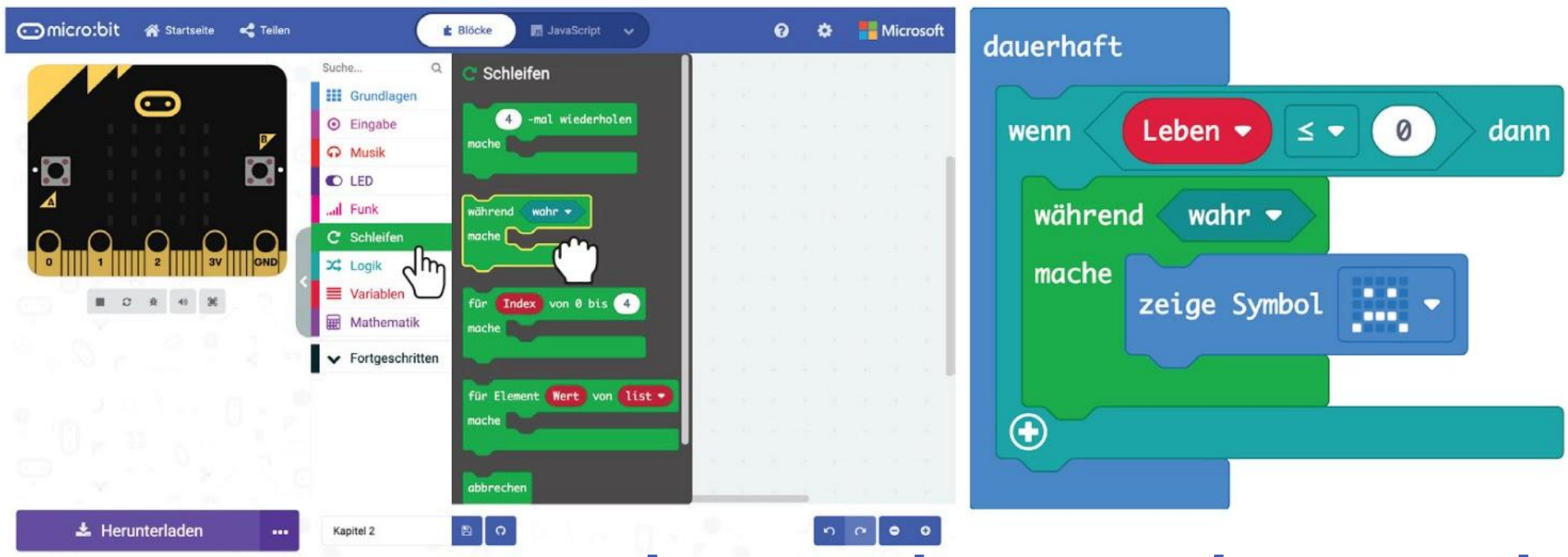
Schritt 16 Klicke auf die Gruppe **[Logik]**, ziehe einen **[wenn-dann]**-Block und einen Block **[_ = _]** in den Block **[dauerhaft]** und ändere das Symbol zu '≤'.



Schritt 17 Klicke auf die Gruppe **[Variablen]** und den Block **[Leben]**. Dupliziere ihn und ziehe ihn in den Block **[zeige Zahl]** und die linke Seite des Blockes **[_ = _]**.

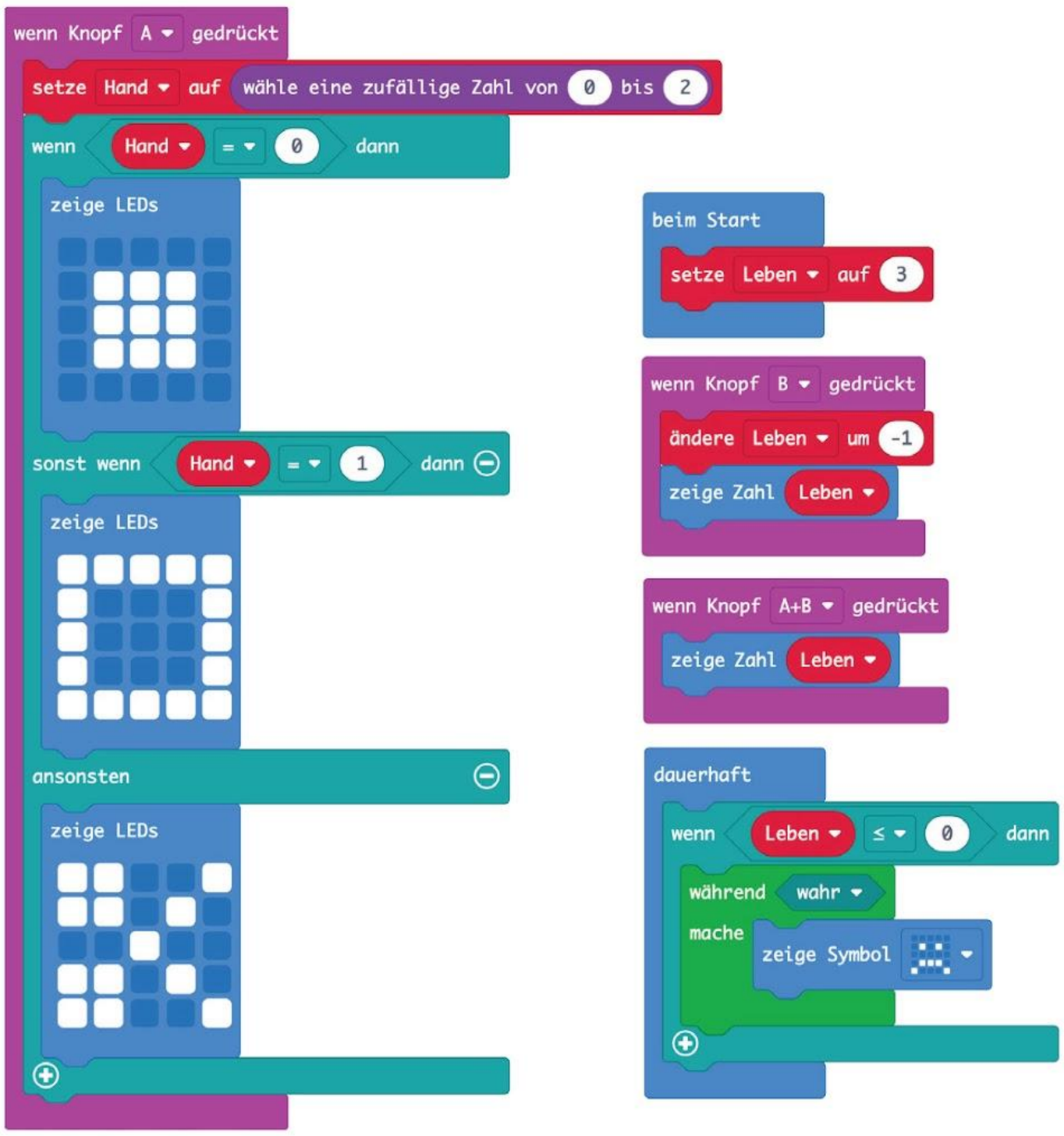


Schritt 18 Klicke auf die Gruppe **[Schleifen]** und wähle den Block **[während _ mache]**. Lege ihn in den **[wenn-dann]**-Block.

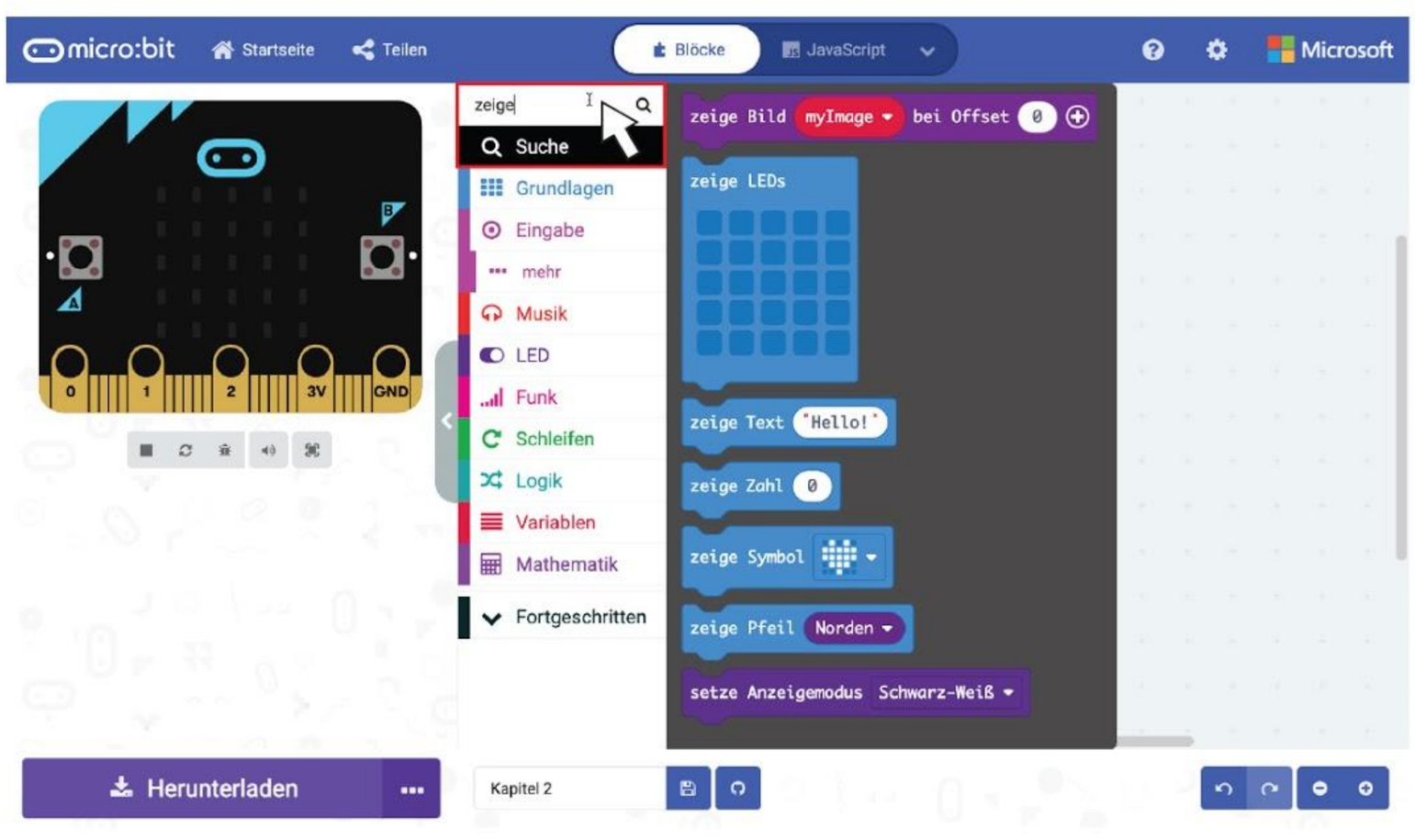


Schritt 19 Klicke auf die Gruppe **[Grundlagen]** und dann auf **[zeige Symbol]**. Ziehe den Block in den Block **[während _ mache]** und ändere das Symbol zu **‘traurig’**.

Schritt 20 Hier ist der vollständige Code. Übertrage ihn auf deinen EDU:BIT und spiele mit deinen Freundinnen und Freunden. Wer gewinnt bei “Schere, Stein, Papier”?

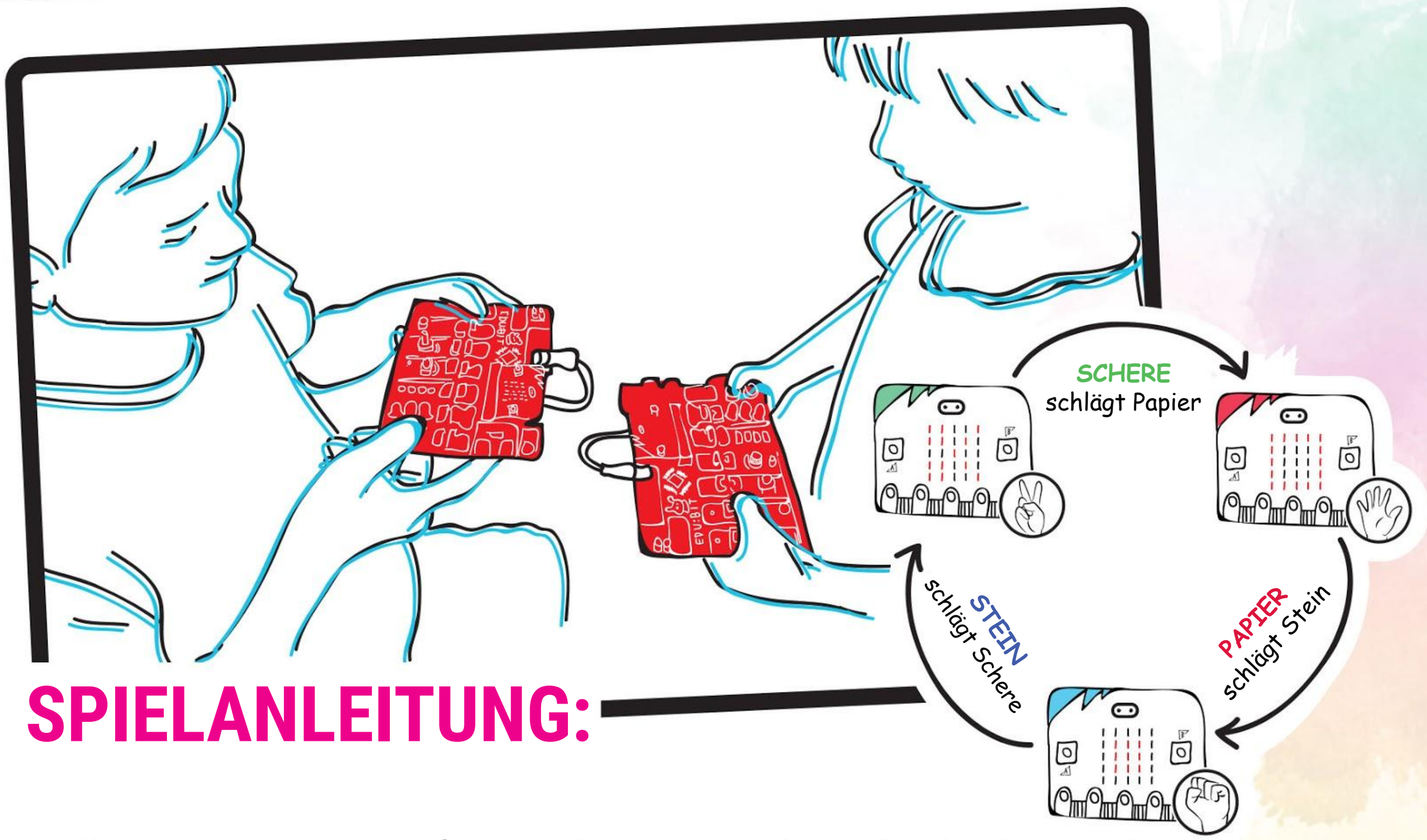


Hier ist ein Tipp für dich…
Alle Blöcke einer Gruppe haben die gleiche Farbe, um sie leichter zuordnen zu können.
Du kannst sie auch über die Suchfunktion finden.



Spielen wir!

Schere, Stein, Papier - Erweiterte Version



SPIELANLEITUNG:

- Stellt euch gegenüber auf. Wenn ihr bereit seid, drückt gleichzeitig den gelben Knopf (Knopf A) um zufällig Schere, Stein oder Papier anzuzeigen.
- Vergleicht das Ergebnis und entscheidet, wer gewonnen hat.
- Wer verloren hat, drückt den blauen Knopf (Knopf B) auf seinem EDU:BIT um ein Leben abzuziehen.
- Drückt beide Knöpfe gleichzeitig um die Anzahl der verbleibenden Leben anzuzeigen.
- Wenn jemand dreimal verloren hat, heißt es “Game Over” und der EDU:BIT zeigt ein trauriges Gesicht an.

MERKE!

- Um eine neue Runde zu spielen, musst du einen RESET durchführen.
- Wenn gerade niemand da ist, mit dem du spielen kannst, spiele gegen den Simulator im MakeCode Editor.

KNACKE DEN CODE

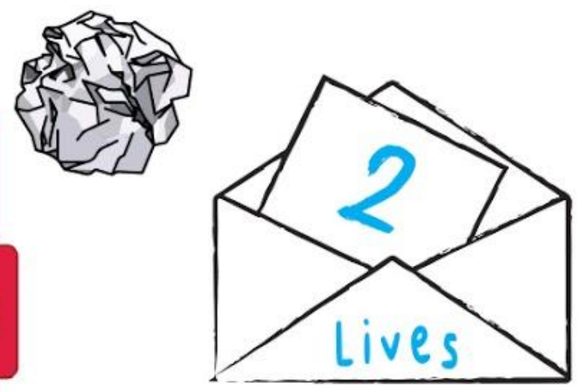
Beim Programmieren benutzen wir **Variablen** um Informationen oder Zahlen zu speichern, die während der Laufzeit (wenn das Programm abläuft) geändert werden können. Du kannst dir eine Variable wie ein beschriftetes Kuvert vorstellen, in dem ein Zettel mit einer Information steckt. Man kann den Zettel herausnehmen und einen anderen Zettel mit einer neuen Information hineingeben.

In unserem Code haben wir eine Variable mit dem Namen “Leben” erstellt und ihr den Startwert 3 gegeben.

Die Zahl 3 auf dem Zettel ist die Information.

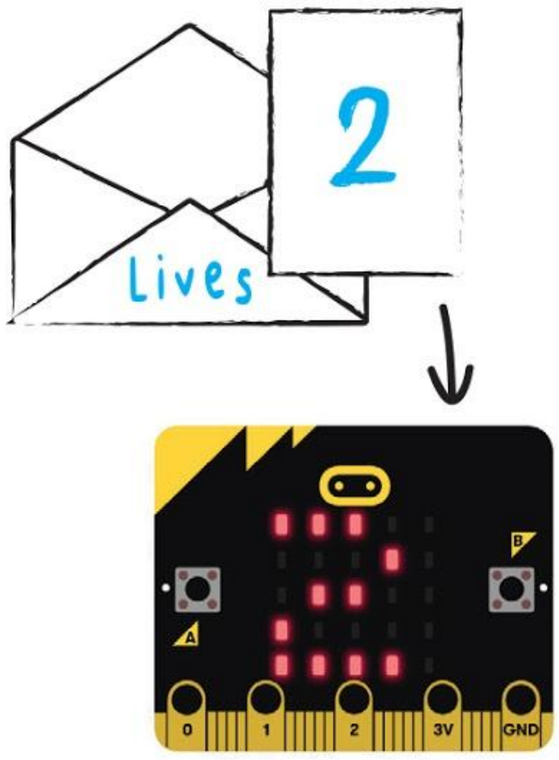
Das Kuvert mit dem Namen “Leben” ist unsere Variable.

Immer wenn Knopf B gedrückt wird, ändern das Programm den Wert um -1.



Wenn Knopf B gedrückt wird, wird der Zettel mit der Zahl “3” entfernt und durch einen neuen Zettel mit der Zahl “2” (3 - 1 = 2) ersetzt.

Wenn die Knöpfe A+B gleichzeitig gedrückt werden, zeigt die LED-Matrix den aktuellen Wert der Variable “Leben” an.



Wenn die Knöpfe A+B gleichzeitig gedrückt werden, nehmen wir den Zettel kurz aus dem Kuvert heraus und “lesen” die Information darauf.

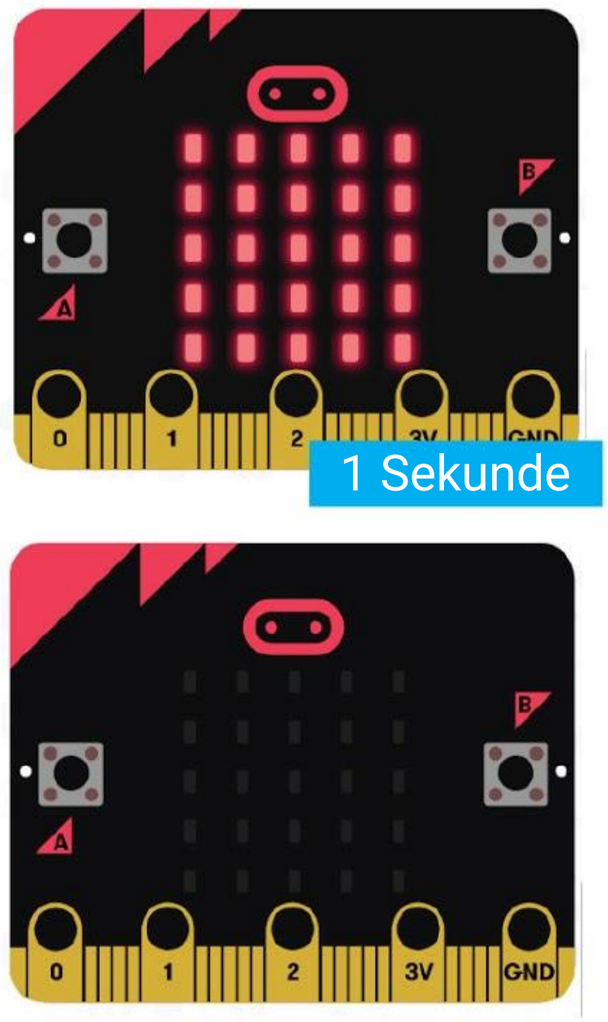
ENTDECKE NOCH MEHR

Neben **[wenn Knopf _ gedrückt]** kannst du auch andere Blöcke aus der Gruppe **[Eingabe]** für ereignisbasierte Programme verwenden. Ereignisse, die ein Benutzer auslöst, wie das Drücken eines Knopfes oder das Schütteln der Platine, werden auch “Events” genannt.

Der folgende Code lässt zum Beispiel die LED-Matrix aufleuchten, wenn der micro:bit geschüttelt wird. Probier es mal aus~

Ergebnis

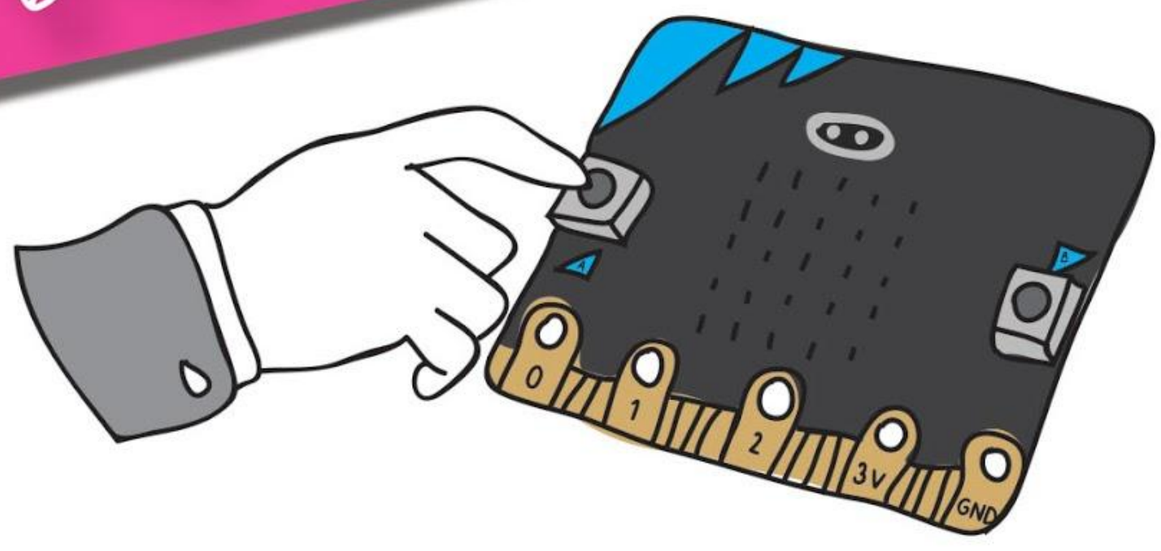
Event



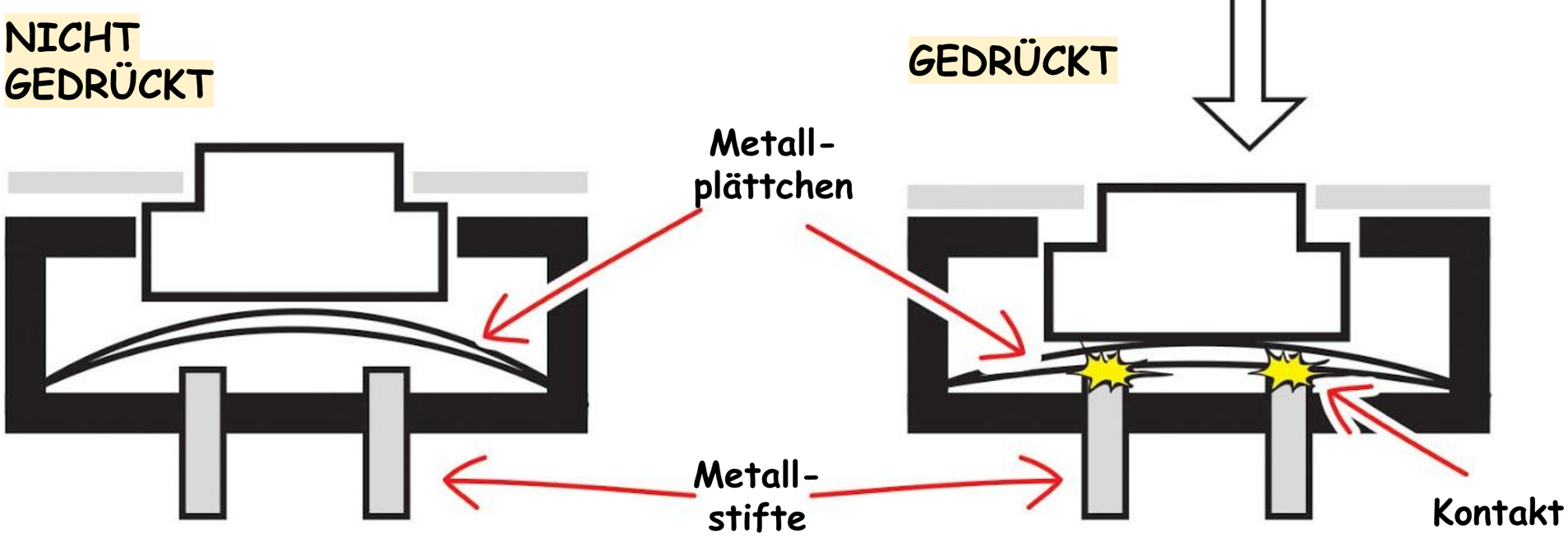
Wenn du im Block auf **[geschüttelt]** klickst, erscheint ein Dialogfenster, in dem du viele andere Auslöser (Trigger) wählen kannst. Programmiere einmal deinen EDU:BIT so, dass er für jeden Auslöser ein anderes Symbol anzeigt. Viel Spaß!

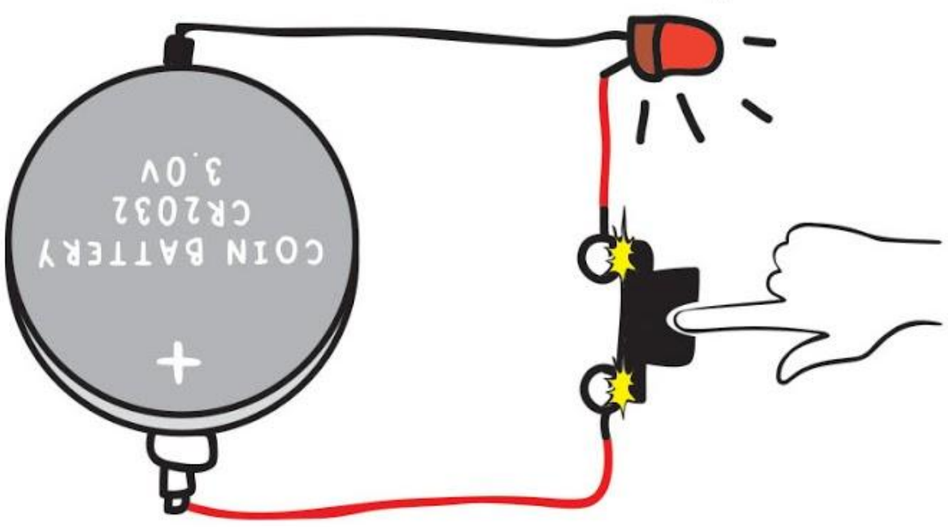
EDU:BIT kann das Schütteln und seine eigene Lage erkennen, weil im micro:bit ein Bewegungssensor eingebaut ist.

GUT ZU WISSEN!



Ein **Knopf** oder **Taster** ist ein Eingabegerät mit zwei möglichen Zuständen - gedrückt oder nicht gedrückt.

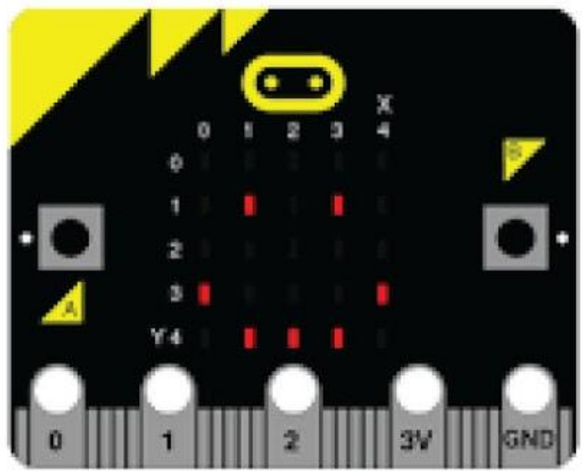




Wenn du den Knopf drückst, fließt der Strom im Stromkreis und die LED leuchtet! Rate mal, was passiert, wenn du den Knopf loslässt?

Schwarz, grau, grün und weiß werden normalerweise für EIN/AUS verwendet. ROT benutzt man für Panikknöpfe oder Not-Aus-Schalter von Maschinen.

HERAUSFORDERUNG

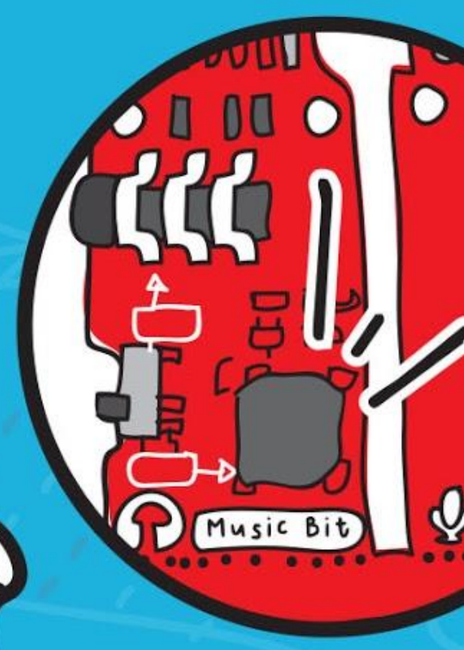


Programmiere mit EDU:BIT einen Zähler für deine Mitschülerinnen und Mitschüler. Wenn ein Mädchen in die Klasse kommt, muss es Knopf A drücken; wenn ein Junge die Klasse betritt, muss er Knopf B drücken.	
beim Start	Zeige einen Smiley. Setze die Variablen Mädchen = 0 und Jungen = 0
wenn Knopf A gedrückt (gelber Knopf)	Erhöhe die Variable Mädchen um 1
wenn Knopf B gedrückt (blauer Knopf)	Erhöhe die Variable Jungen um 1
wenn Knöpfe A+B gedrückt	Scrolle die folgenden Informationen über den LED-Bildschirm: Gesamt = (Mädchen + Jungen) Mädchen = (Mädchen) Jungen = (Jungen)

KAPITEL 3

Lass uns Musik machen~

Musik Bit (Piezo-Summer + Kopfhöreranschluss)

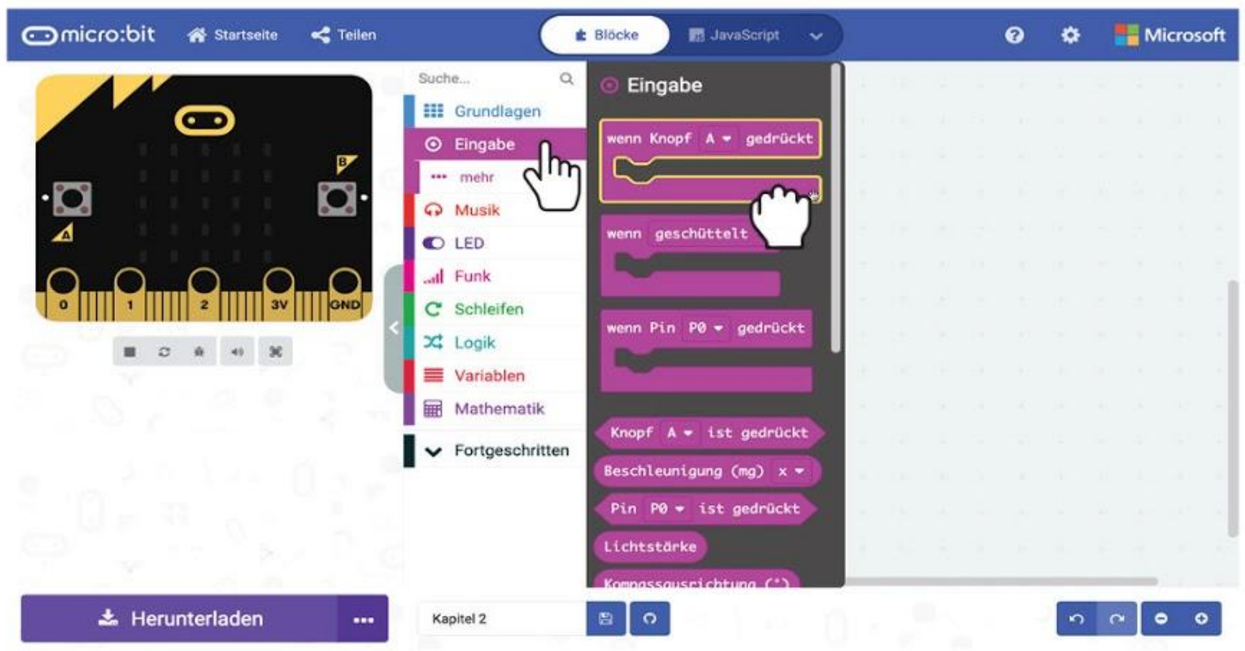


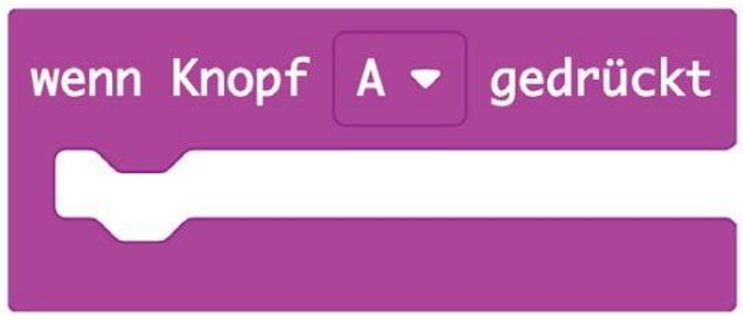
Scanne mich !

link.cytron.io/edubit-chapter-3

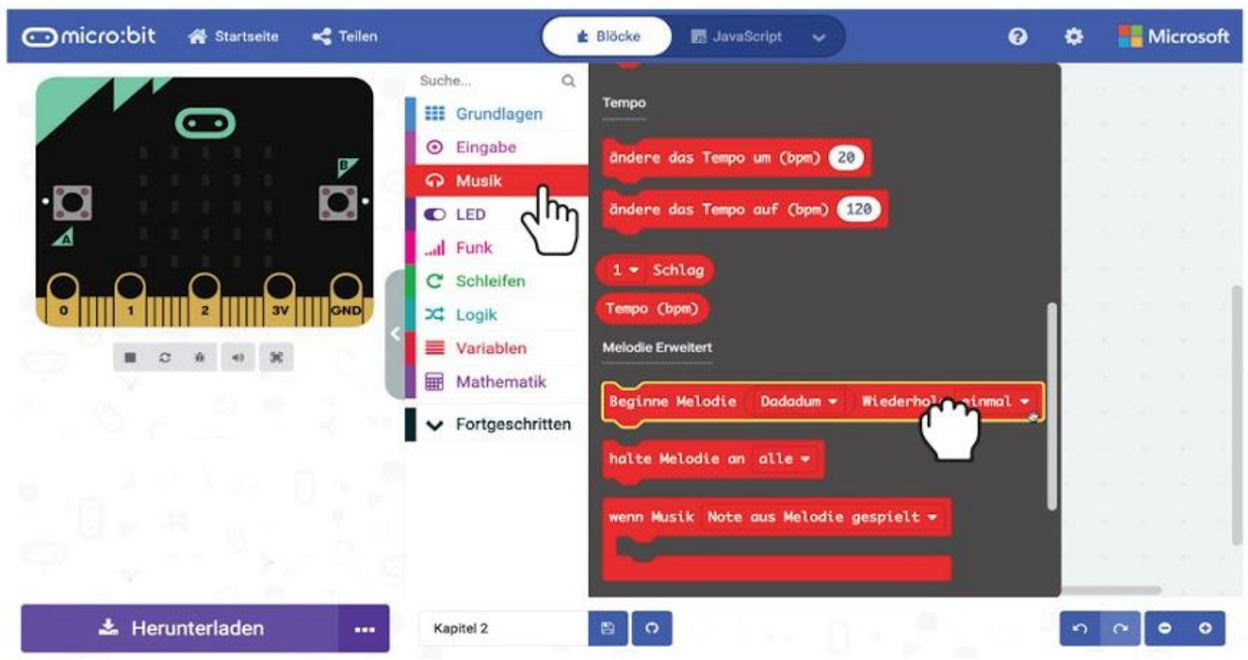
LASS UNS PROGRAMMIEREN!

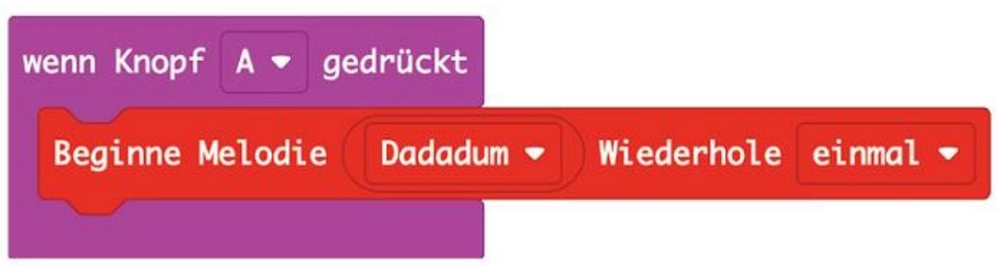
Schritt 1 Erstelle ein neues Projekt im MakeCode Editor. Klicke auf **[Eingabe]** und auf den Block **[wenn Knopf _ gedrückt]**.



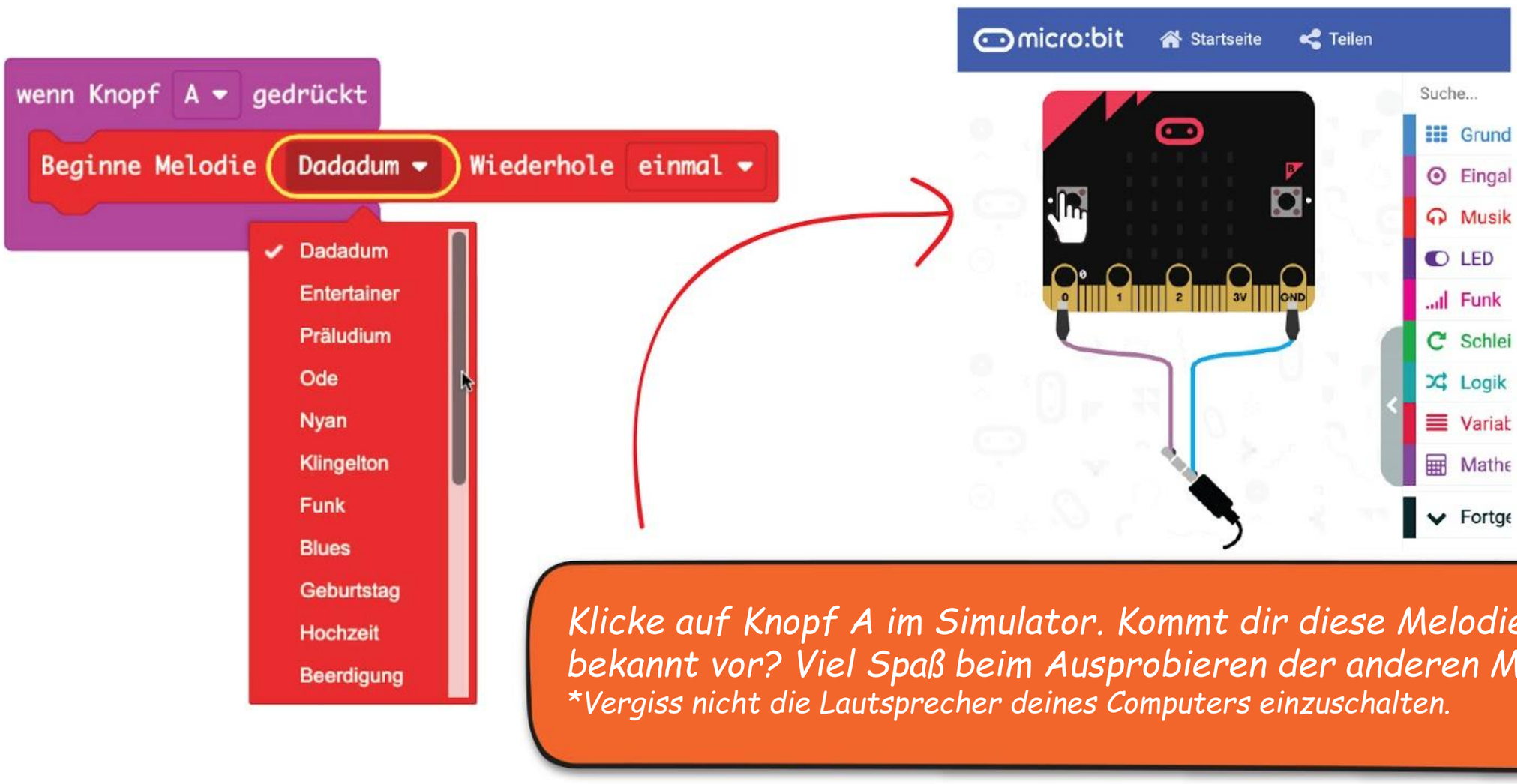


Schritt 2 Klicke auf die Gruppe **[Musik]** und wähle den Block **[Beginne Melodie _ Wiederhole _]**.





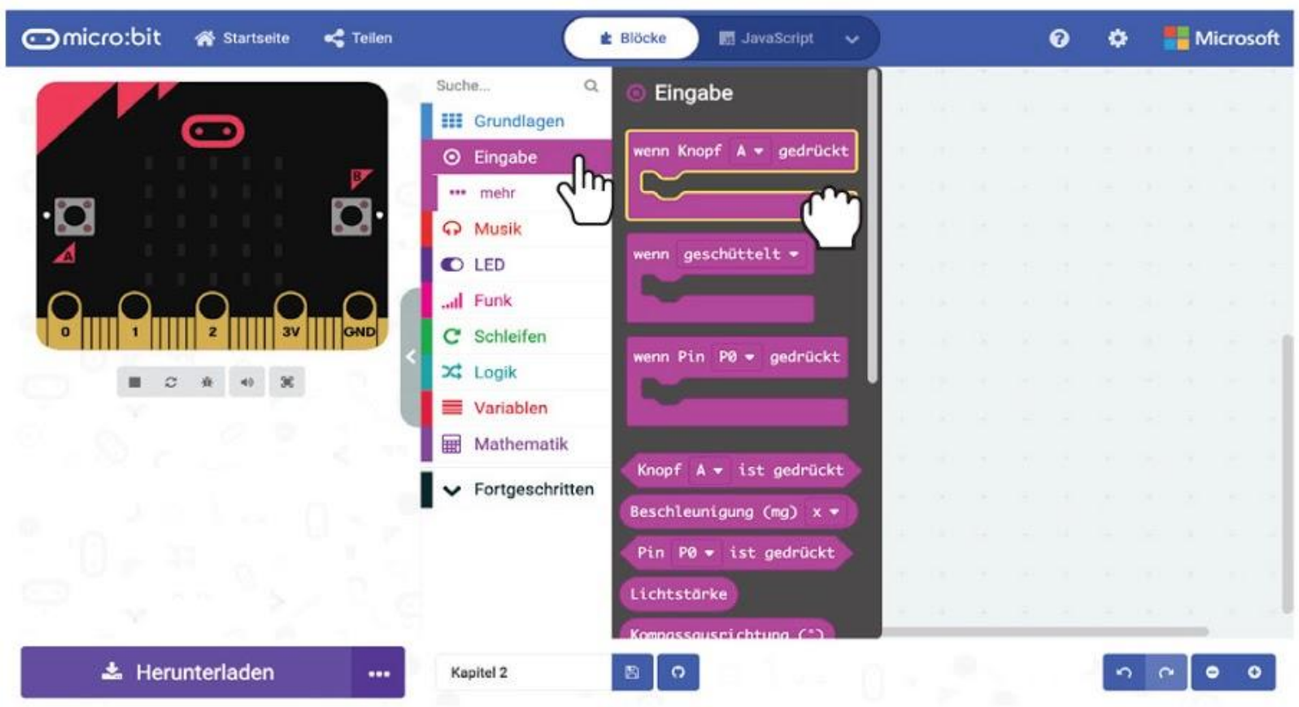
Schritt 3 Klicke auf **[Dadadum]** und wähle die Melodie **'Geburtstag'** aus der Klappliste.



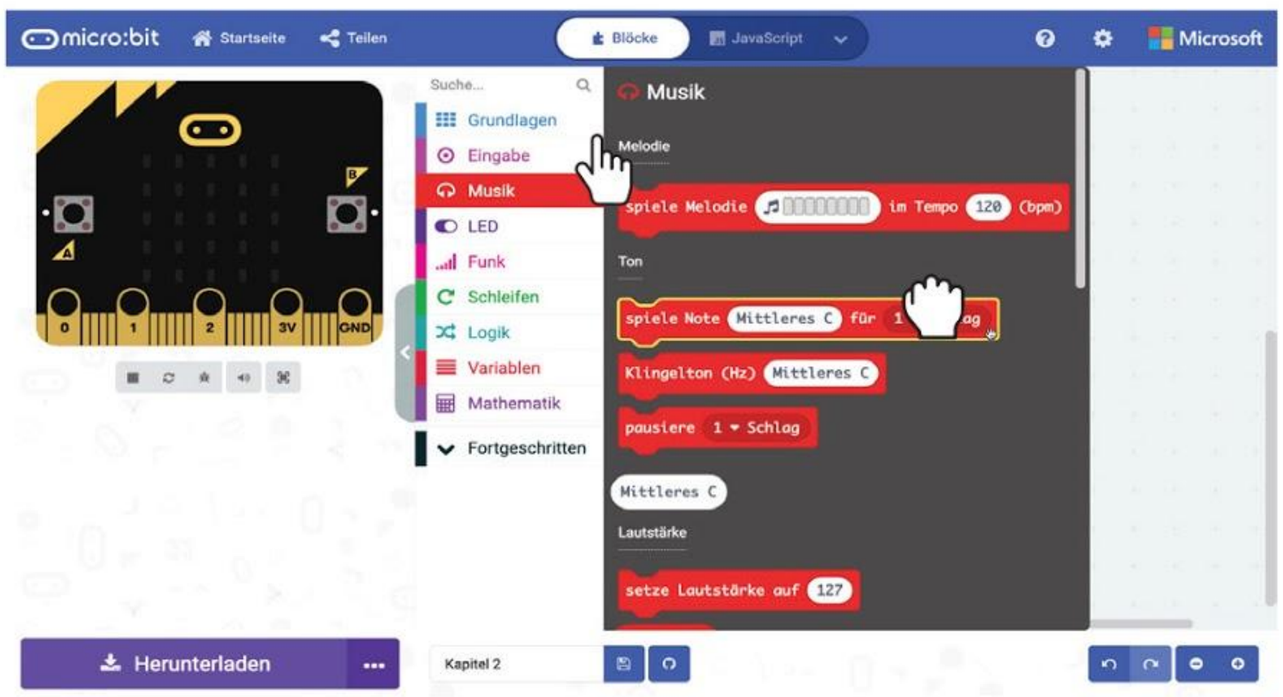
Klicke auf Knopf A im Simulator. Kommt dir diese Melodie bekannt vor? Viel Spaß beim Ausprobieren der anderen Melodien~
**Vergiss nicht die Lautsprecher deines Computers einzuschalten.*

Neben den voreingestellten Melodien kannst du EDU:BIT so programmieren, dass er alle Lieder spielt, die du magst. Probieren wir mal diesen Ohrwurm~

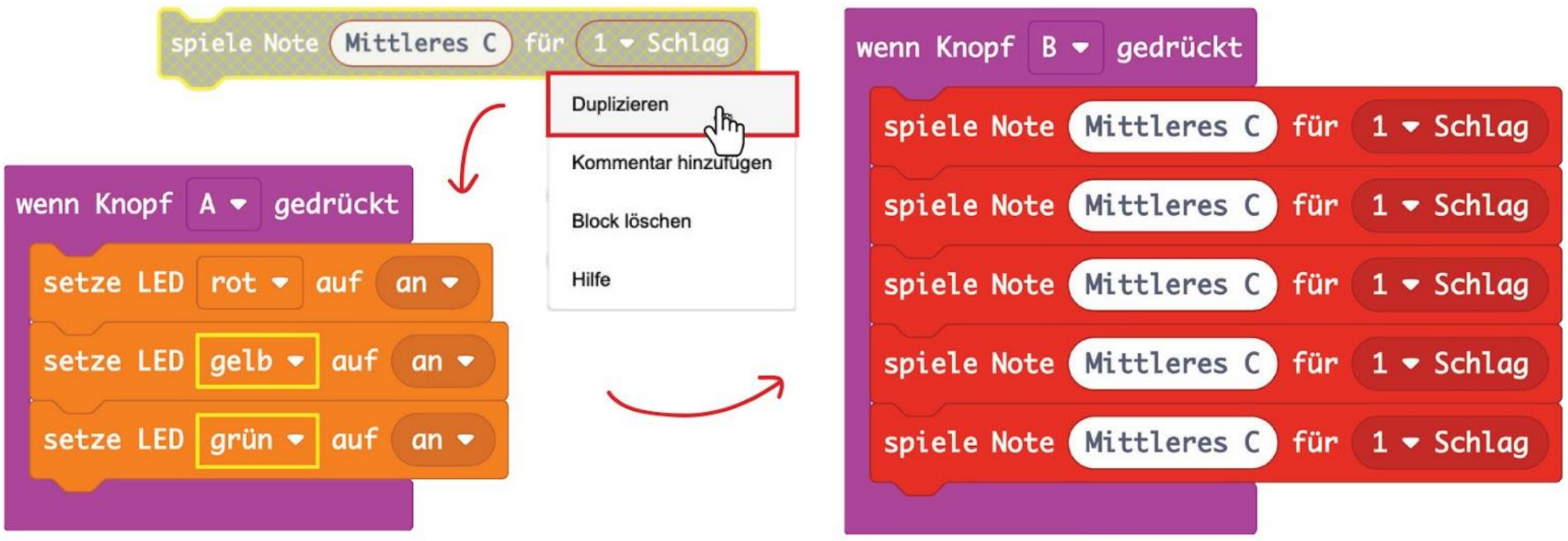
Schritt 4 Klicke auf **[Eingabe]** und wähle den **[wenn Knopf _ gedrückt]**-Block aus. Wähle Knopf **“B”**.



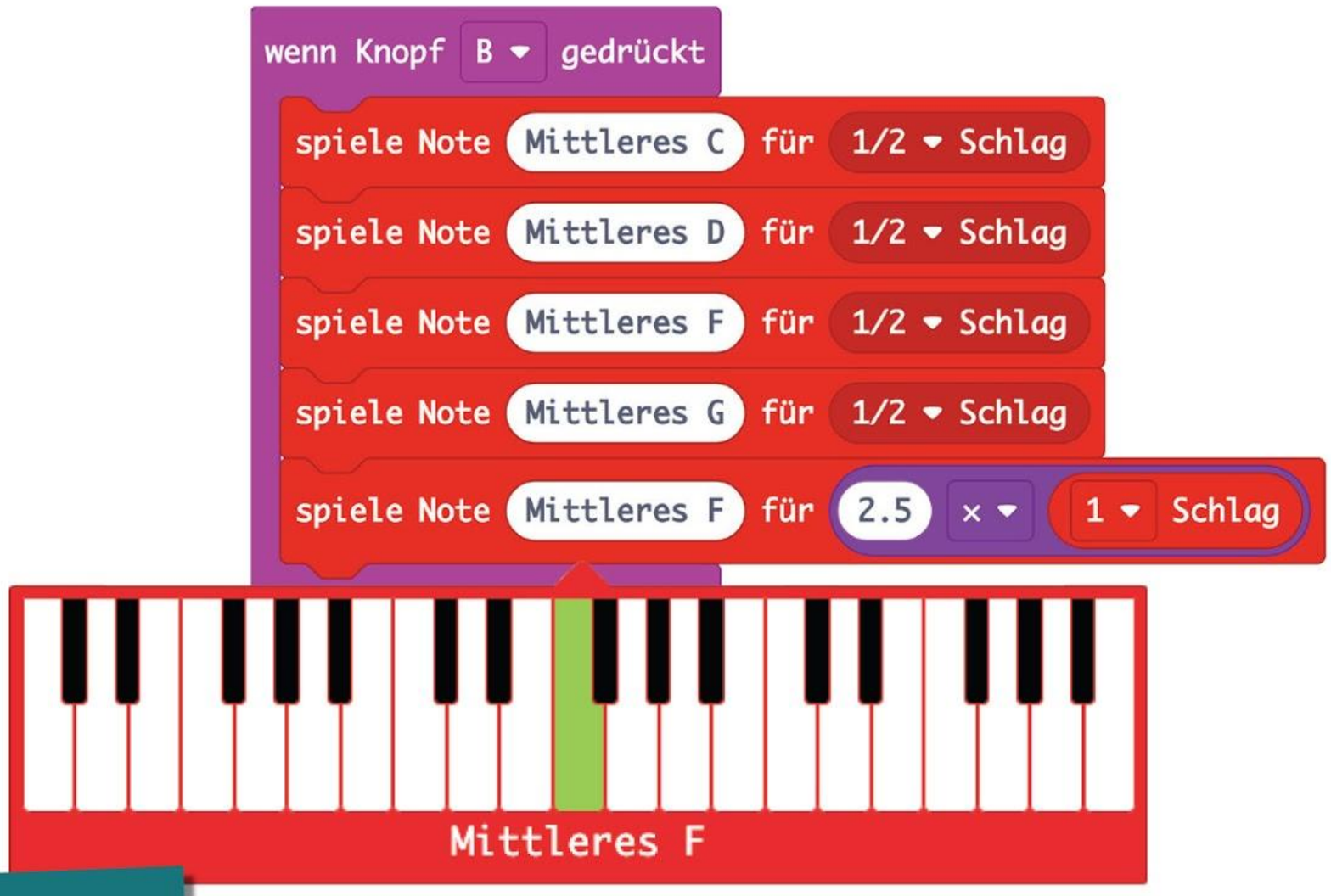
Schritt 5 Klicke in der Gruppe **[Musik]** auf den Block **[spiele Note _ für _]**.



Schritt 6 Klicke im Arbeitsbereich mit der rechten Maustaste auf den Block **[spiele Note _ für _]** und dann auf ‘Duplizieren’. Wiederhole, bis du fünf **[spiele Note _ für _]**-Blöcke hast. Lege die Blöcke in **[wenn Knopf B gedrückt]**.

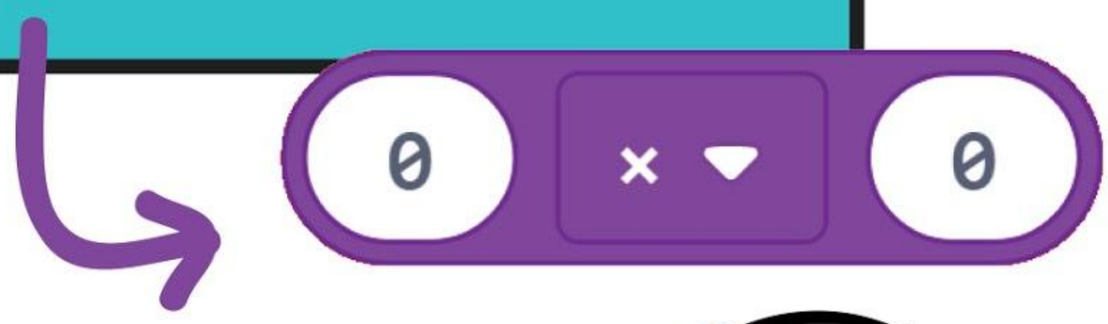


Schritt 7 Stelle **‘Note’** und **‘Schlag’** in den Blöcken **[spiele Note _ für _]** so ein wie in diesem Beispiel:



HINWEIS!

Den violetten Block findest du in der Gruppe [Mathematik].

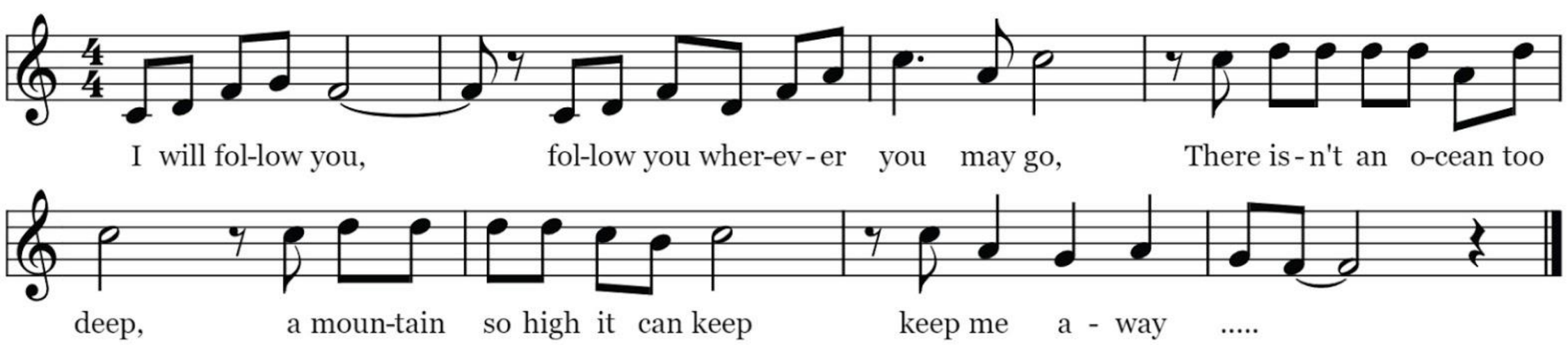


Klicke auf Knopf B im Simulator.
Kennst du dieses Lied?

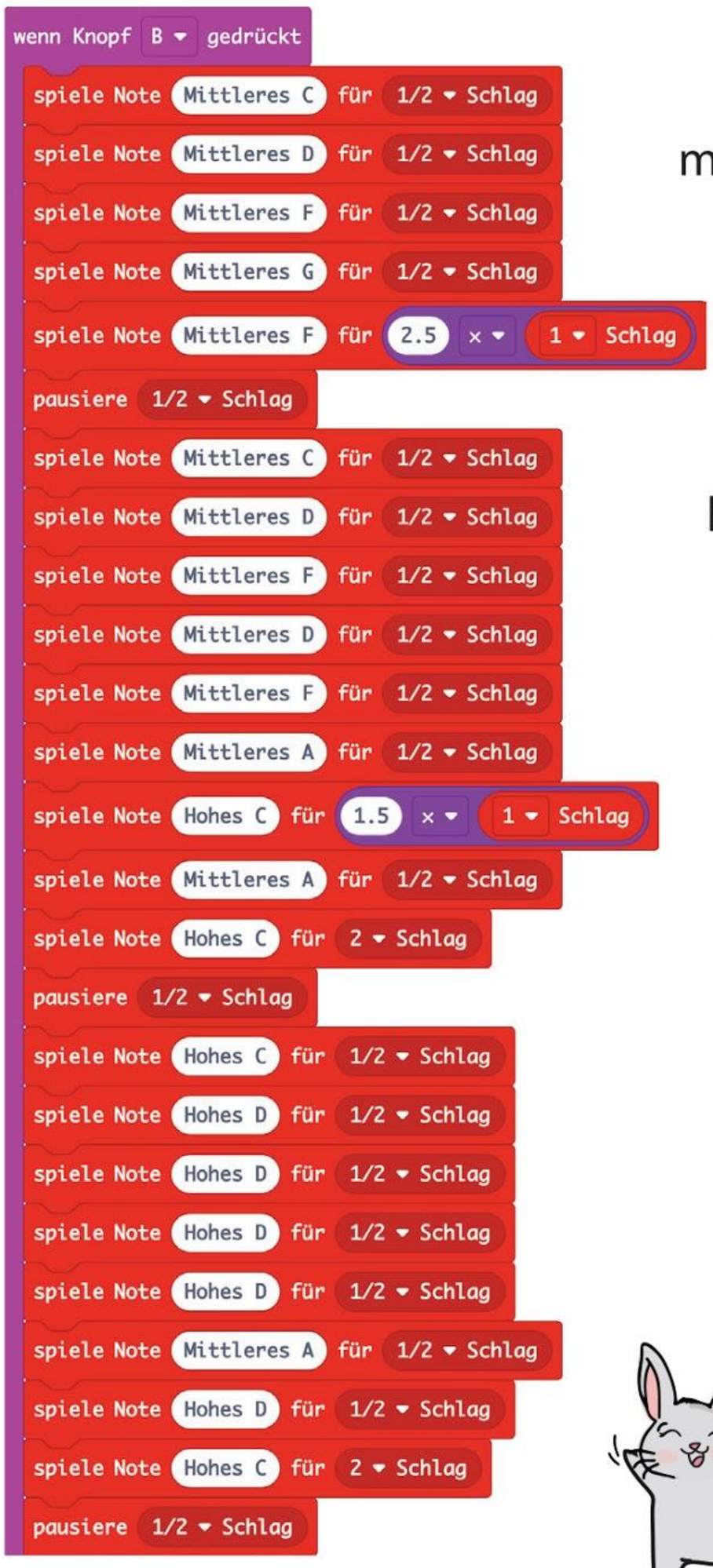
Beim Programmieren ist es schlau, immer wieder zu testen, ob dein Programm so funktioniert, wie du es beabsichtigst. Nutze dafür den Simulator.

Schritt 8 Programmiere selbst den Rest des Liedes, indem du weitere **[spiele Note _ für _]**-Blöcke einfügst und **‘Note’** sowie **‘Schlag’** entsprechend einstellst.
Du findest die richtigen Noten auf der nächsten Seite.

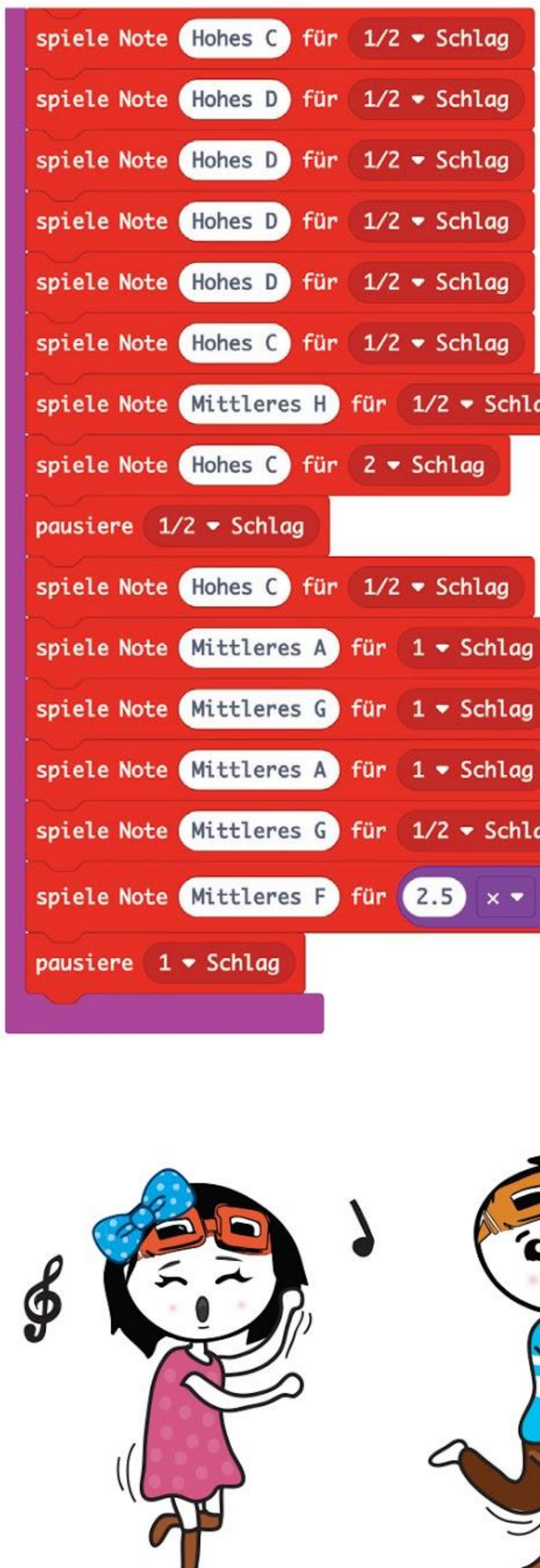
I Will Follow You



I will fol- low you wher- ev- er you may go,



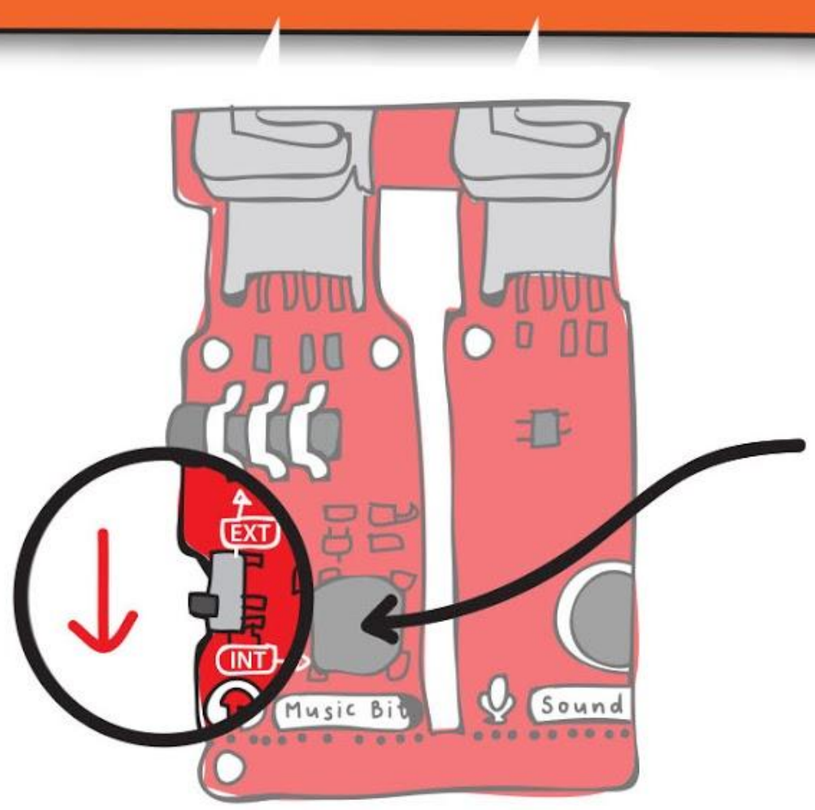
A moun- tain so high it can keep, Keep me a- way ... ...

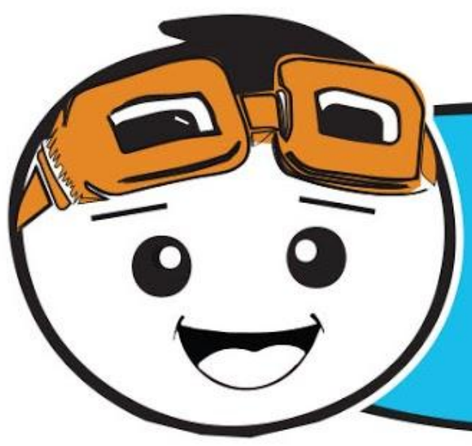


There is n't an o- cean too deep

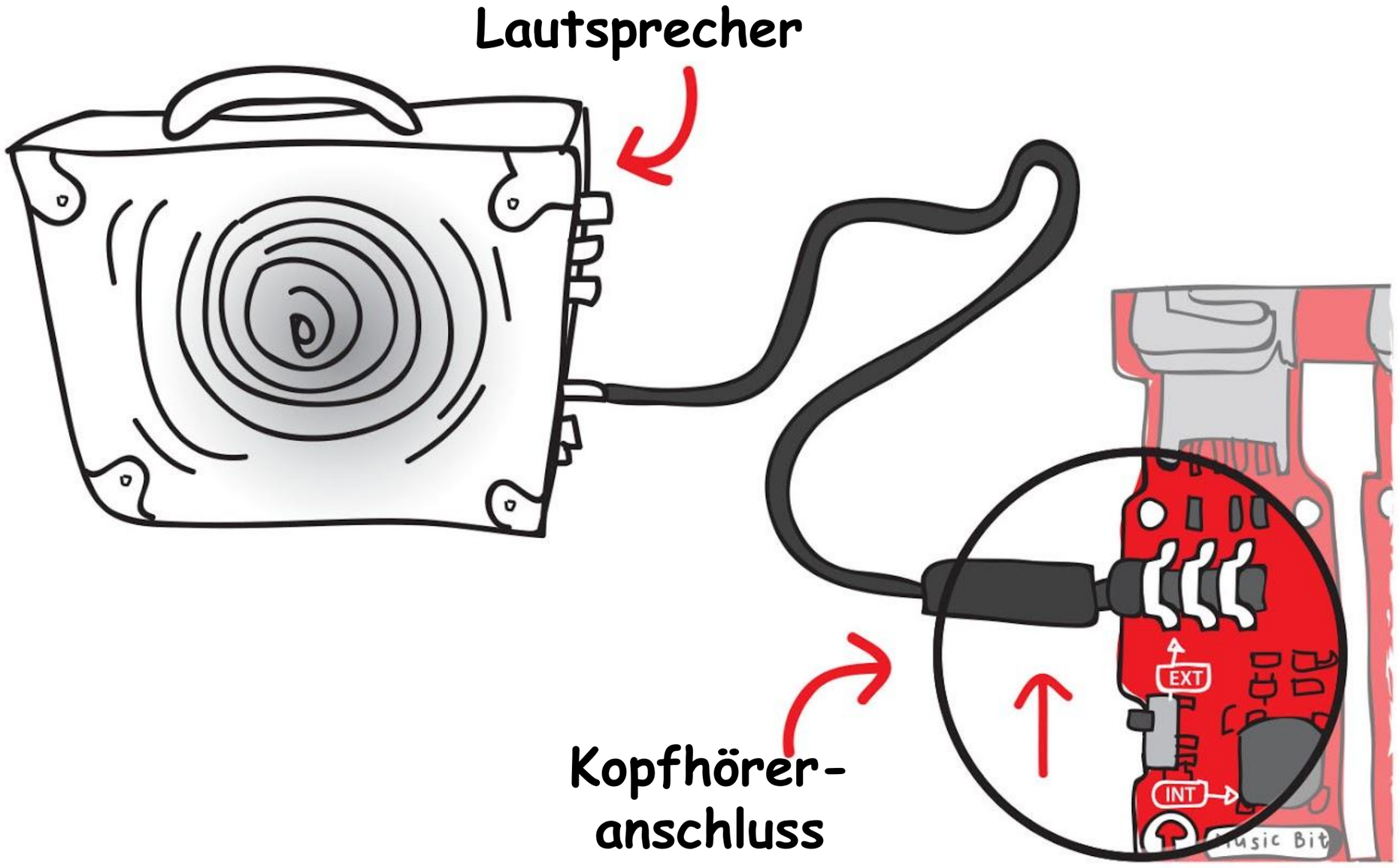
Schritt 9 Übertrage den fertigen Code auf deinen EDU:BIT.

EDU:BIT spielt den Song "I Will Follow You", wenn du auf den blauen Knopf (Knopf B) drückst. Denke daran, deinen EDU:BIT mit Strom zu versorgen und den Piezo-Summer einzuschalten, indem du den Schalter auf INT (intern) stellst.



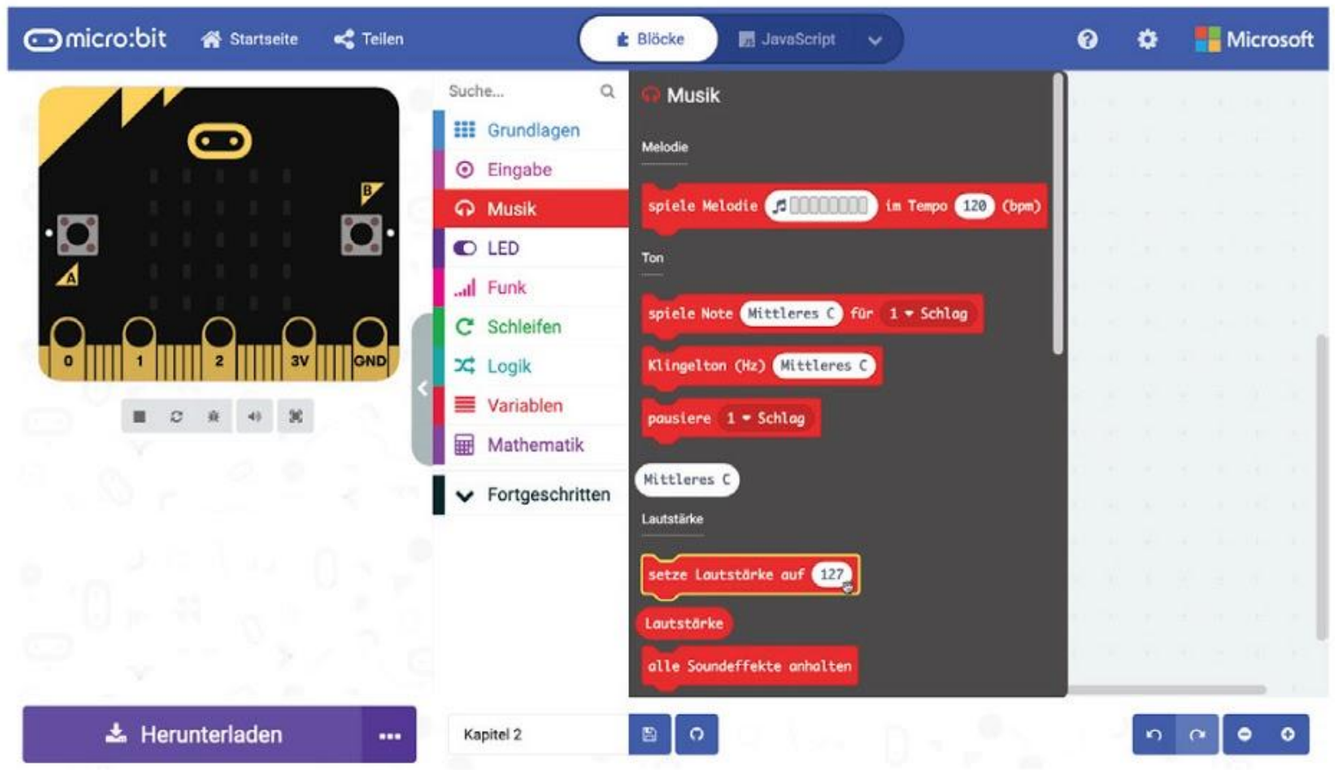


Du kannst auch einen Lautsprecher oder Kopfhörer an den Kopfhörerausgang auf deinem EDU:BIT anschließen. Dann musst du den Schalter auf EXT (external) stellen.



Ist die Musik zu leise? Oder zu laut? Du kannst einfach einen [setze Lautstärke auf _]-Block hinzufügen, um die Lautstärke zu ändern. Sie reicht von 0 bis 255 (maximale Lautstärke).

Schritt 10 Klicke in der Gruppe **[Musik]** auf **[setze Lautstärke auf _]**. Bewege diesen Block in den Block **[beim Start]** und ändere den Wert auf **200**.

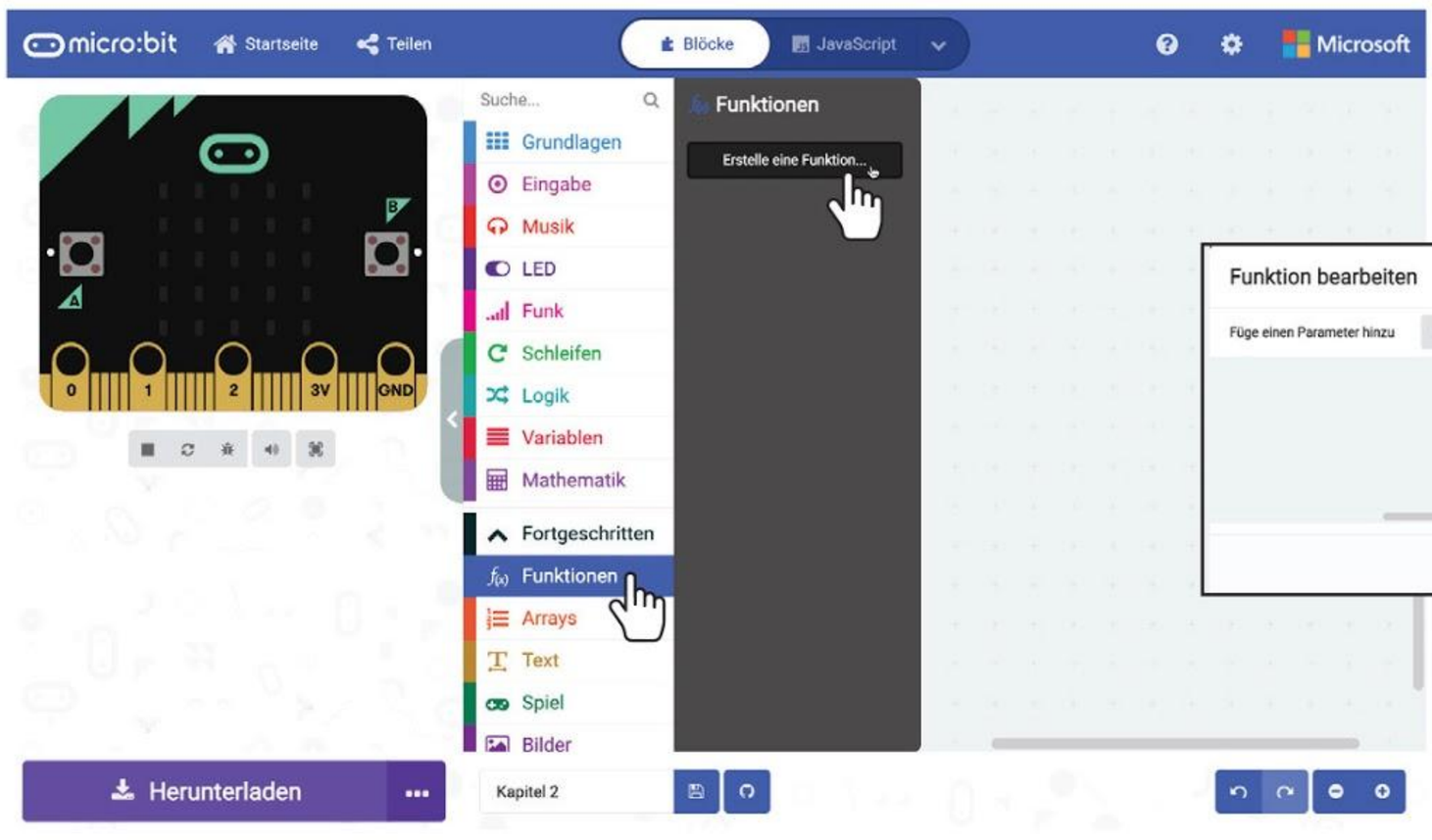


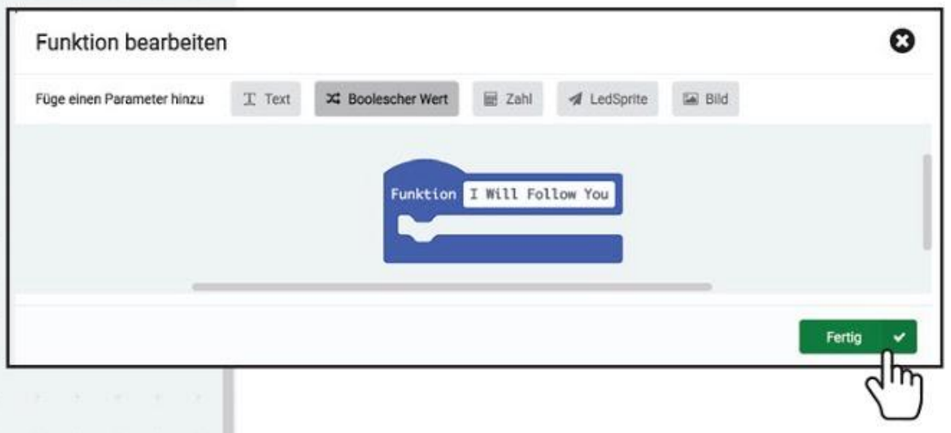
MERKE!

Wir können mehrere Code-Blöcke, die eine bestimmte Aufgabe erledigen, in eine **Funktion** geben, zum Beispiel den Code der "I Will Follow You" spielt.

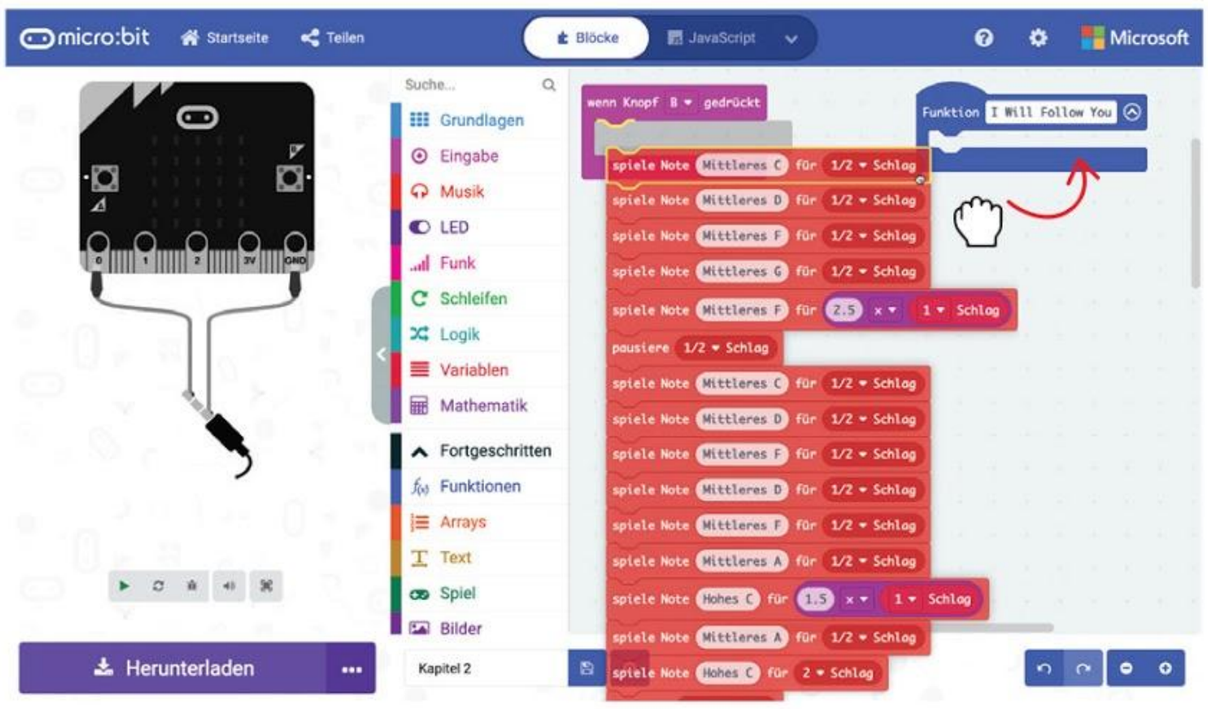
Eine Funktion ist einfach eine Sammlung von Code-Blöcken. Diese Funktion kann dann an mehreren Stellen in deinem Programm verwendet werden, ohne dass du dieselben Code-Blöcke immer wieder erstellen musst.

Schritt 11 Klicke auf die Gruppe **[Fortgeschritten]** und dann auf die Gruppe **[Funktionen]**. Klicke auf **[Erstelle eine Funktion]** und ändere im Dialogfenster 'macheEtwas' zu **'I Will Follow You'**. Klicke dann auf **'Fertig'.**

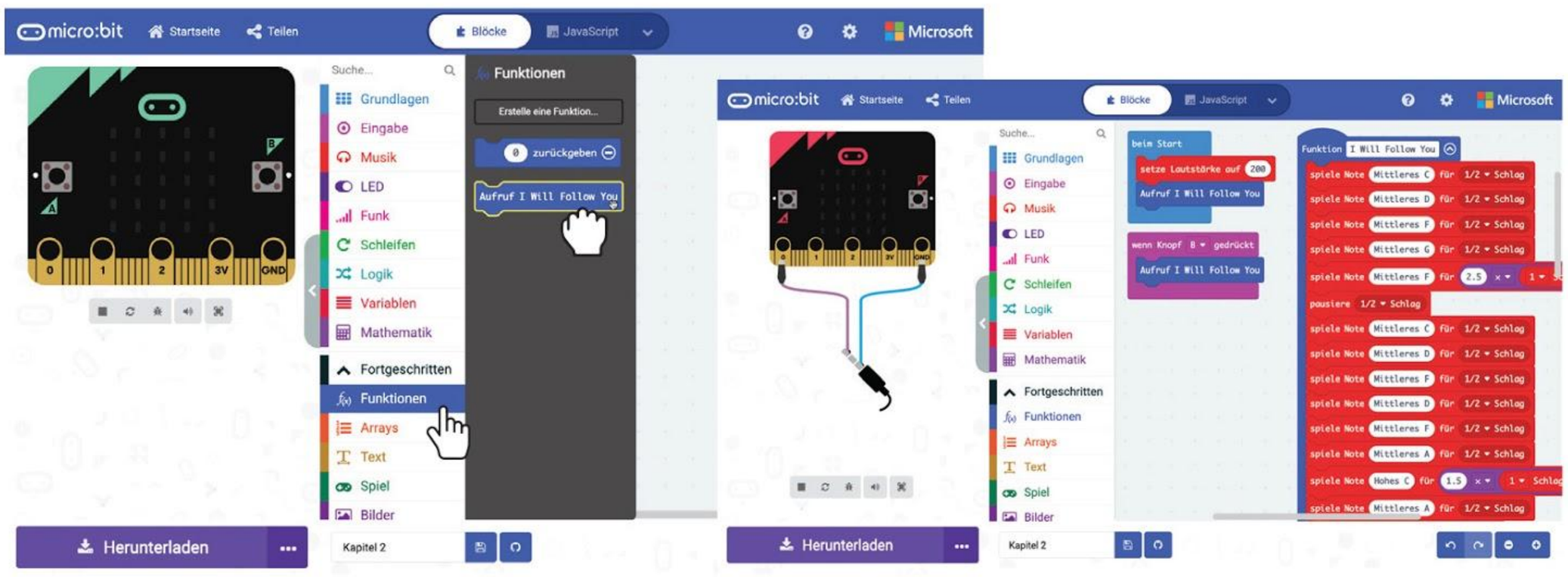




Schritt 12 Ein Block **[Funktion I Will Follow You]** erscheint in deinem Editor. Klicke auf den obersten Block im **[wenn Knopf B gedrückt]**-Block und ziehe alle Blöcke zu **[Funktion I Will Follow You]**.

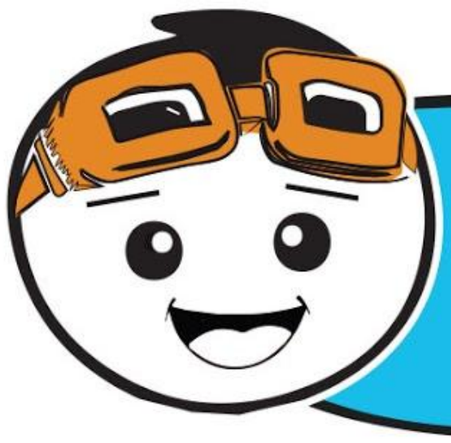


Schritt 13 Klicke auf die Gruppe **[Funktionen]** und wähle den Block **[Aufruf I Will Follow You]**. Dupliziere diesen Block. Füge die beiden Blöcke **[Aufruf I Will Follow You]** jeweils zu den Blöcken **[beim Start]** und **[wenn Knopf B gedrückt]** hinzu. Hier ist der Beispielcode:

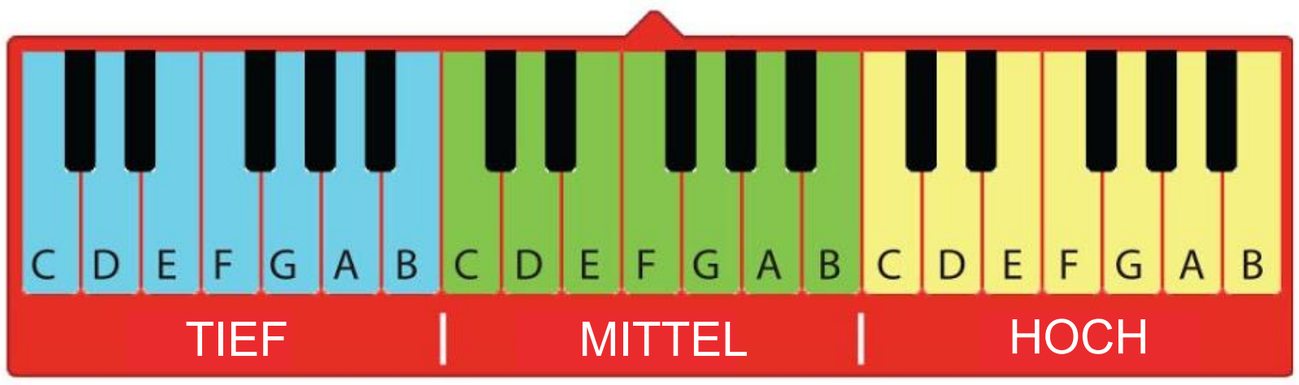


Schritt 14 Übertrage den fertigen Code auf deinen EDU:BIT. Genieße die Musik~

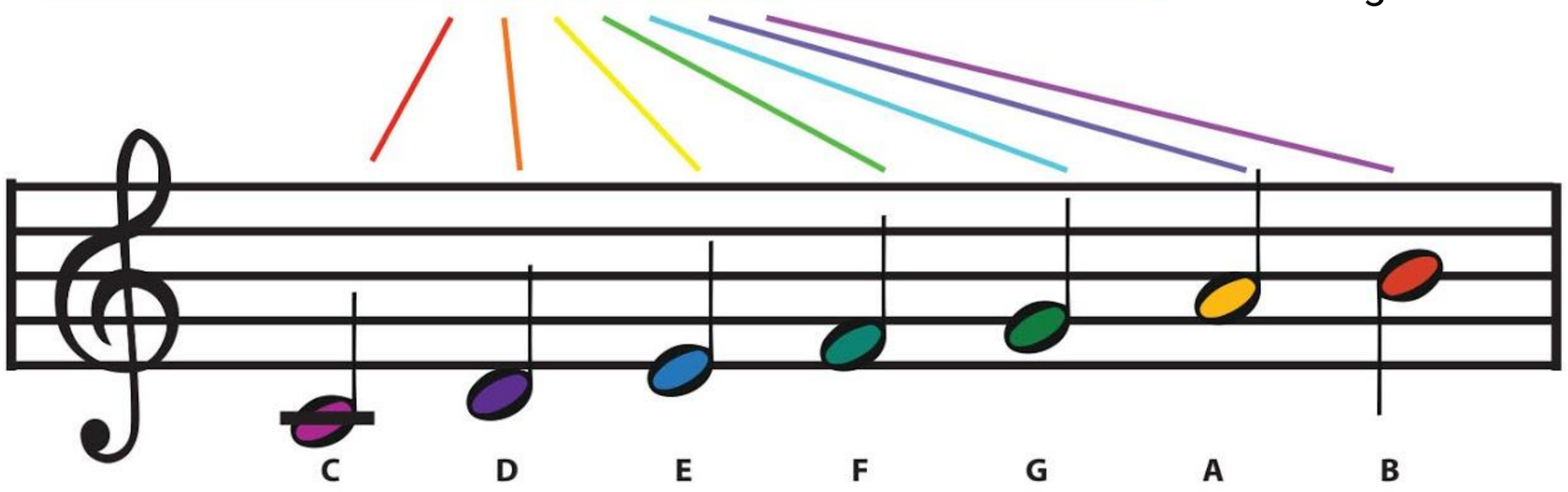
KNACKE DEN CODE



Wenn du Noten lesen kannst, kannst du EDU:BIT programmieren, alle möglichen Lieder zu spielen,. Hier hast du eine einfache Anleitung, die dir dabei hilft ein Notenblatt zu "dekodieren".



Die Position einer Note auf der Notenzeile (den fünf Linien) sagt uns, welche Note wir spielen müssen. Je höher die Note liegt, desto höher ist die Tonhöhe oder Frequenz des Tons, und umgekehrt.



Zeichen	Pause	Relative Länge	Dauer
𝅝	𝄻	Ganze Note	4 Schläge
𝅗𝅥	𝄼	Halbe Note	2 Schläge
♩	𝄽 oder	Viertelnote	1 Schlag
♪	𝄾	Achtelnote	1/2 Schlag
𝅘𝅥𝅯	𝄿	Sechzehntelnote	1/4 Schlag

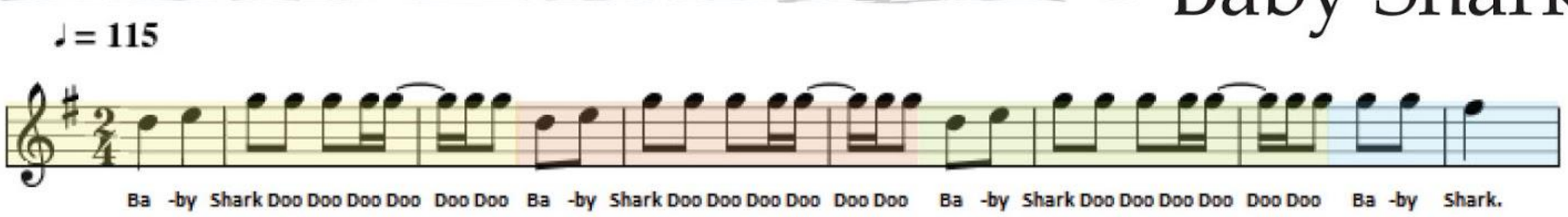
Verschiedene Notenzeichen sagen uns,
wie lange eine Note gehalten wird (die Dauer der Note).

KNACKE DEN CODE

Kannst du mit dieser Anleitung das folgende Lied "dekodieren"?

Baby Shark

♩ = 115



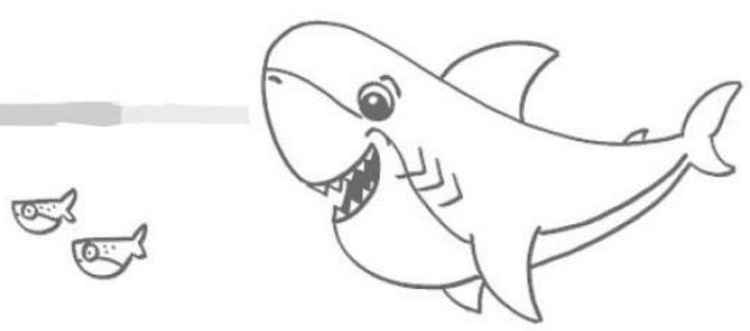
Line 1	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D	High E	High G		High G	High G	High G		High G
Beat	1		1/2	1/2	1/2		1/2	1/4	1/2

Line 2	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D		High G	High G	High G		High G	High G	High G
Beat	1/2	1/2	1/2		1/2	1/4	1/2		1/2

Repeat the same for Line 3

Line 4	Ba	-by	Shark
Note		High G	High F#
Beat	1/2	1/2	

Programmiere EDU:BIT "Baby Shark" zu spielen, wenn der gelbe und der blaue Knopf (A+B) gleichzeitig gedrückt werden.



HINWEIS!

Mit dem Block [setze Lautstärke auf _] kannst du die Lautstärke einstellen.

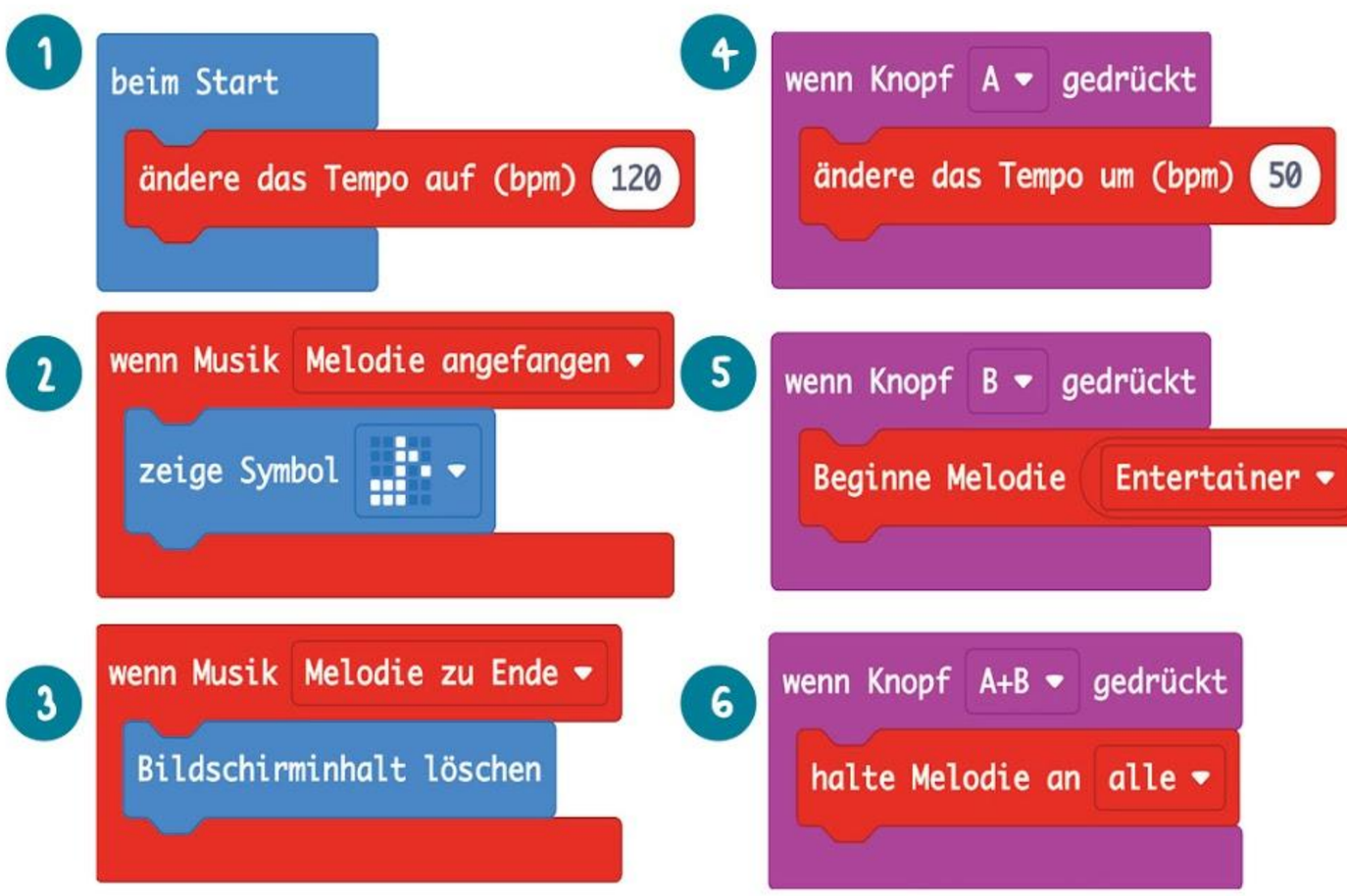
ENTDECKE NOCH MEHR

#1 Du kannst das “Tempo” (die Geschwindigkeit deines Liedes) mit dem Block **[ändere das Tempo auf (bpm) _]** einstellen. Je höher die bpm (Schläge pro Minute) sind, desto schneller wird dein Lied. Benutze den Block **[ändere das Tempo um (bpm) _]** für relative Tempo-Änderungen..

#2 Der Block **[halte Melodie an _]** stoppt die Melodie, die gerade gespielt wird.

#3 Du kannst den Block **[wenn Musik _]** mit seinen Bedingungen, wie “Melodie angefangen” und “Melodie zu Ende”, als Event-Trigger in deinem Code benutzen.

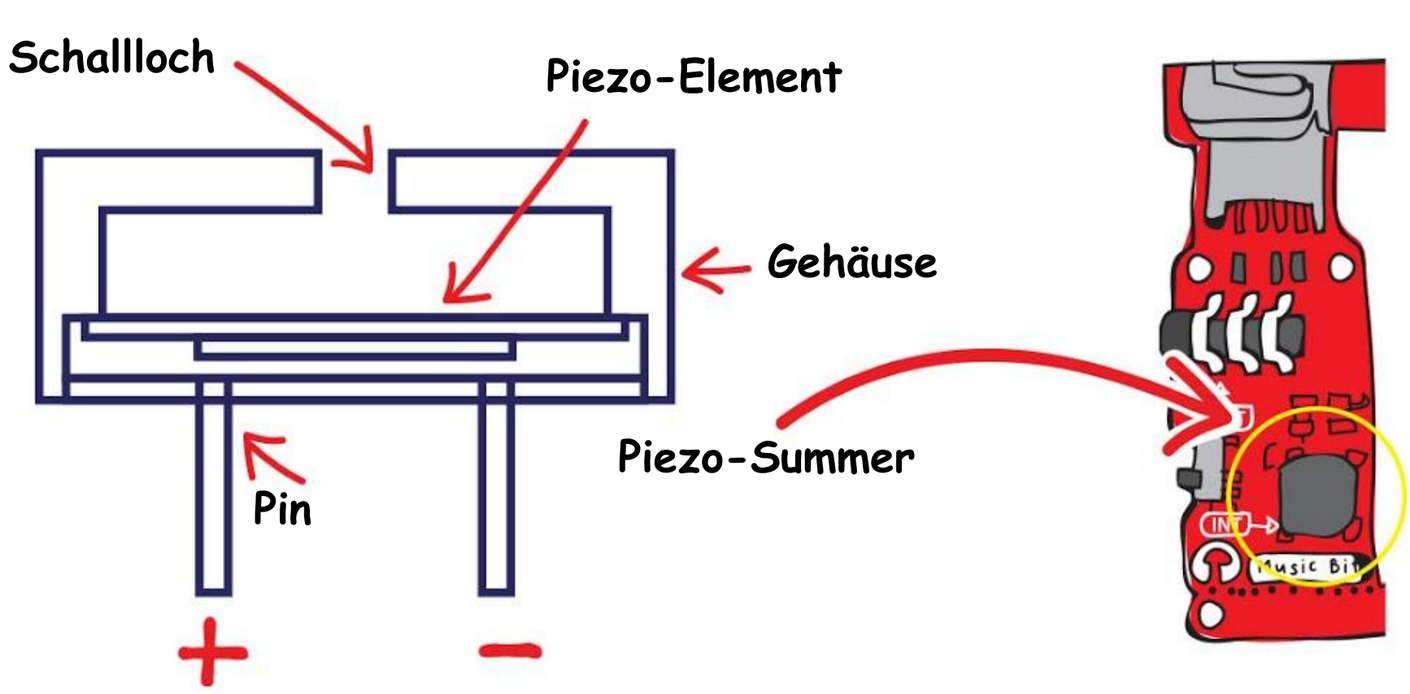
Hier ist ein Beispielcode:



1. Mit diesen Befehlen wird das Anfangs-Tempo auf 120 bpm gesetzt.
2. Wenn die Melodie anfängt, zeigt die LED-Matrix ein Notensymbol an.
3. Wenn die Melodie zu Ende ist, werden alle LEDs auf der Matrix ausgeschaltet.
4. Jedesmal wenn Knopf A gedrückt wird, wird das Tempo um 50 bpm erhöht.
5. Die Melodie “Entertainer” wird gespielt, wenn Knopf B gedrückt wird.
6. Die Melodie wird angehalten, wenn die Knöpfe A+B gleichzeitig gedrückt werden.

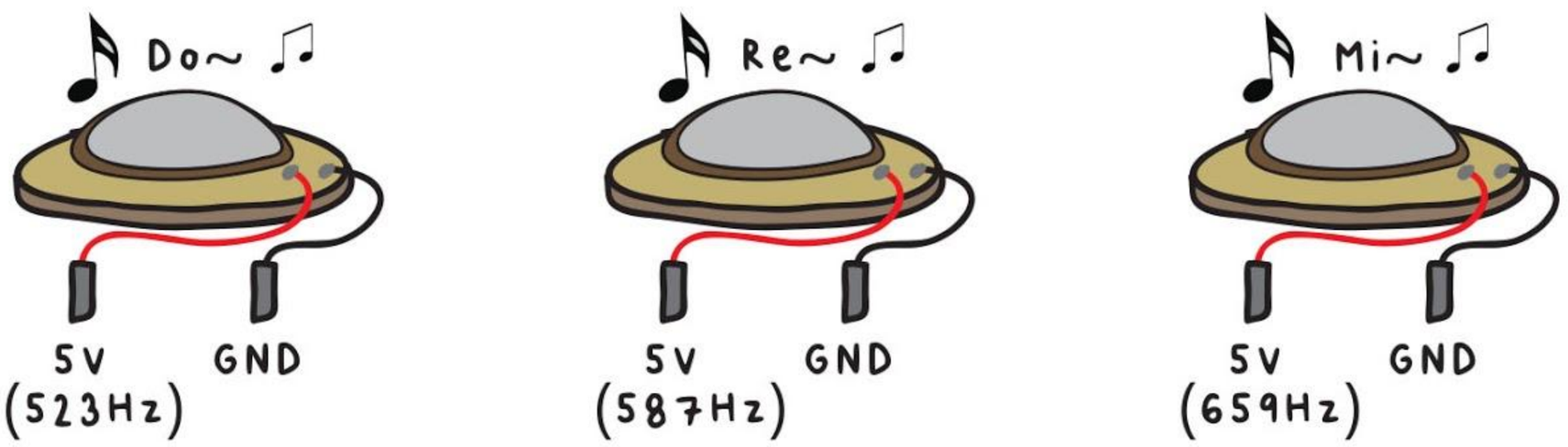
GUT ZU WISSEN!

Piezo-Summer werden oft für die Tonwiedergabe genutzt. Ein Piezo-Element vibriert, wenn ein elektrisches Signal hindurchfließt.



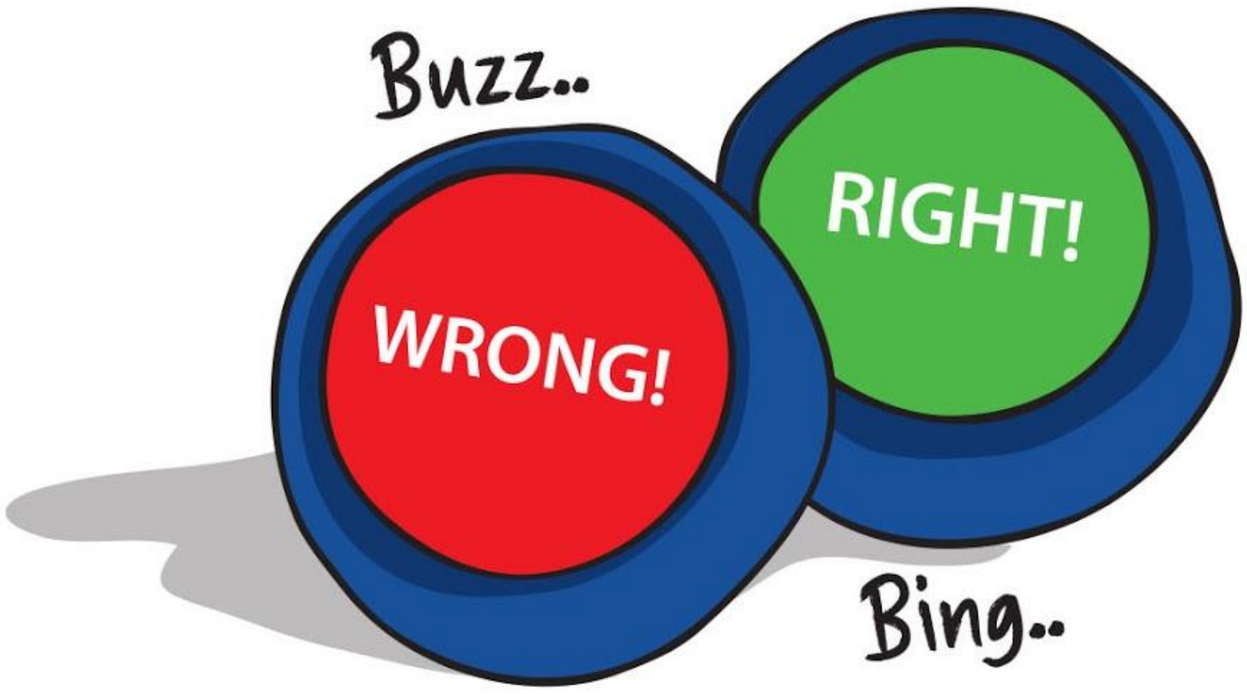
Durch das Ändern der Frequenz des elektrischen Signals ändert sich die Geschwindigkeit der Vibration; somit entstehen Klänge mit unterschiedlichen Tonhöhen.

Piezo-Element



Das menschliche Ohr kann Töne im Frequenzbereich zwischen 20 Hz und 20.000 Hz hören. Tiefere Töne als 20 Hz werden Infraschall und höhere Töne als 20.000 Hz Ultraschall genannt.

HERAUSFORDERUNG



Schreibe ein Programm, das EDU:BIT in einen einen Buzzer für eine Fernsehshow verwandelt.

beim Start	Zeige einen Smiley an.
wenn Knopf A gedrückt (gelber Knopf)	Zeige das Symbol ✔ an und spiele die Melodie “Einschalten” einmal.
wenn Knopf B gedrückt (blauer Knopf)	Zeige das Symbol ✕ an und spiele die Melodie “wawawawaa” einmal.
wenn Knöpfe A+B gedrückt	Lösche den Bildschirminhalt.

KAPITEL 4

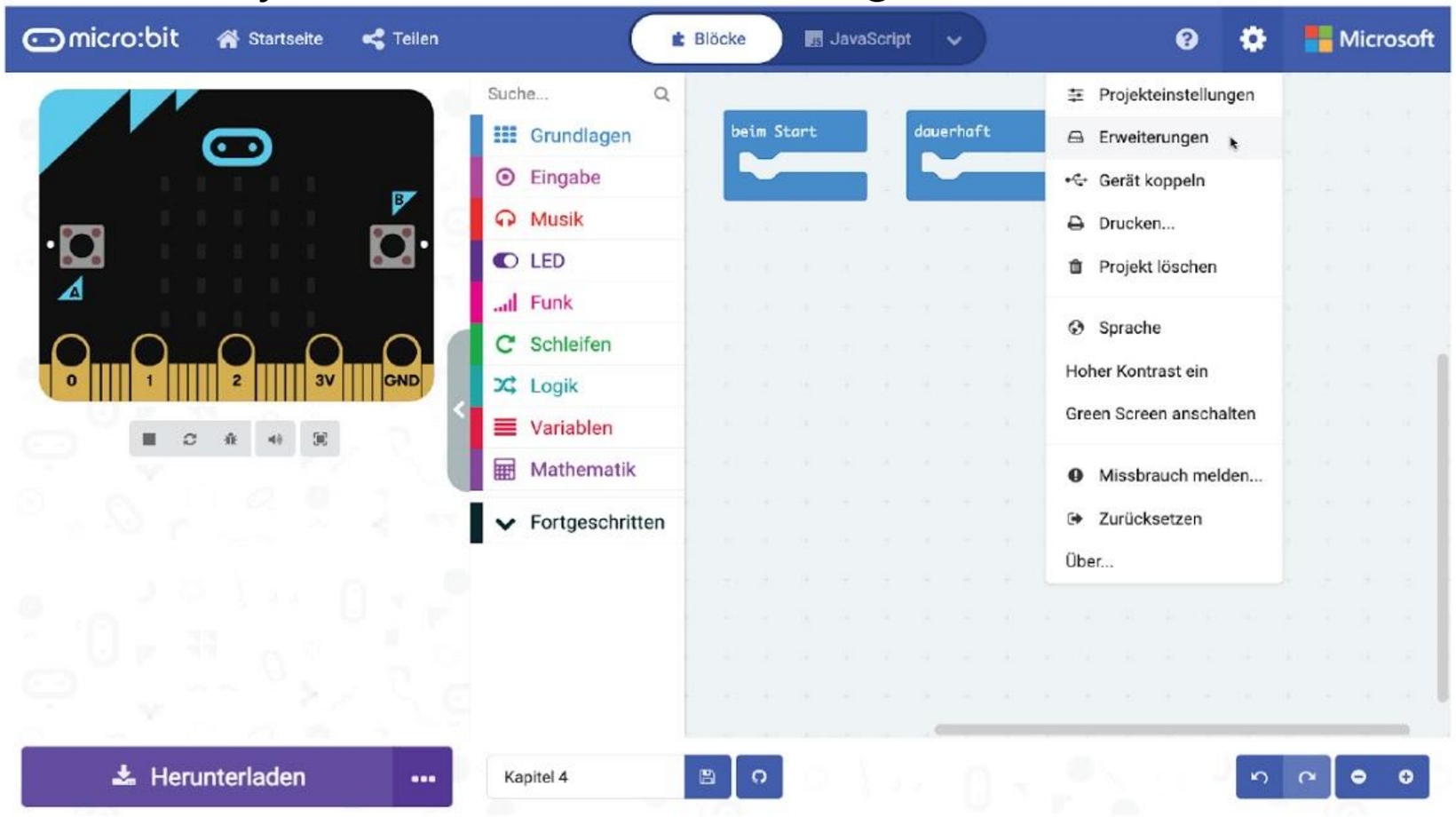
Begriffe raten~

Ampel Bit

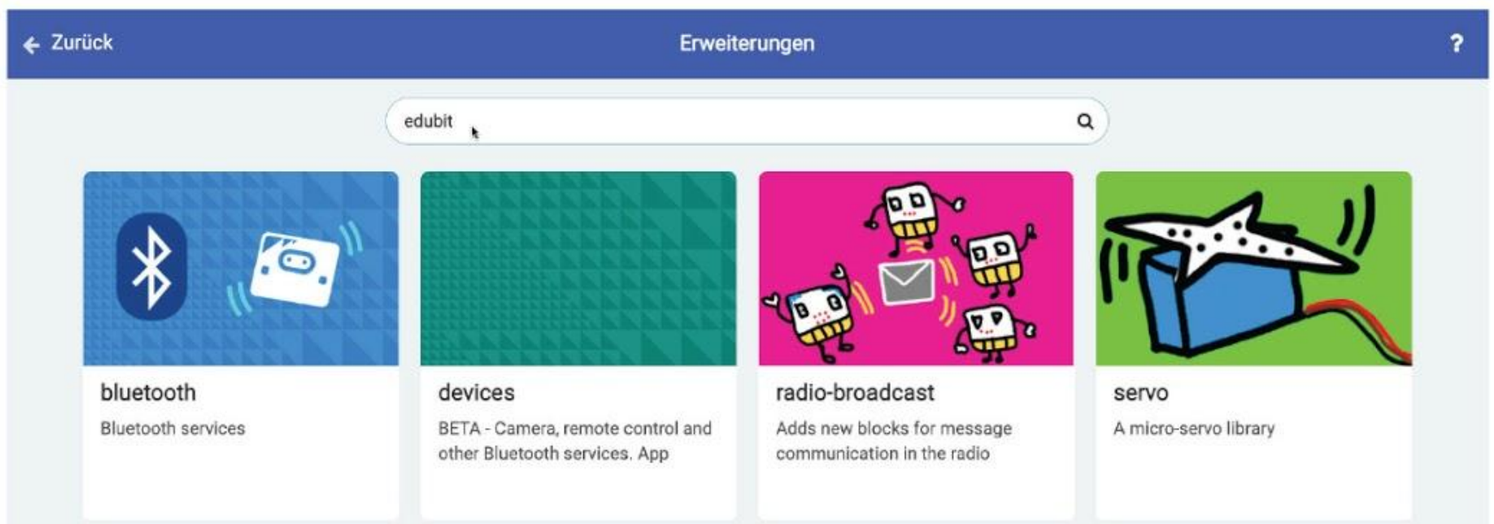
Hast du schon die Reihe mit einem roten, gelben und grünen LED auf deinem EDU:BIT entdeckt? Das ist Ampel Bit. Um es zu programmieren brauchst du die EDU:BIT Erweiterung für deinen MakeCode Editor. Erweiterungen sind spezielle Blöcke, die wir zum Editor hinzufügen, damit wir Zubehör für den micro:bit wie zum Beispiel die EDU:BIT-Platine programmieren können.

LASS UNS PROGRAMMIEREN!

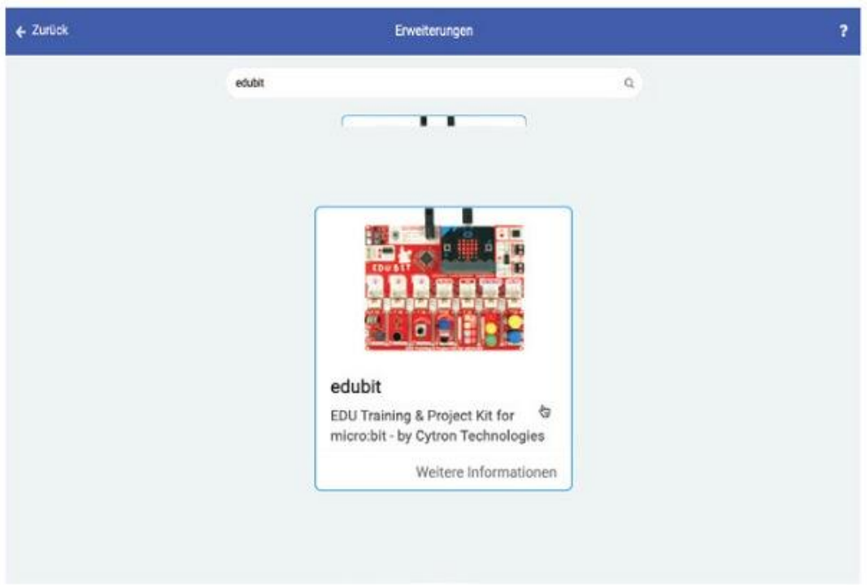
SCHRITT 1 Erstelle ein neues Projekt in deinem MakeCode Editor. Klicke auf das Zahnrad-Symbol und auf **‘Erweiterungen’**. *Dafür brauchst du eine Internetverbindung.

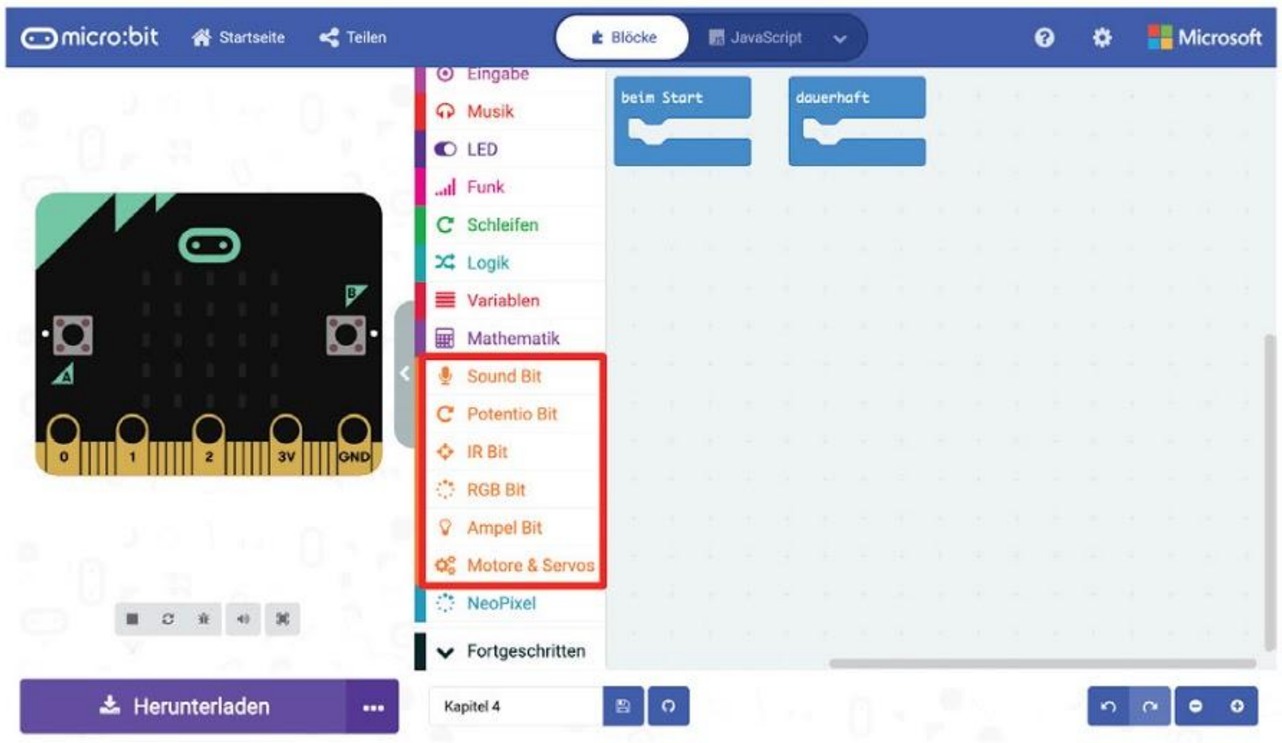


Schritt 2 Schreibe “**edubit**” in das Suchfeld und drücke auf Enter.

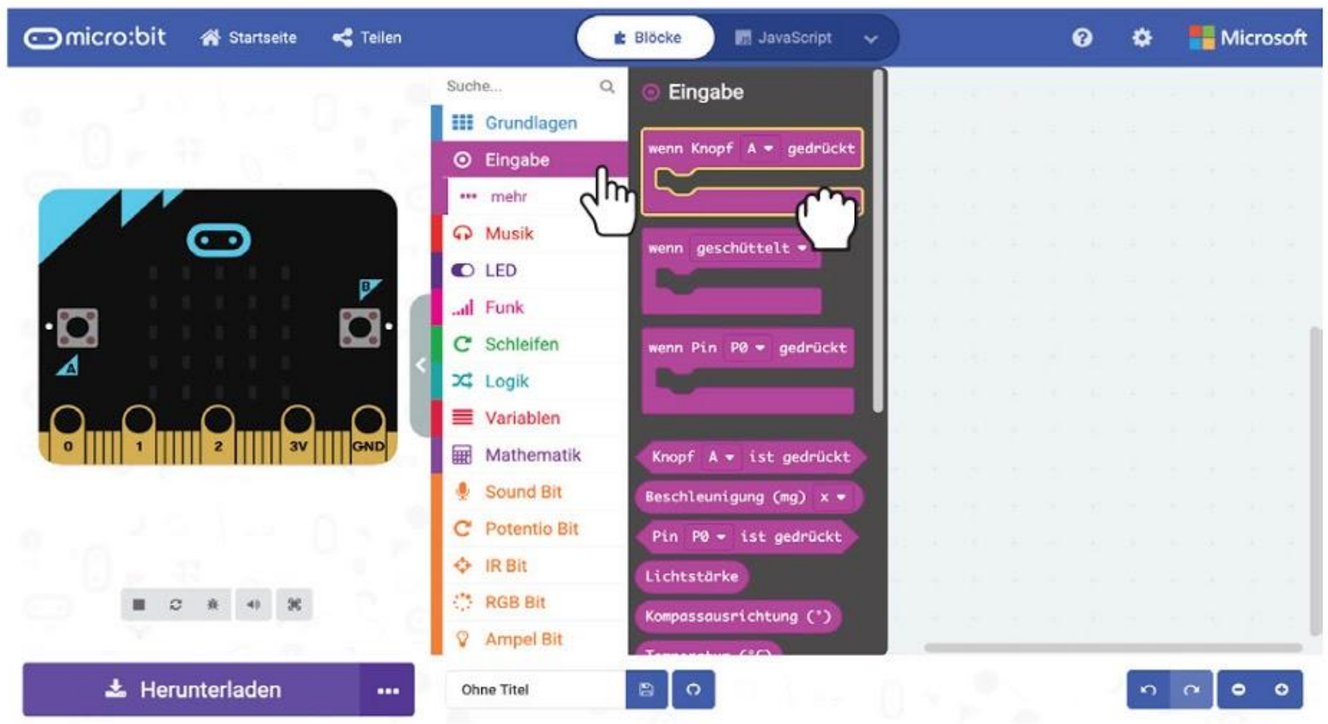


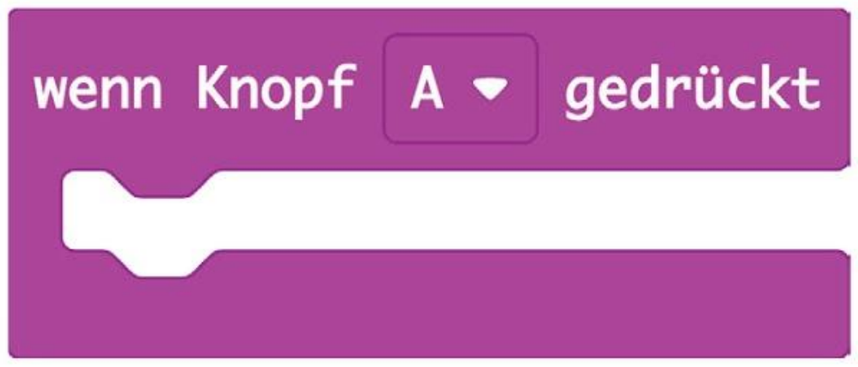
Schritt 3 Klicke auf die Erweiterung **‘edubit’**. Warte bis sie geladen ist und achte auf diese Veränderungen in deinem MakeCode Editor.



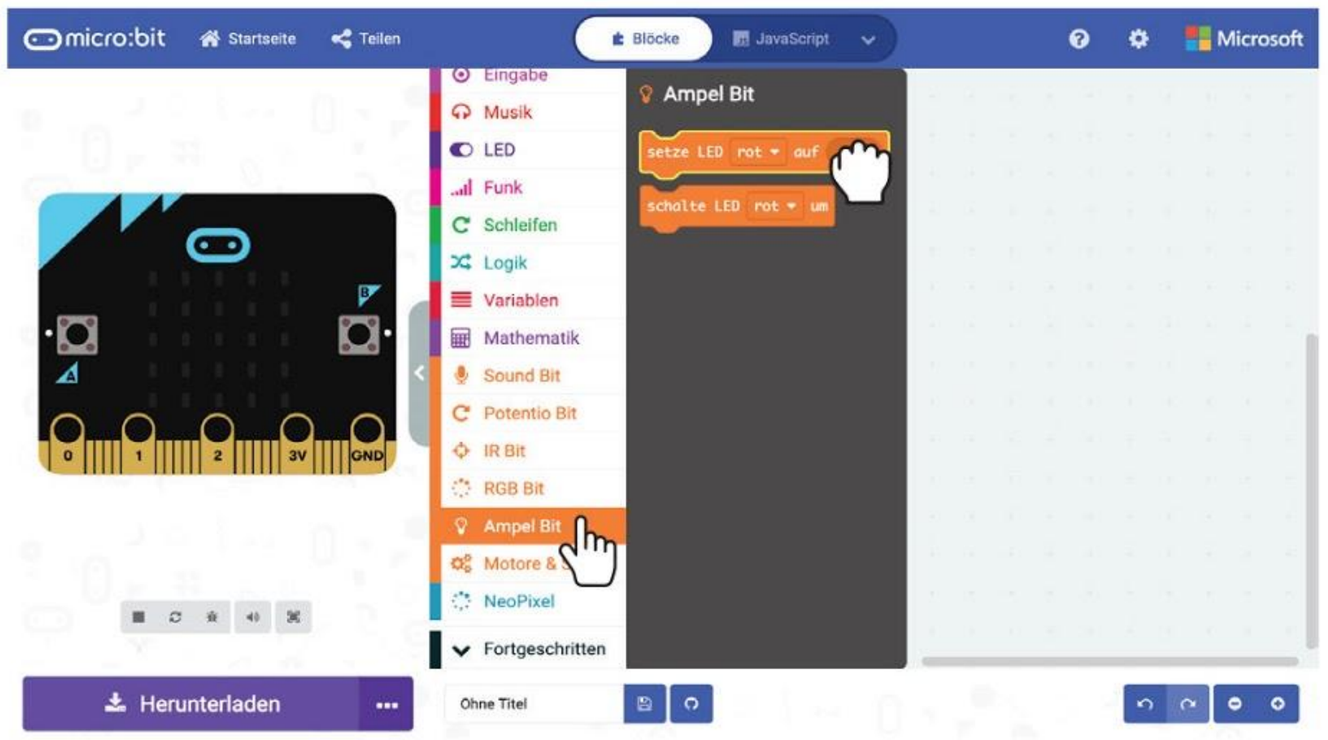


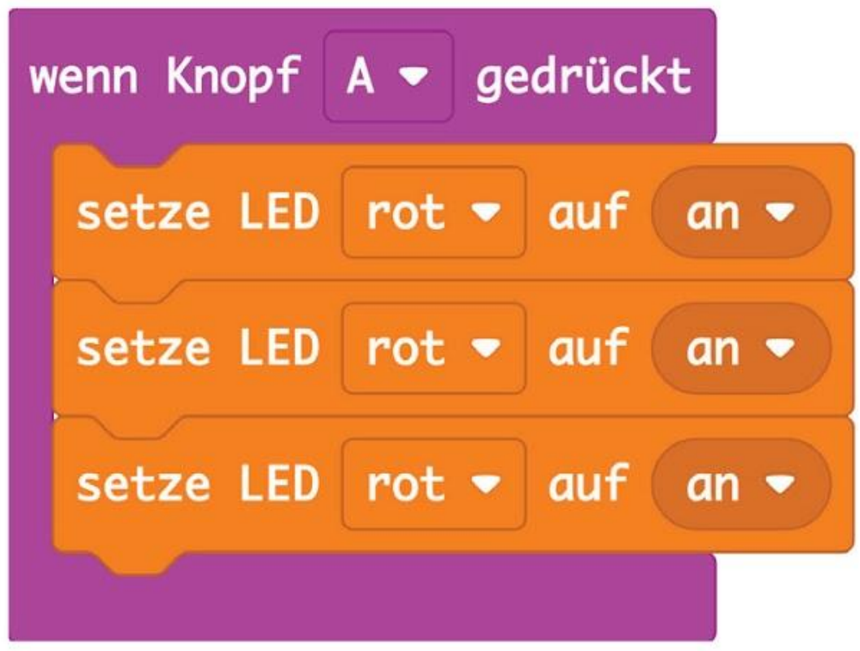
Schritt 4 Klicke auf **[Input]** und wähle den Block **[wenn Knopf _ gedrückt]** aus.



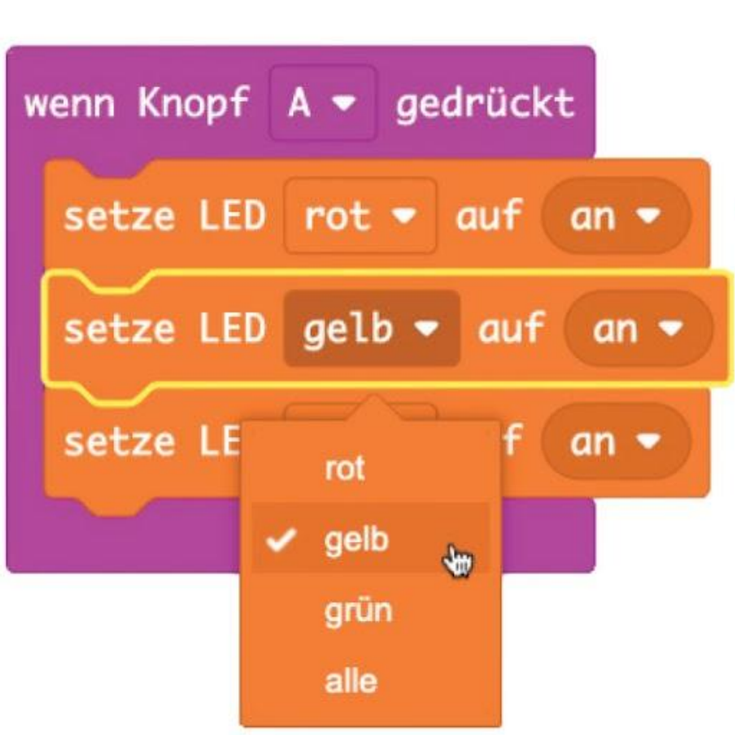


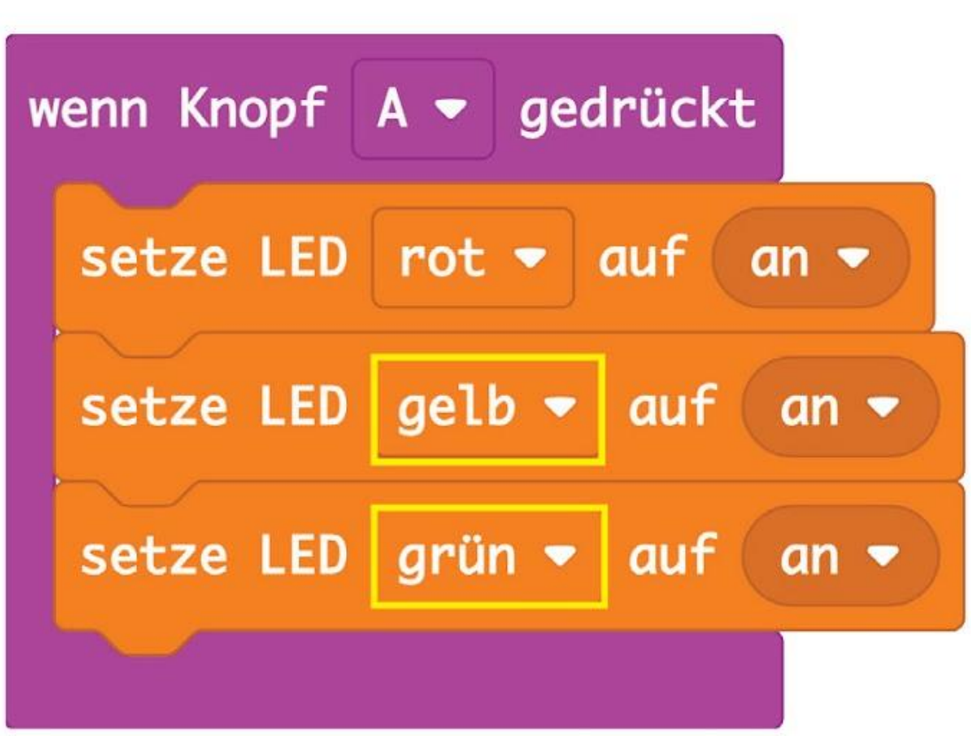
Schritt 5 Klicke in der Gruppe **[Ampel Bit]** auf den Block **[setze LED _ auf _]**. Klicke im Arbeitsbereich auf den **[setze LED _ auf _]**-Block und klicke auf ‘Duplizieren’. Wiederhole diesen Schritt, bis du drei **[setze LED _ auf _]**-Blöcke hast. Ziehe diese in den Block **[wenn Knopf A gedrückt]**.



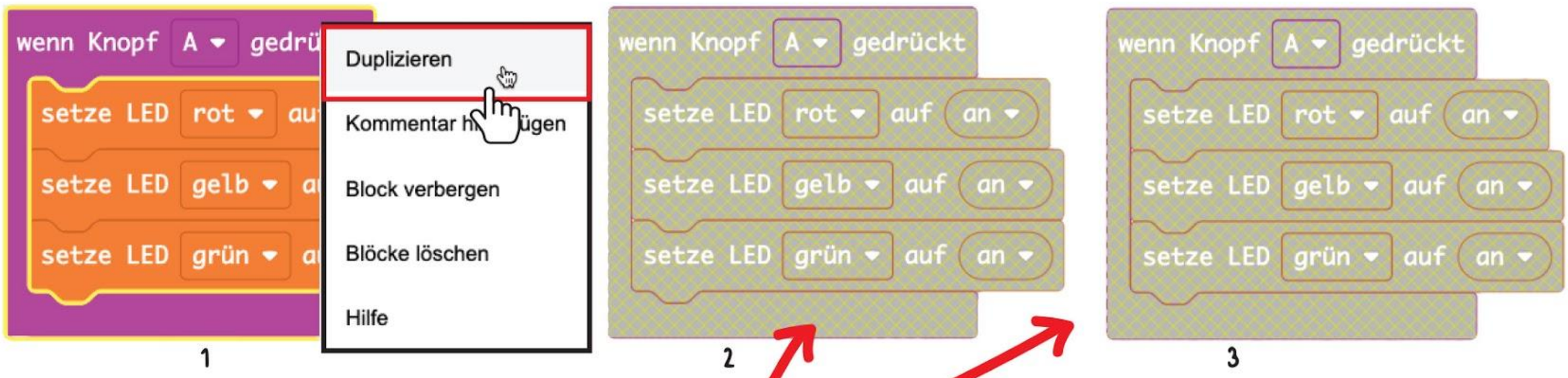


Schritt 6 Klicke auf die Farbauswahl und ändere den zweiten und dritten Block jeweils auf **‘gelb’** und **‘grün’**.





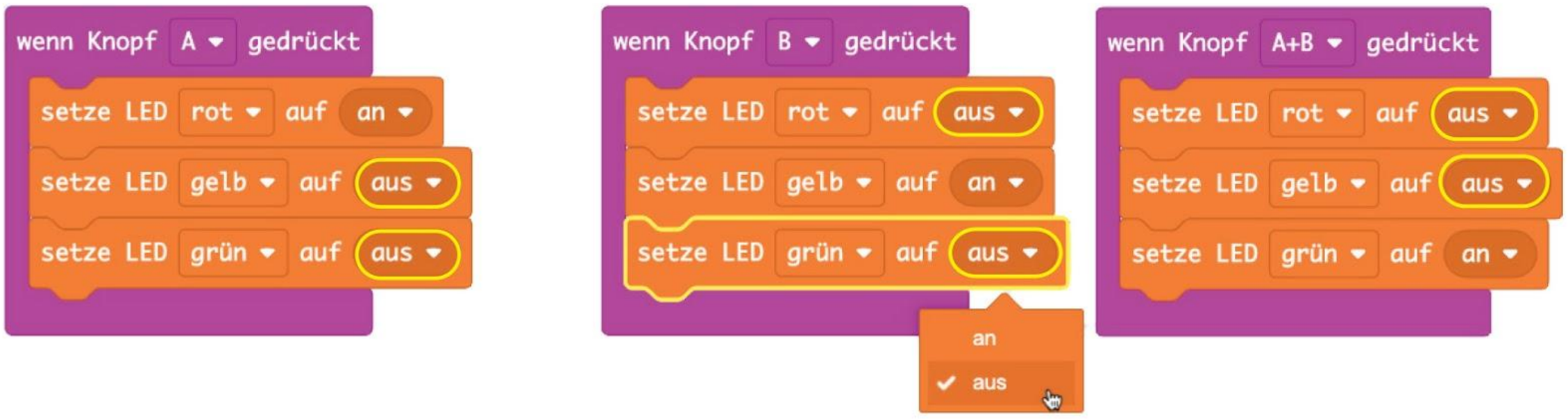
Schritt 7 Klicke mit der rechten Maustaste auf **[wenn Knopf _ gedrückt]** und wähle ‘Duplizieren’. Wiederhole das, bis du dreimal die gleichen Blöcke hast.



*Diese Blöcke sind gesperrt und werden nicht ausgeführt, weil es nicht mehrere **[wenn Knopf A gedrückt]**-Blöcke geben darf.

Schritt 8 Ändere “A” beim zweiten und dritten **[wenn Knopf _ gedrückt]**-Block jeweils in “**B**” und “**A+B**”.

Schritt 9 Ändere die Status der LEDs von ‘an’ auf ‘aus’ wie im folgenden Bild:



Schritt 10 Flashe den Code auf deinen EDU:BIT und beobachte, was passiert, wenn du Knopf A, Knopf B und dann die Knöpfe A+B gleichzeitig drückst.

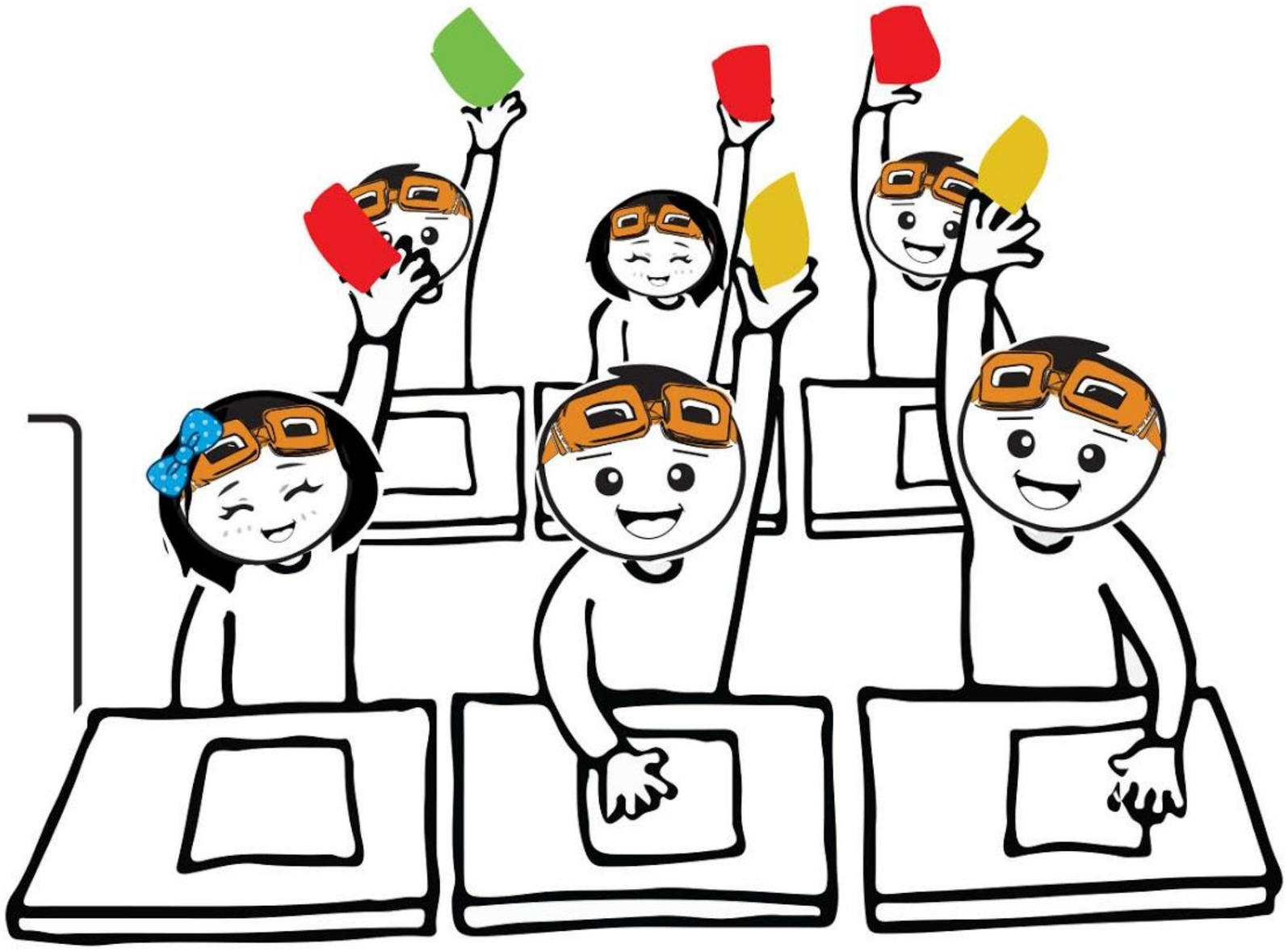
KAPITEL 4 : Begriffe raten~

STOP!

WARTE!

KOMM HEREIN!

Wow… du kannst deinen EDU:BIT jetzt als persönliche Anzeige benutzen. Fallen dir auch andere Möglichkeiten ein?

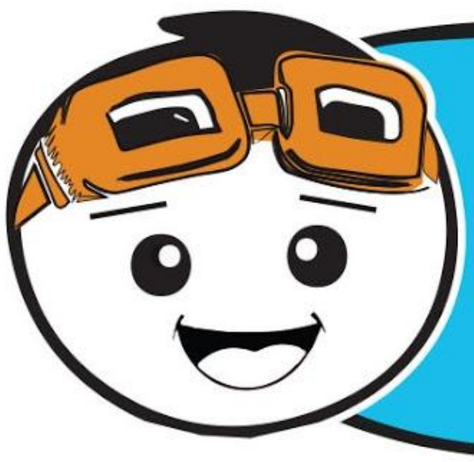


AMPEL RÜCKMELDUNG

"Bitte um Hilfe!"

"Ich arbeite noch"

"Geschafft!"



Eine LED, oder Leuchtdiode, ist ein digitales Ausgabegerät. Das heißt es hat nur zwei mögliche Zustände - AN oder AUS; wobei AN oft mit 1 (eins) angegeben wird, und AUS mit 0 (null).

KAPITEL 4 : Begriffe raten~

Du kannst deinen EDU:BIT auch so programmieren, dass er wie ein Timer funktioniert. Hier ist der Beispielcode.

Der Timer wird eingeschaltet durch Schütteln des EDU:BIT.

Spiele ein Signal für den Start des Timers.

Die grüne LED leuchtet.

Die gelbe LED leuchtet.

Die rote LED leuchtet.

Spiele die Melodie “wawawawaa”, wenn die Zeit vorbei ist.

Schalte die rote LED zehnmal um.

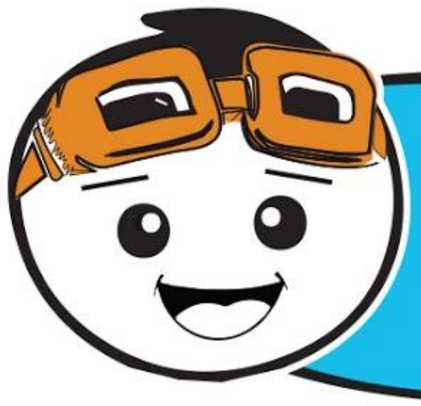
```
wenn geschüttelt
spiele Note Mittleres C für 1 Schlag
setze LED rot auf aus
setze LED gelb auf aus
setze LED grün auf an
pausiere (ms) 2000
setze LED rot auf aus
setze LED gelb auf an
setze LED grün auf aus
pausiere (ms) 2000
setze LED rot auf an
setze LED gelb auf aus
setze LED grün auf aus
pausiere (ms) 2000
Beginne Melodie wawawawaa Wiederhole einmal
10 -mal wiederholen
mache
    schalte LED rot um
pausiere (ms) 500
```

In diesem Beispiel leuchtet jede LED für 2000 ms (2 Sekunden).

Wenn du möchtest, dass jede LED für eine Minute aufleuchtet, welchen Wert müsstest du hier eingeben?

Hier ist ein Hinweis:
1 Minute = 60 Sekunden

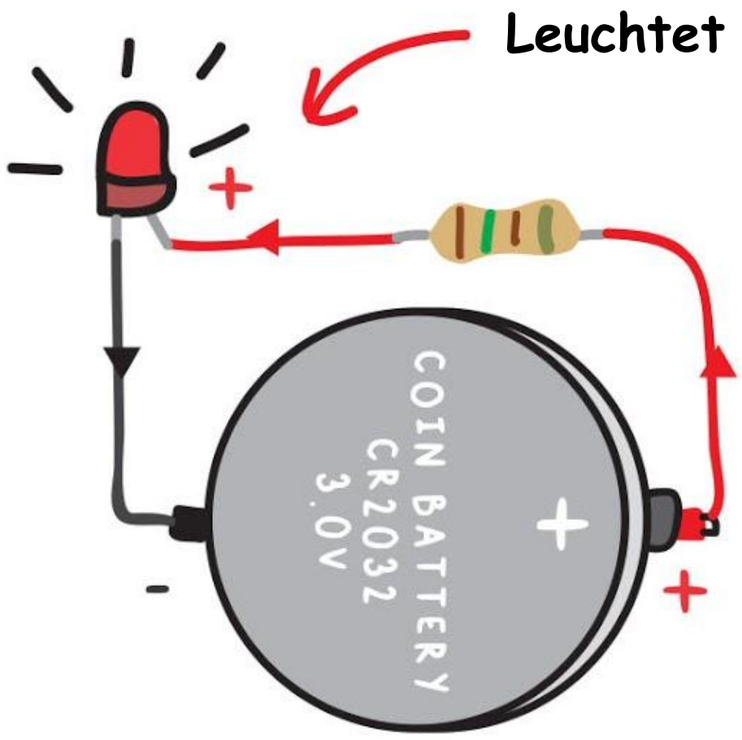
```
schalte LED rot um
```

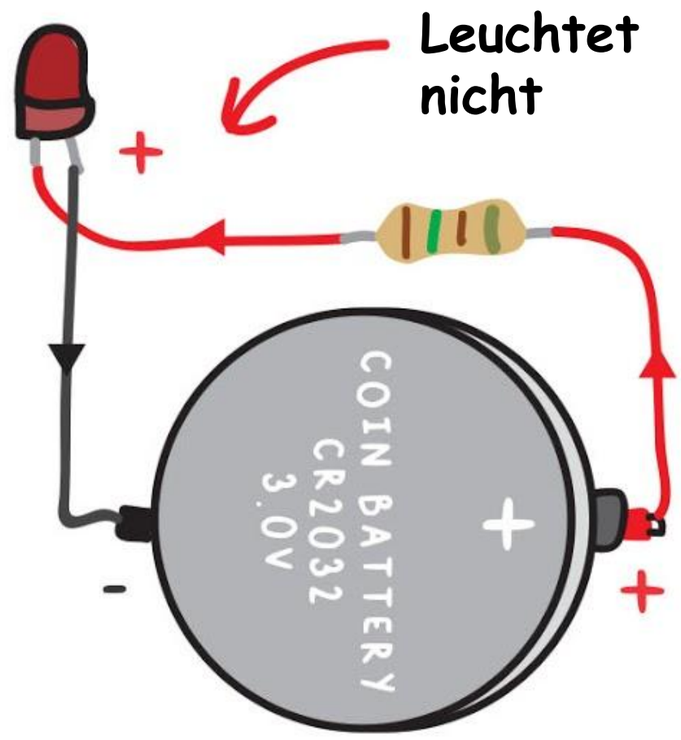


“Umschalten” bedeutet, von einem Zustand auf einen anderen zu wechseln. Wenn der aktuelle Zustand AN ist, wird auf AUS geschaltet, und umgekehrt. Das heißt, wenn wir eine LED immer wieder umschalten, blinkt sie.

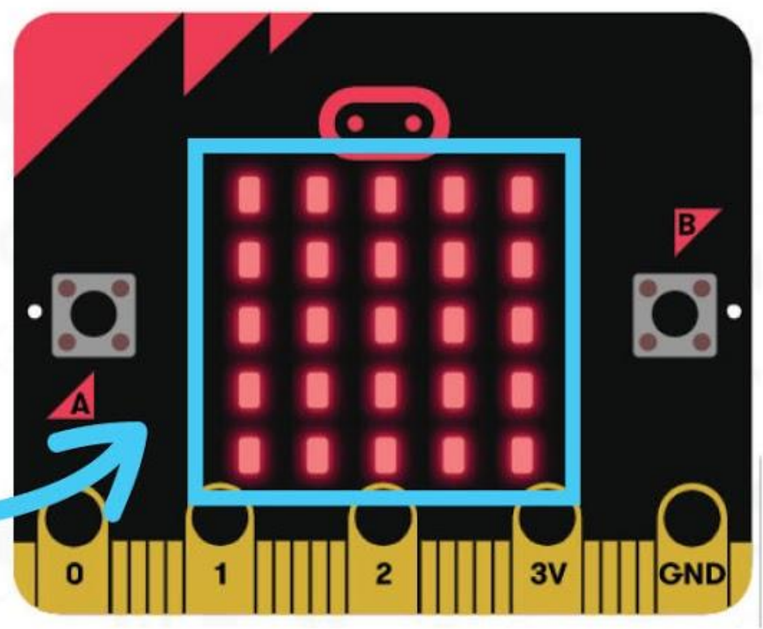
GUT ZU WISSEN!

Eine **Leuchtdiode (LED)** ist ein Halbleiter-Bauelement, das Strom in Licht umwandelt. Sie hat zwei Anschlüsse, einen positiven und einen negativen Anschluss. Wenn eine LED in der richtigen Polarität angeschlossen wird und Strom durch sie fließt, leuchtet sie.





Die LEDs auf dem micro:bit sind ohne Drähte auf der Oberfläche montiert. Man nennt das SMT (surface-mount technology).



Außer den LEDs auf dem micro:bit gibt es noch 41 andere SMT-LEDs auf dem EDU:BIT. Kannst du sie alle finden?

youtu.be/qqBmvHD5bCw

HERAUSFORDERUNG

Programmiere EDU:BIT, dass er als Punktezähler und als Timer für Spiele wie “Begriffe raten” und “Scharade” verwendet werden kann.	
beim Start	Setze Variable Team A = 0 Setze Variable Team B = 0
wenn Knopf A gedrückt (gelber Knopf)	Ändere Team A um 1. Zeige die aktuellen Punkte von A.
wenn Knopf B gedrückt (blauer Knopf)	Ändere Team B um 1. Zeige die aktuellen Punkte von B.
wenn Knöpfe A+B gedrückt	Scrolle die aktuellen Punkte von Team A und Team B
wenn geschüttelt	Starte einen Timer, der eine Minute lang ist, indem die **grüne LED** (für 30 Sekunden), dann die **gelbe LED** (für 20 Sekunden) und schließlich die **rote LED** (für 10 Sekunden) leuchtet. Spiele die Melodie “wawawawaa” wenn die Zeit um ist. Schalte die **rote LED** 10 Mal um.

Hier ist ein Tipp für dich: Du musst zwei Variablen erstellen und sie jeweils 'Team A' und 'Team B' benennen.

Spielen wir!

Win, Lose or Draw~

SPIELANLEITUNG:

- Teile die Klasse in 2 Teams - Team A und Team B.
- Eine Person von Team A zieht eine Karte mit einem Wort. Sie liest das Wort leise für sich, schüttelt den EDU:BIT und startet damit den Timer (1 Minute).
- Dann zeichnet die Person Bilder auf die Tafel und die anderen aus Team A müssen raten. Sprechen oder Zeichen geben ist nicht erlaubt!
- Für jeden erratenen Begriff kriegt Team A einen Punkt (drücke Knopf A oder den gelben Knopf) bevor die Zeit abgelaufen ist.
- Dann kann Team B einmal raten um den Punkt zu stehlen.
- Beide Teams wechseln sich beim Ziehen der Karte ab, bis das Spiel vorbei ist.
- Das Team mit den meisten Punkten gewinnt!

HINWEIS!

Scanne hier

Scanne den QR-Code um Karten mit Begriffen zum Ausdrucken herunterzuladen. Wenn dir Zeichnen nicht gefällt, kannst du auch Scharade spielen. Es gelten die gleichen Regeln, aber statt zu zeichnen, werden die Begriffe pantomimisch dargestellt. Viel Spaß!

KAPITEL 5

Der digitale Würfel~

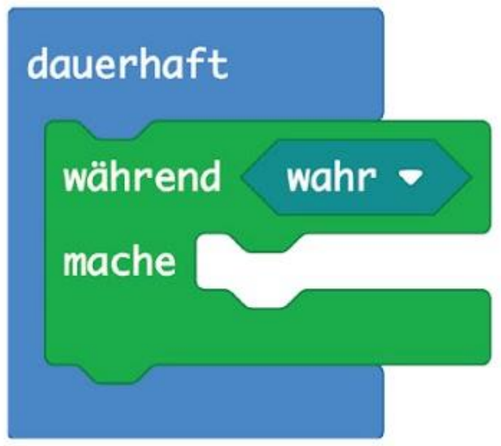
IR Bit

Scanne mich !

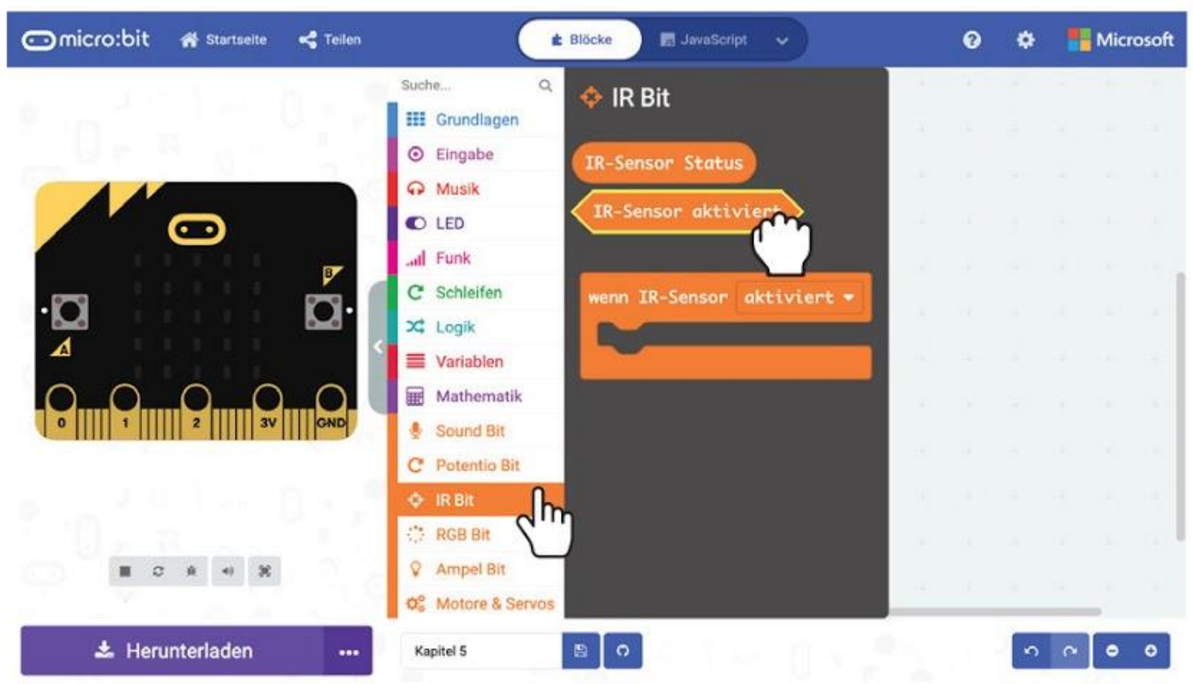
link.cytron.io/edubit-chapter-5

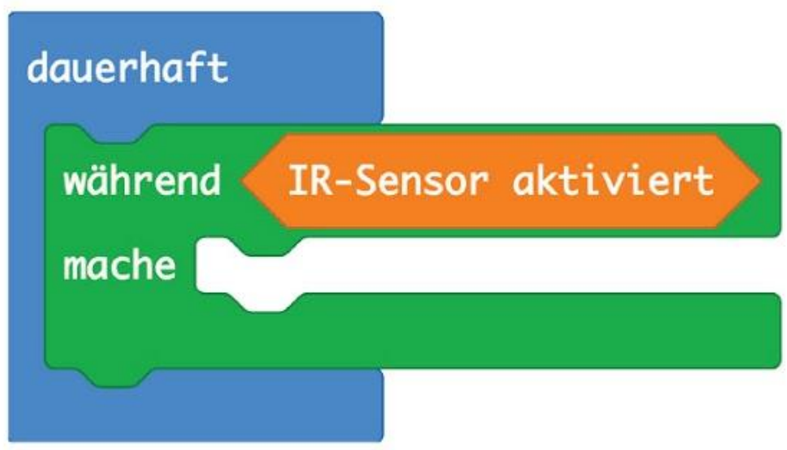
LASS UNS PROGRAMMIEREN!

Schritt 1 Erstelle ein neues Projekt im MakeCode Editor und füge die EDU:BIT-Erweiterung hinzu. Klicke auf die Gruppe **[Schleifen]** und wähle den Block **[während _ mache]**. Lege den Block in den **[dauerhaft]**-Block.

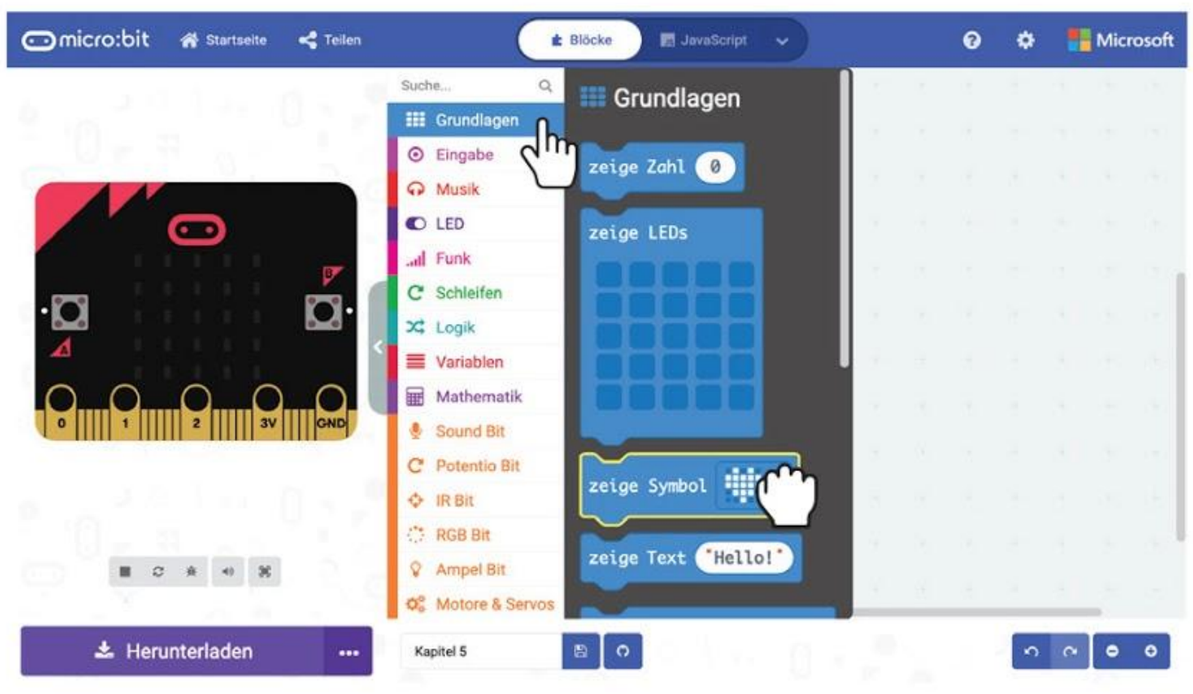


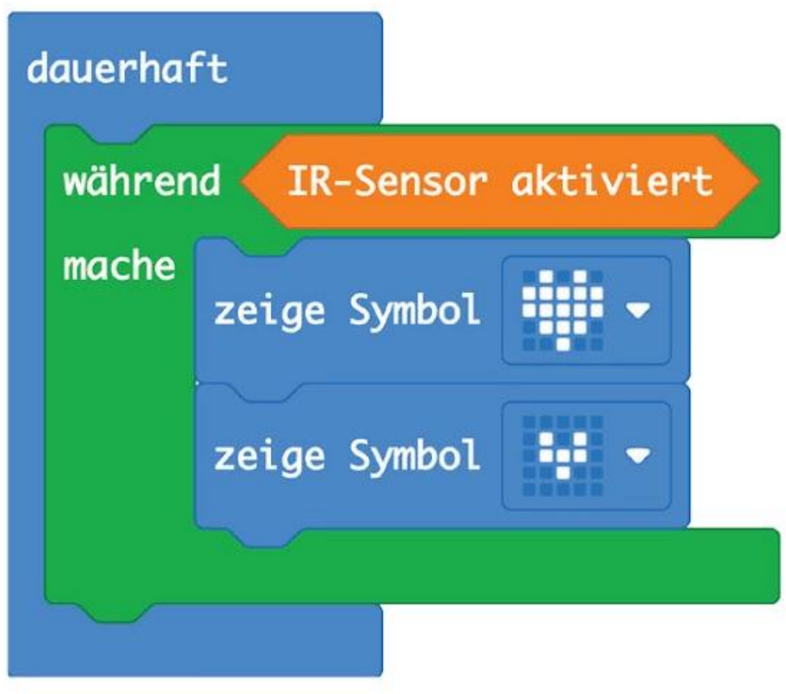
Schritt 2 Klicke auf die Gruppe **[IR Bit]** und wähle den Block **[IR-Sensor aktiviert]**. Lege diesen Block in den Vergleichs-Platz im **[während _ mache]**-Block.



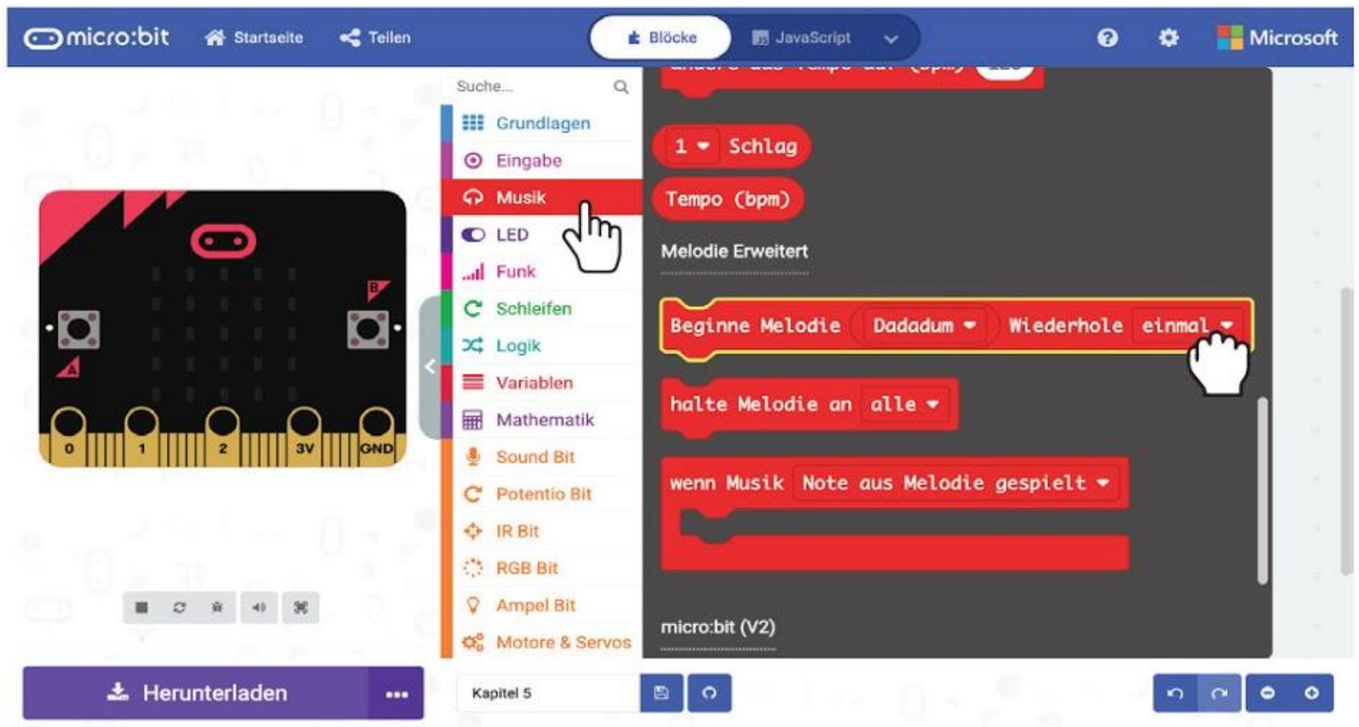


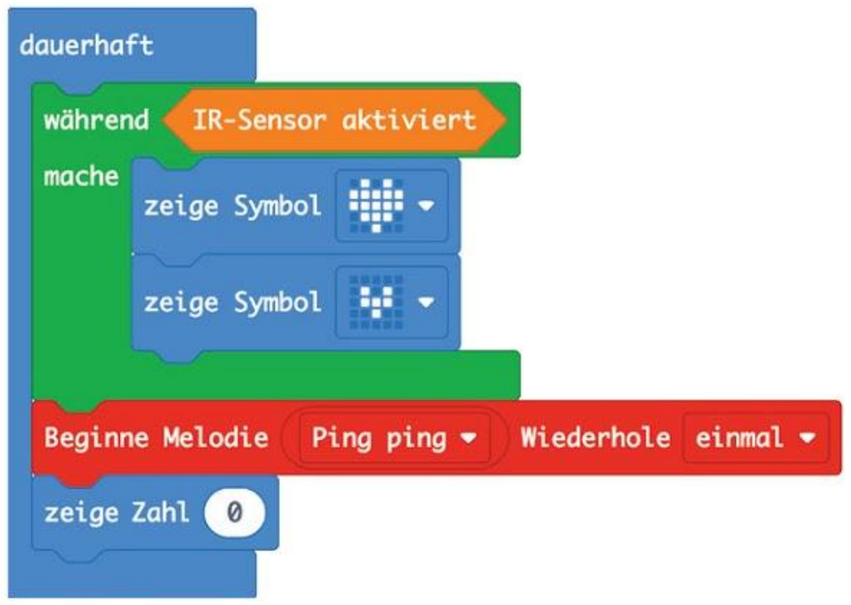
Schritt 3 Füge zwei **[zeige Symbol]**- Blöcke aus dem Bereich **[Grundlagen]** ein. Ändere ein Symbol zu “kleines Herz”. Lege beide Blöcke in **[während _ mache]**.



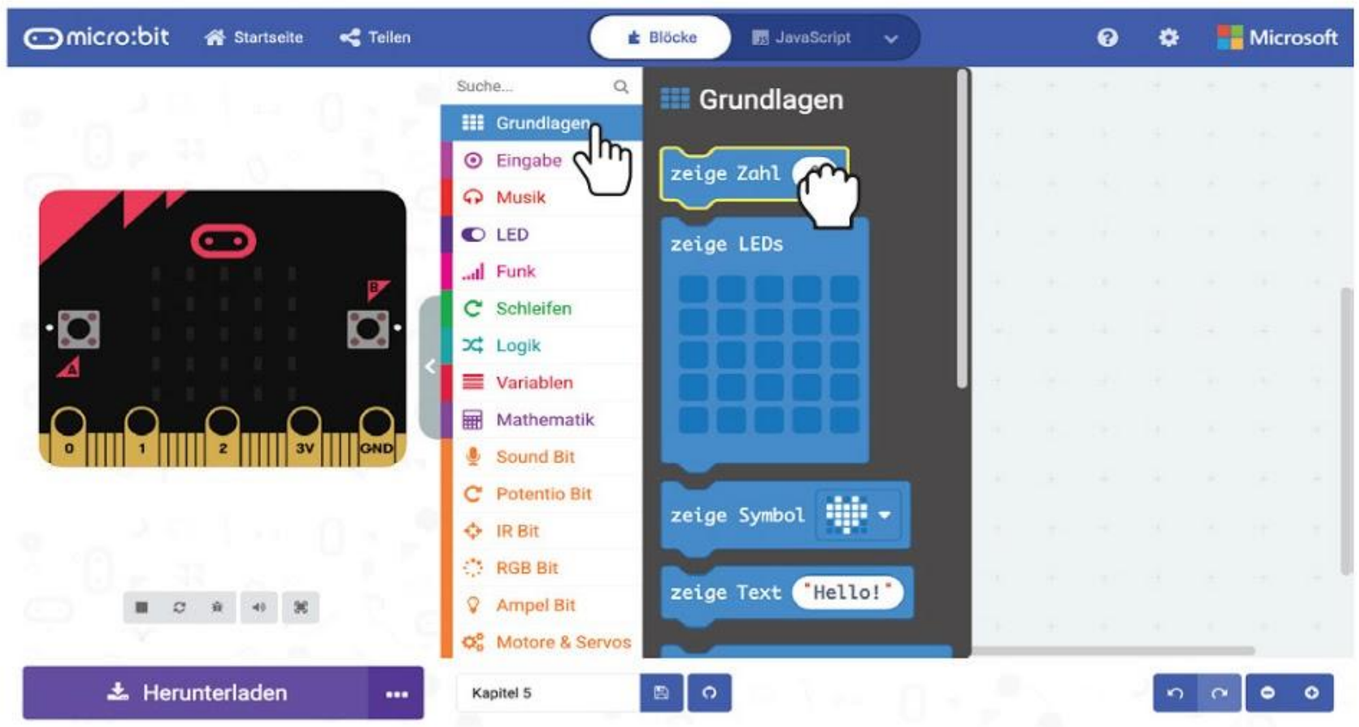


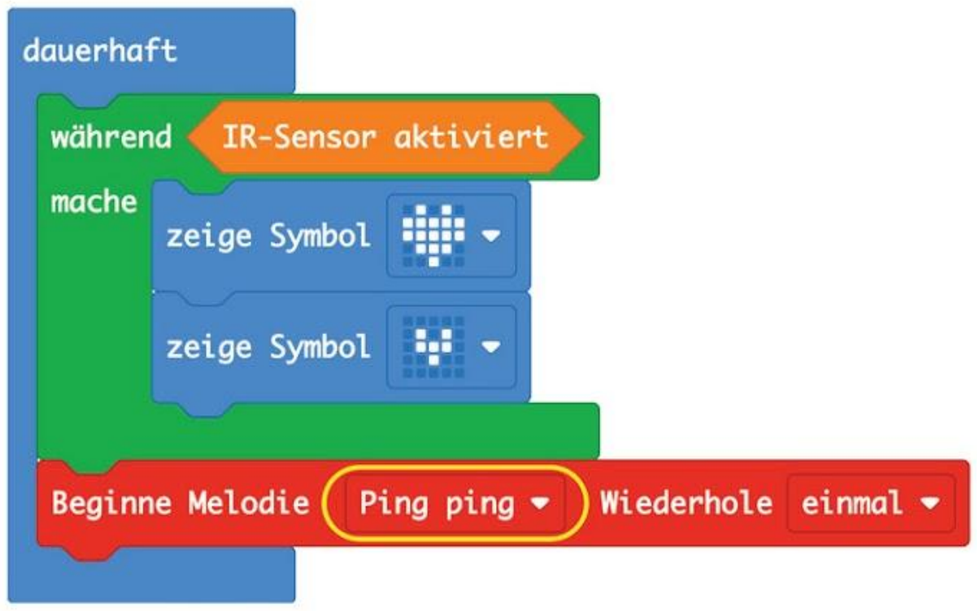
Schritt 4 In der Gruppe **[Musik]** wähle den **[Beginne Melodie _ Wiederhole _]**-Block. Ändere die Melodie “Dadadum” zu “**Ping ping**”.



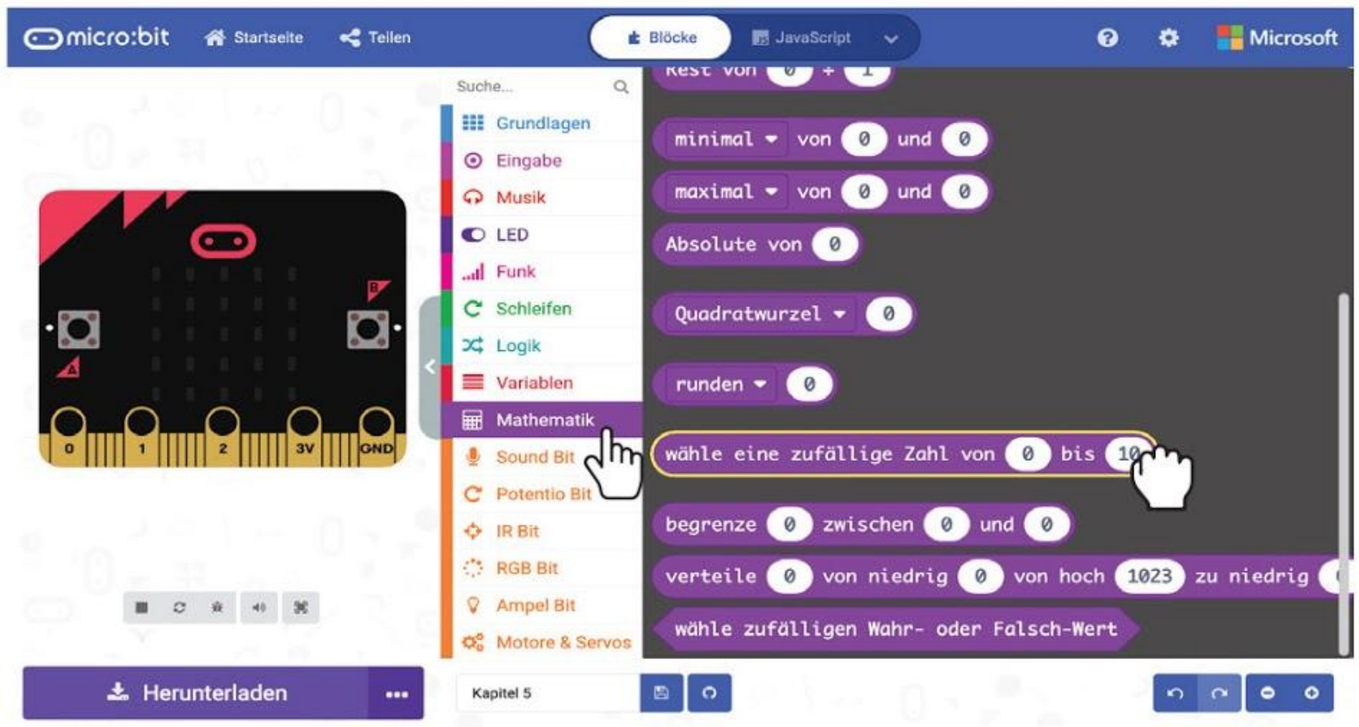


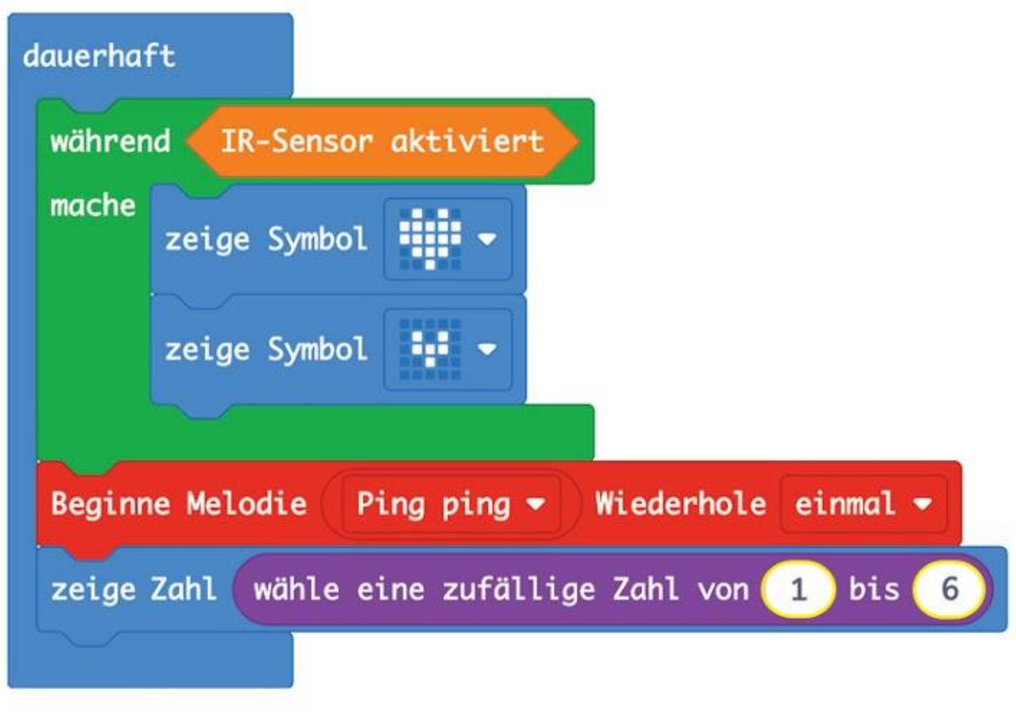
Schritt 5 Klicke in der Gruppe **[Grundlagen]** auf den Block **[zeige Zahl]**.



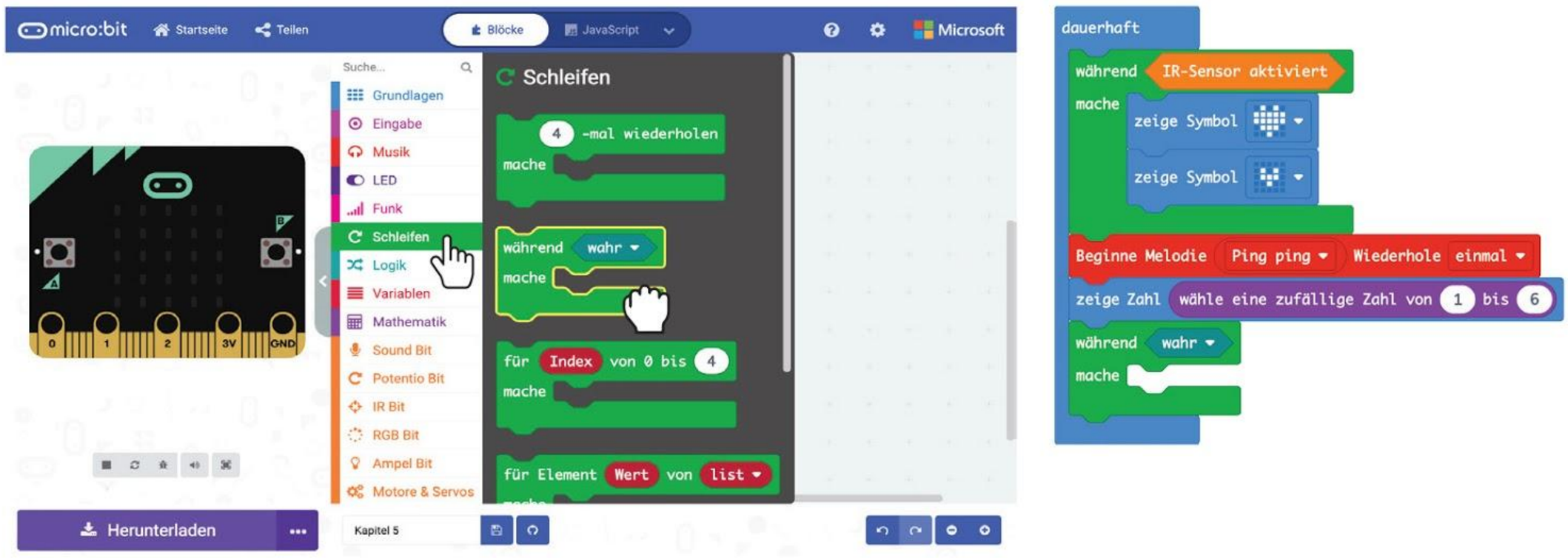


Schritt 6 Klicke auf die Gruppe **[Mathematik]** und dann auf den Block **[wähle eine zufällige Zahl von _ bis _]**. Ändere die Zahlen auf **1** und **6**.

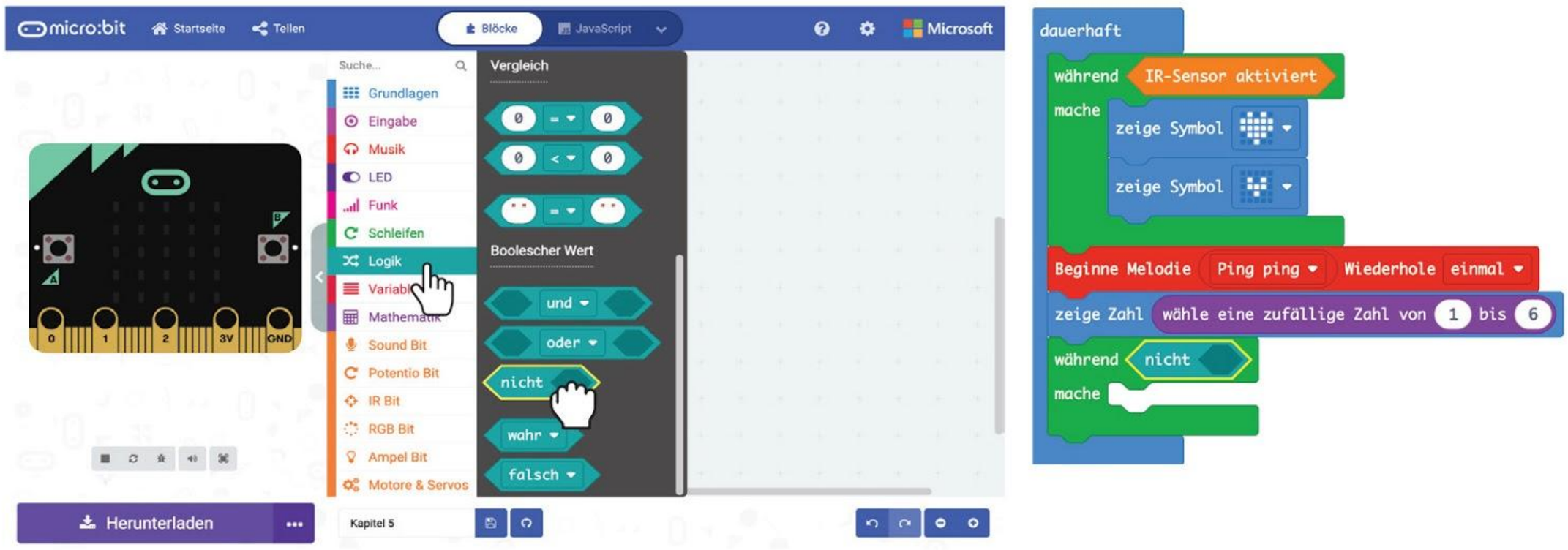




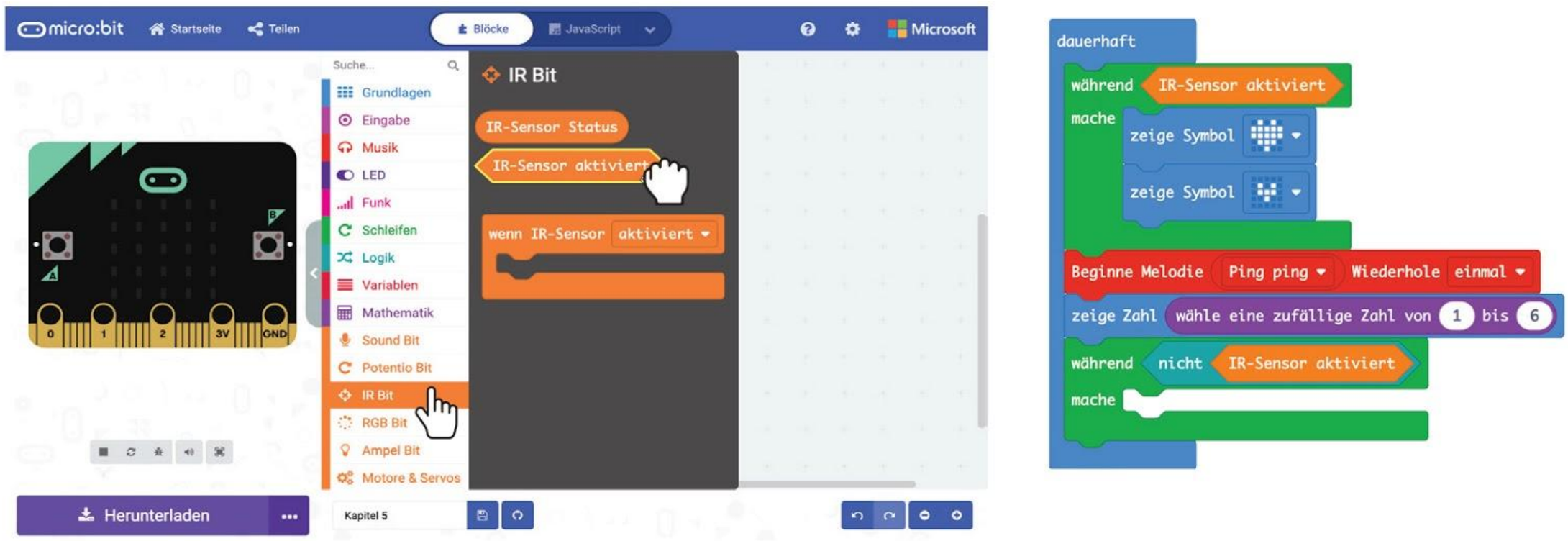
Schritt 7 Klicke in der Gruppe **[Schleifen]** auf den Block **[während _ mache]**.



Schritt 8 Klicke auf die Gruppe **[Logik]** und auf den Block **[nicht _]**. Ziehe den Block in den Vergleichs-Bereich im **[während _ mache]**-Block.



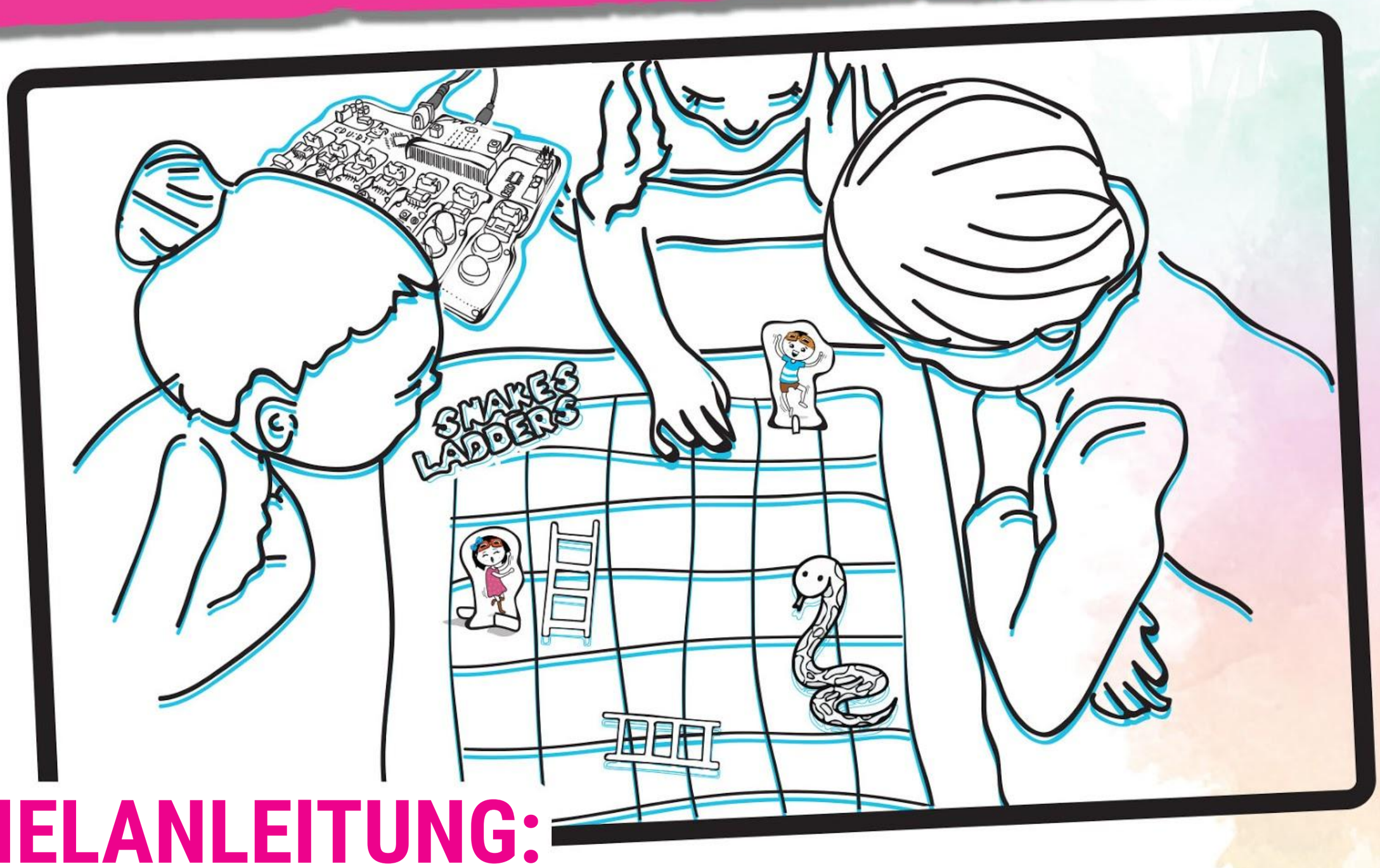
Schritt 9 Klicke auf die Gruppe **[IR Bit]** und klicke auf den Block **[IR-Sensor aktiviert]**. Ziehe den Block in den freien Bereich im **[nicht _]**-Block.



Schitt 10 Lade den Code auf deinen EDU:BIT.

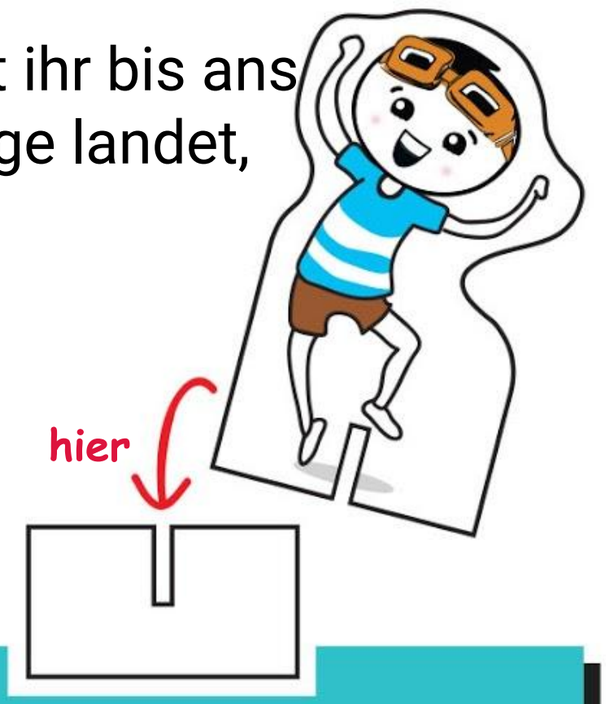
Spielen wir!

Leiterspiel



SPIELANLEITUNG:

- Alle die mitspielen, wählen eine Spielfigur aus und stellen sie auf das Feld ‘Start Here’ (Starte hier).
- Dann wird abwechselnd “gewürfelt” - halte deine Hand über das IR-Bit. Wenn du die Herz-Animation siehst, ziehe deine Hand wieder weg.
- Fahre mit deiner Spielfigur so viele Felder weiter wie auf der LED-Matrix angezeigt werden.
- Wenn deine Spielfigur am Fuß einer Leiter landet, darfst du mit ihr bis ans Ende der Leiter springen. Wenn sie auf dem Kopf einer Schlange landet, musst du bis zum Schwanz der Schlange hinunter rutschen.
- Wer zuerst 100 erreicht, gewinnt. Viel Spaß!



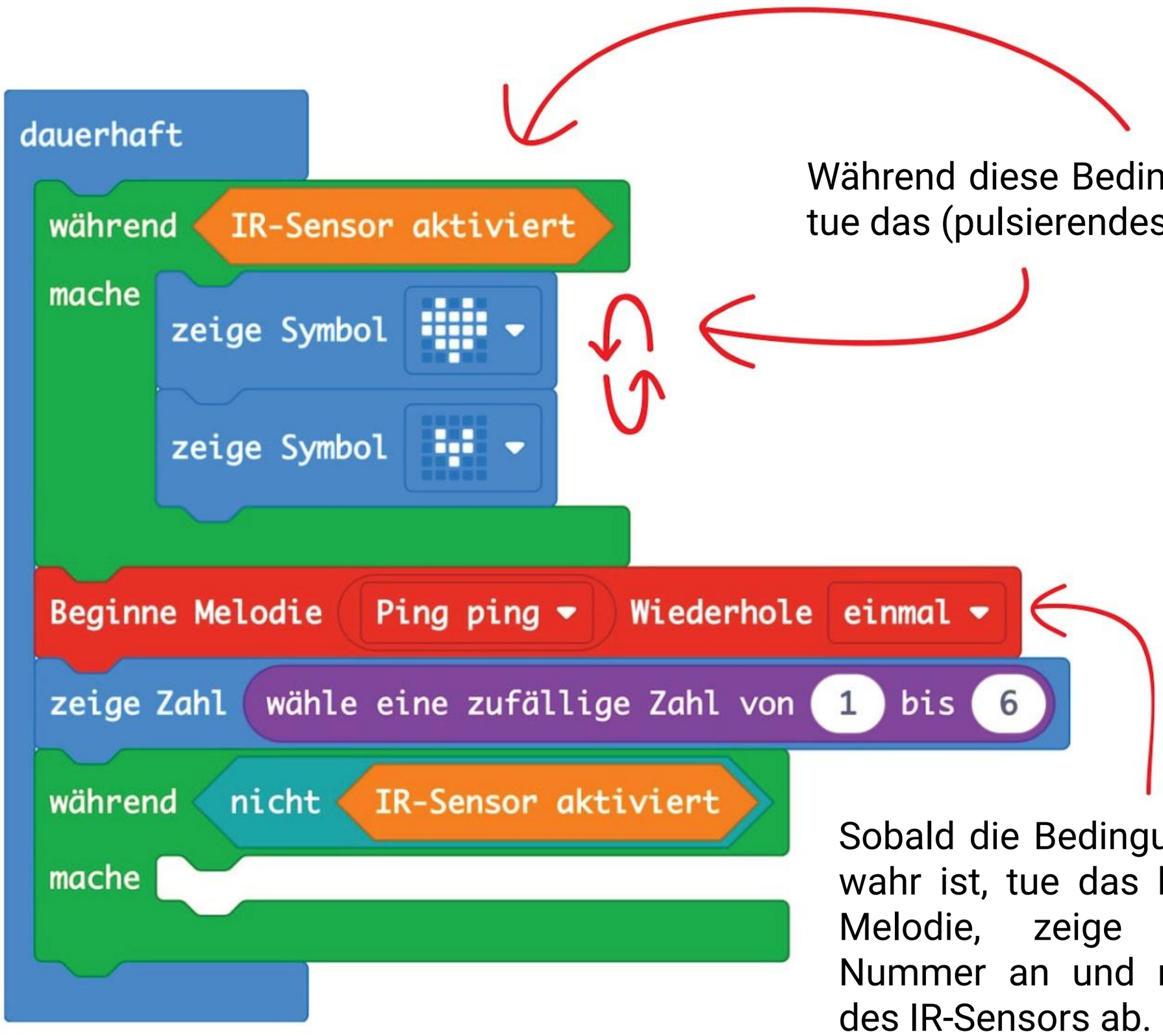
HINWEIS!

Das Spielbrett und die Spielfiguren findest du in der EDU:BIT-Box. Drücke die Figuren und die Standfüße aus dem Karton und schiebe sie so zusammen.

KNACKE DEN CODE

Wir haben den Block **[während _ mache]** *aus der Gruppe* **[Schleifen]** *vorhin in unserem Code benutzt. Weißt du wie eine während-Schleife funktioniert?*

Wenn ein Programm einen **[während _ mache]**-Block ausführt, überprüft es zuerst die Bedingung. Solange die Bedingung WAHR ist, werden die Blöcke im **[während _ mache]**-Block ausgeführt. Sobald die Bedingung FALSCH wird, verlässt das Programm die Schleife und führt die Blöcke dahinter aus.

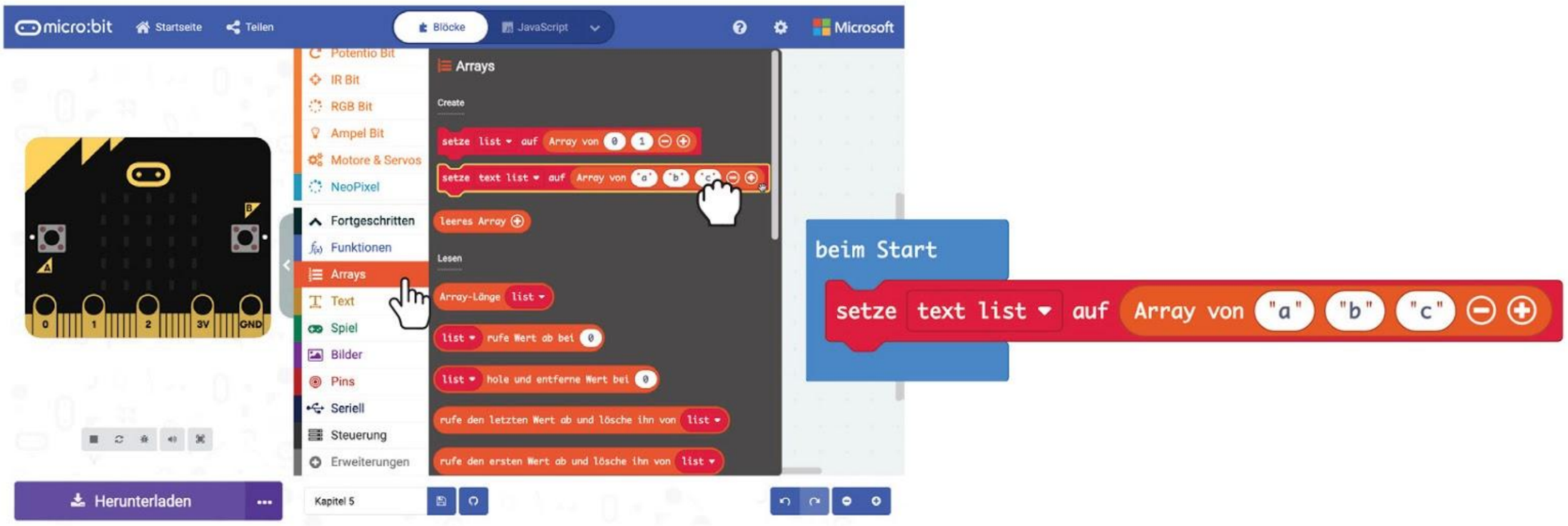


Während diese Bedingung wahr ist, tue das (pulsierendes Herz).

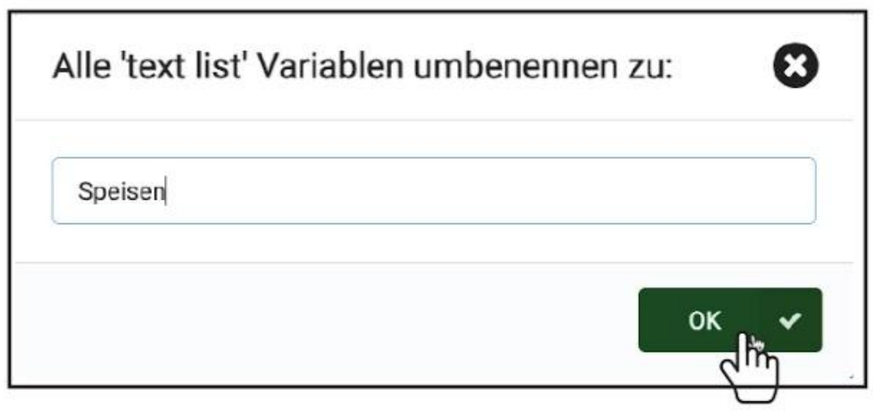
Sobald die Bedingung NICHT mehr wahr ist, tue das hier (spiele eine Melodie, zeige eine zufällige Nummer an und rufe den Status des IR-Sensors ab.

Anstatt EDU:BIT als digitalen Würfel zu benutzen, kannst du auch das letzte Programm so verändern, dass es dir hilft, zwischen verschiedenen verlockenden Möglichkeiten zu wählen, wenn du dich einmal nicht entscheiden kannst - zum Beispiel, was du zu Mittag essen willst... mjam mjam~

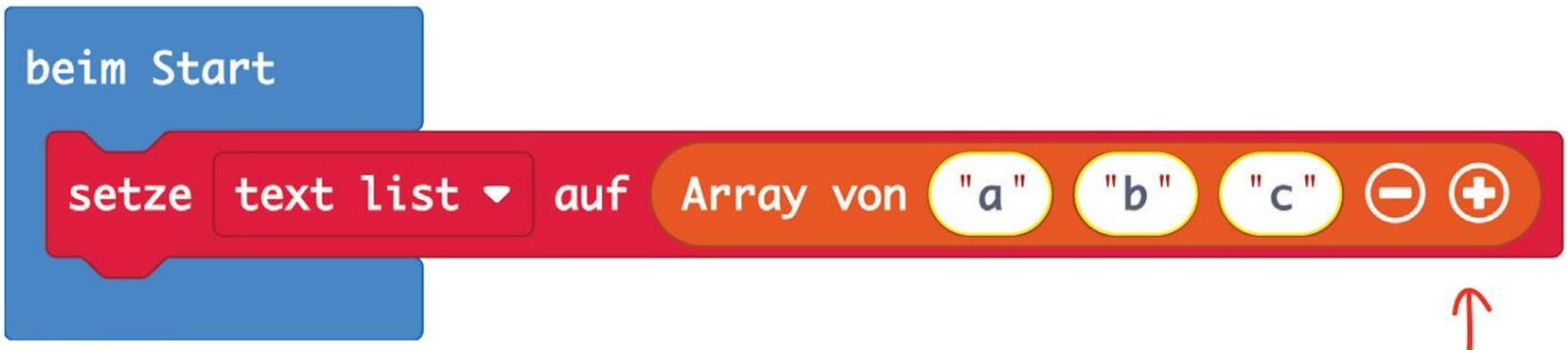
Schritt 11 Klicke in der Kategorie **[Fortgeschritten] : [Array]** auf den Block **[setze text list auf Array von _ _ _]**. Ziehe diesen in den **[Grundlagen] : [beim Start]**-Block.



Schritt 12 Klicke im Block auf **[text list]** und wähle "**Variable umbenennen**". Tippe "**Speisen**" ein und klicke auf OK.

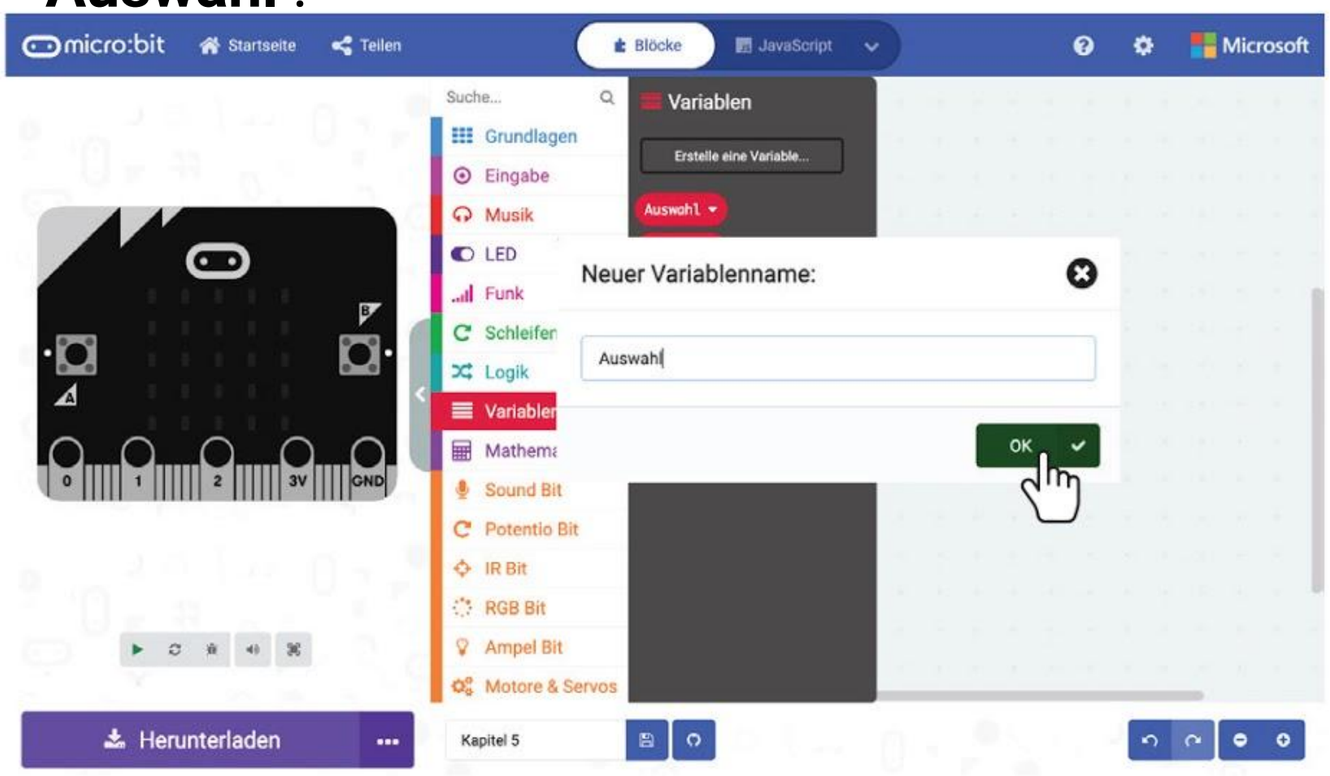


Schritt 13 Klicke im Array-Block nacheinander auf die Textfelder und tippe in jedes eine Speise ein.

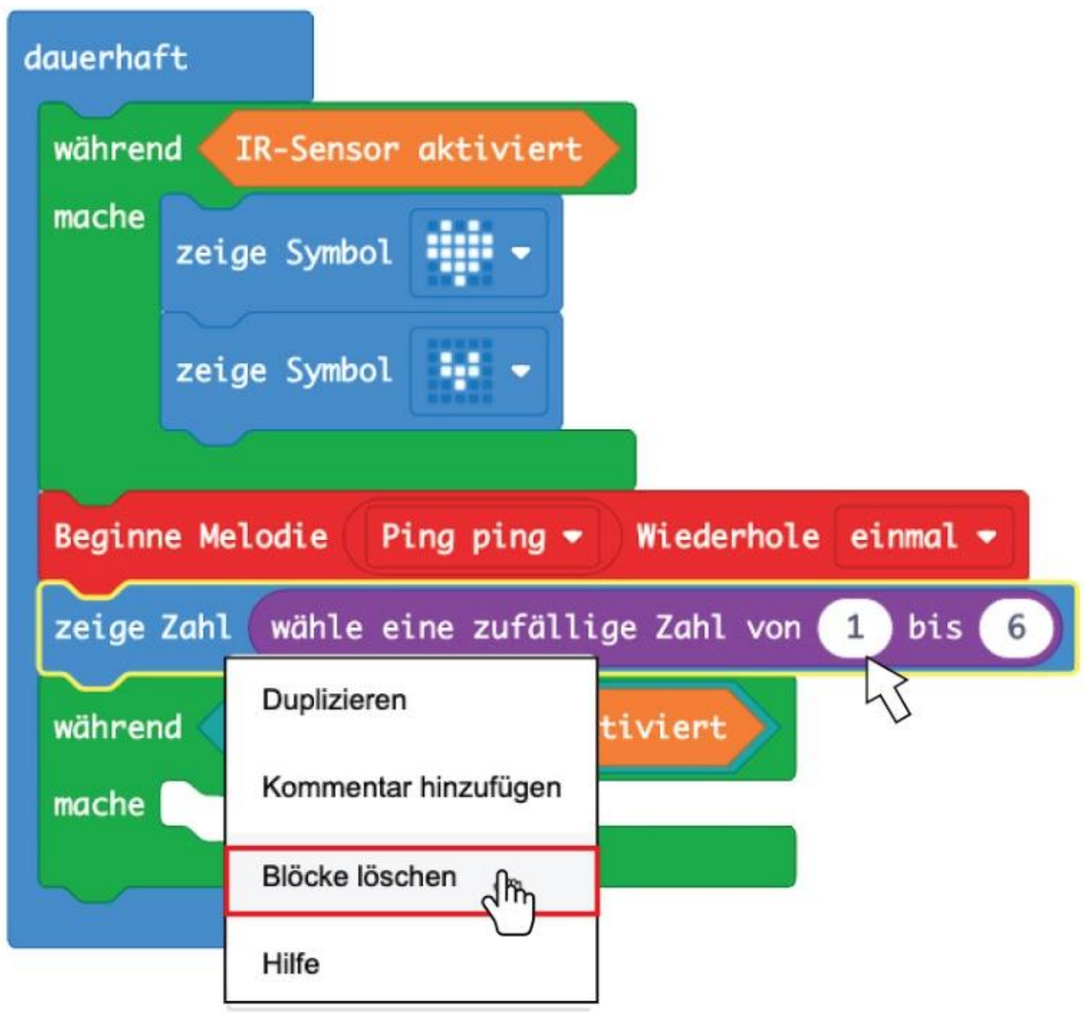


KAPITEL 5 : Der digitale Würfel~

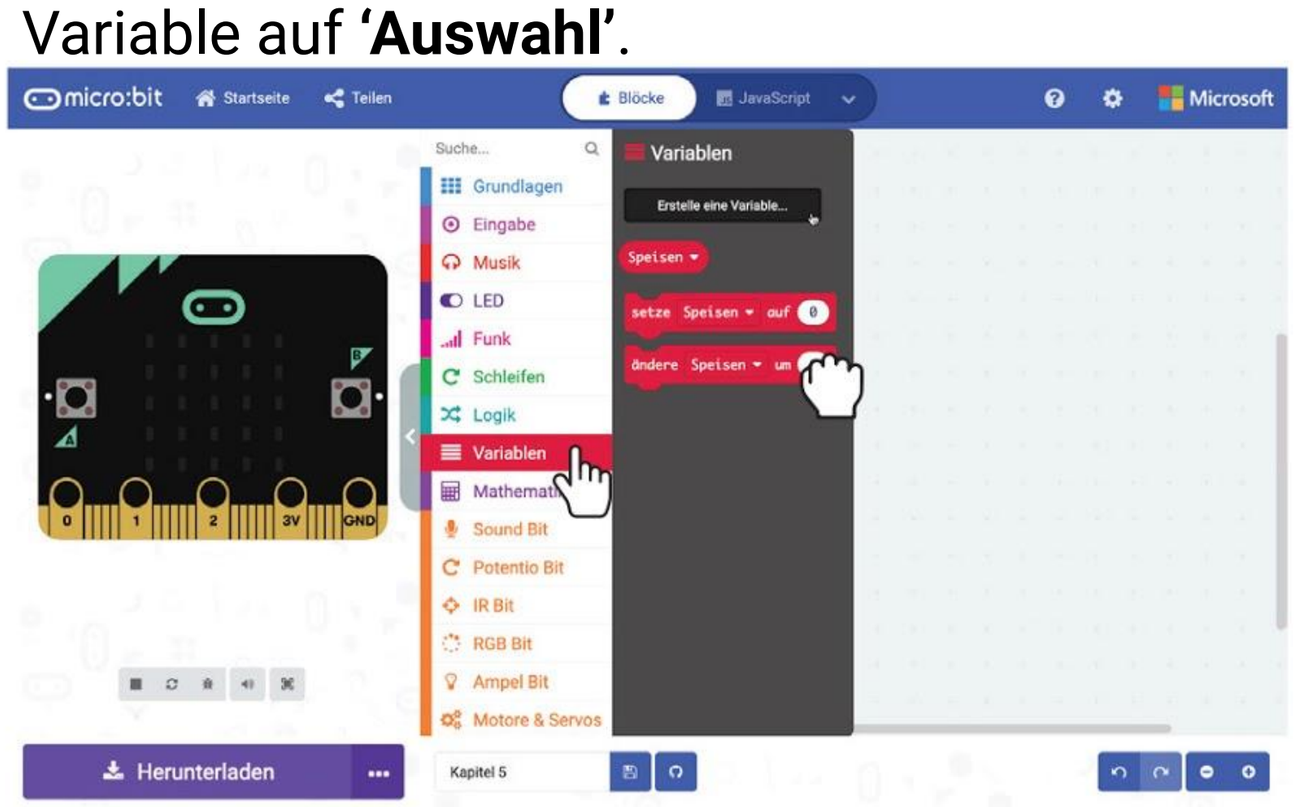
Schritt 14 Klicke auf **[Variablen]** und erstelle eine neue Variable namens **‘Auswahl’**.

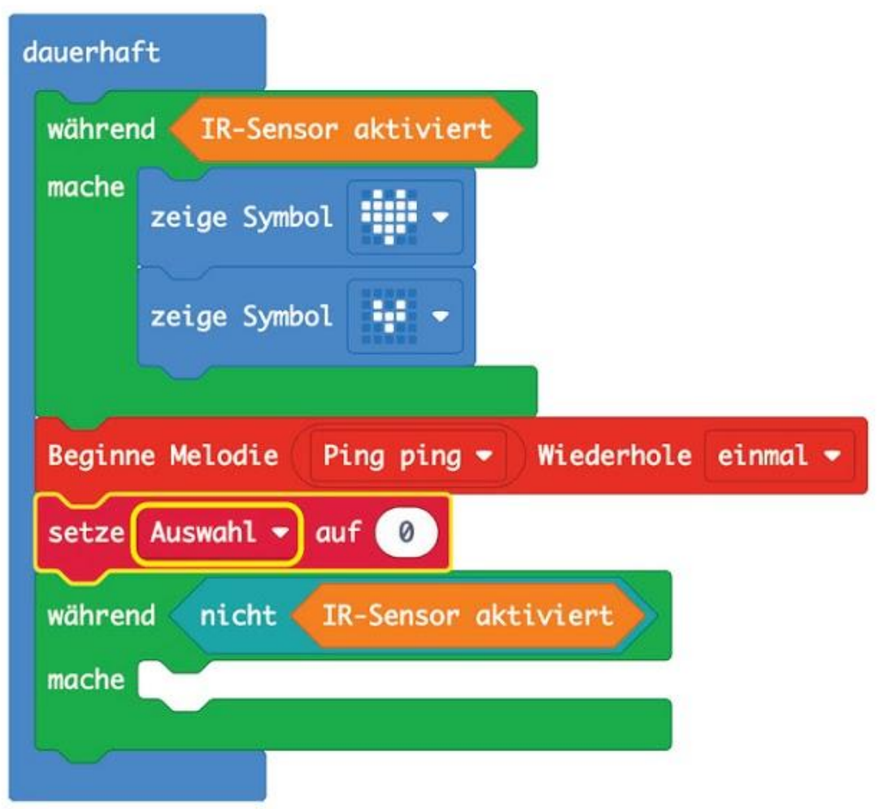


Schritt 15 Klicke mit der rechten Maustaste auf **[zeige Zahl [wähle eine zufällige Zahl von _ bis _]]** und klicke auf ‘**Blöcke löschen**’.

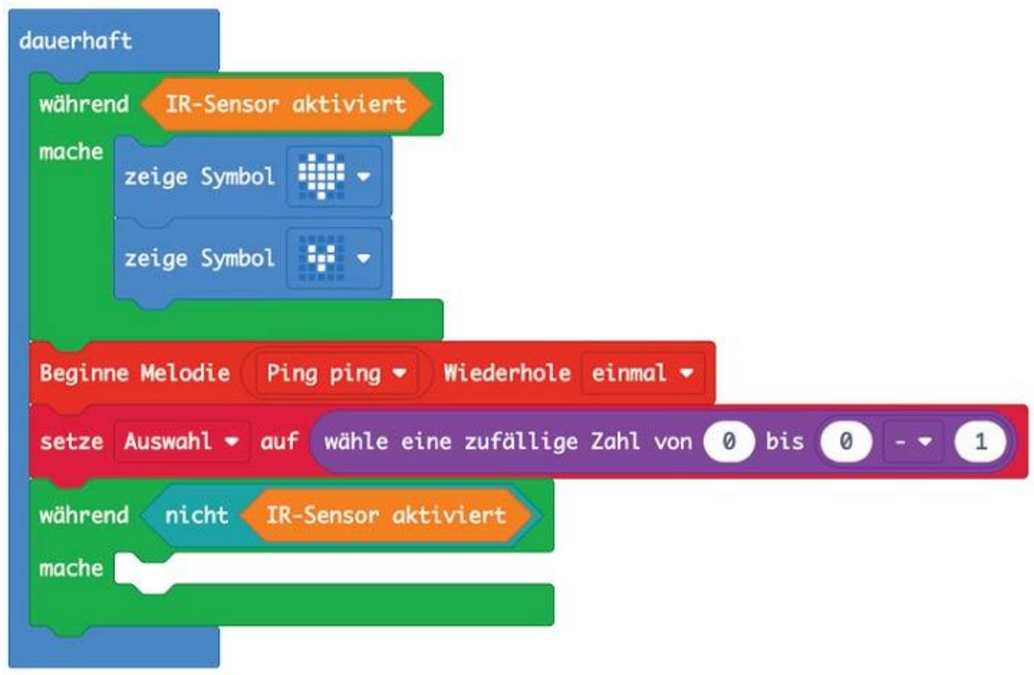


Schritt 16 Klicke in der Gruppe **[Variablen]** auf **[setze _ auf _]**. Lege diesen Block zwischen **[Beginne Melodie _ Wiederhole _]** und **[während _ mache]**. Ändere die Variable auf **‘Auswahl’**.

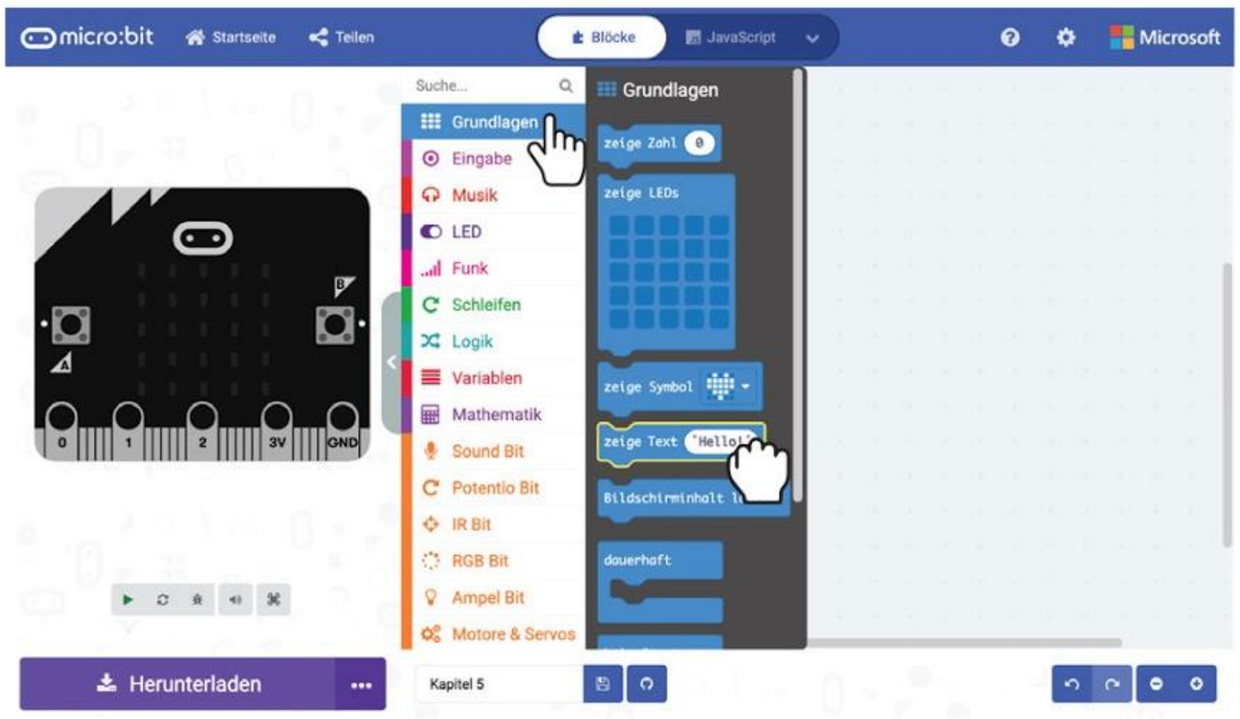


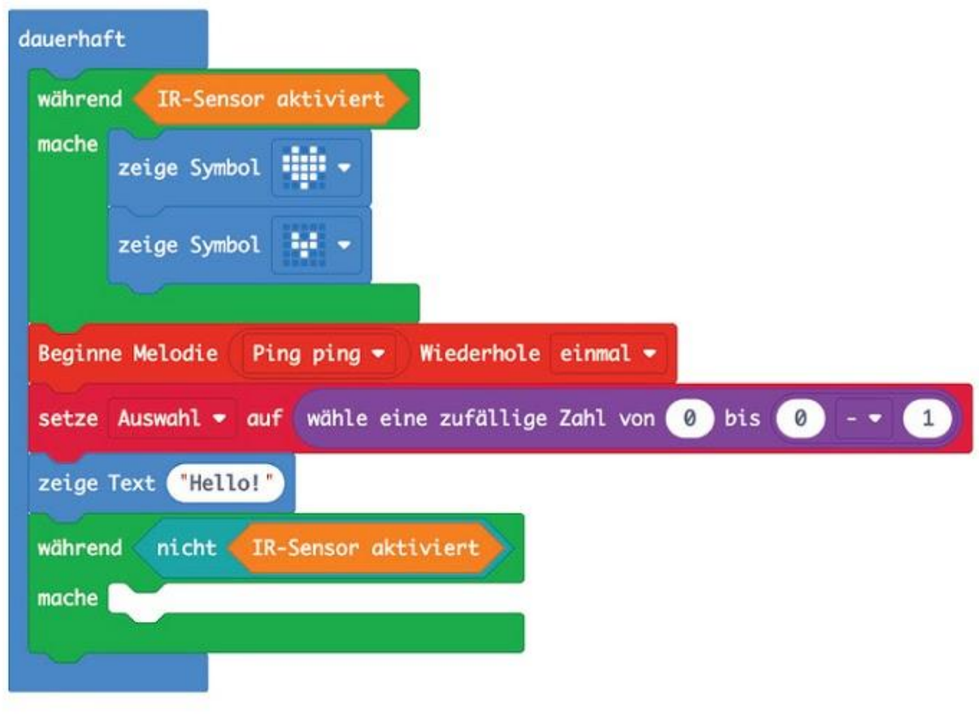


Schritt 17 Klicke auf die Gruppe **[Mathematik]** und füge die Blöcke **[wähle eine zufällige Zahl von _ bis _]** und **[_ - _]** hinzu. Ändere die letzte Zahl zu **1**.

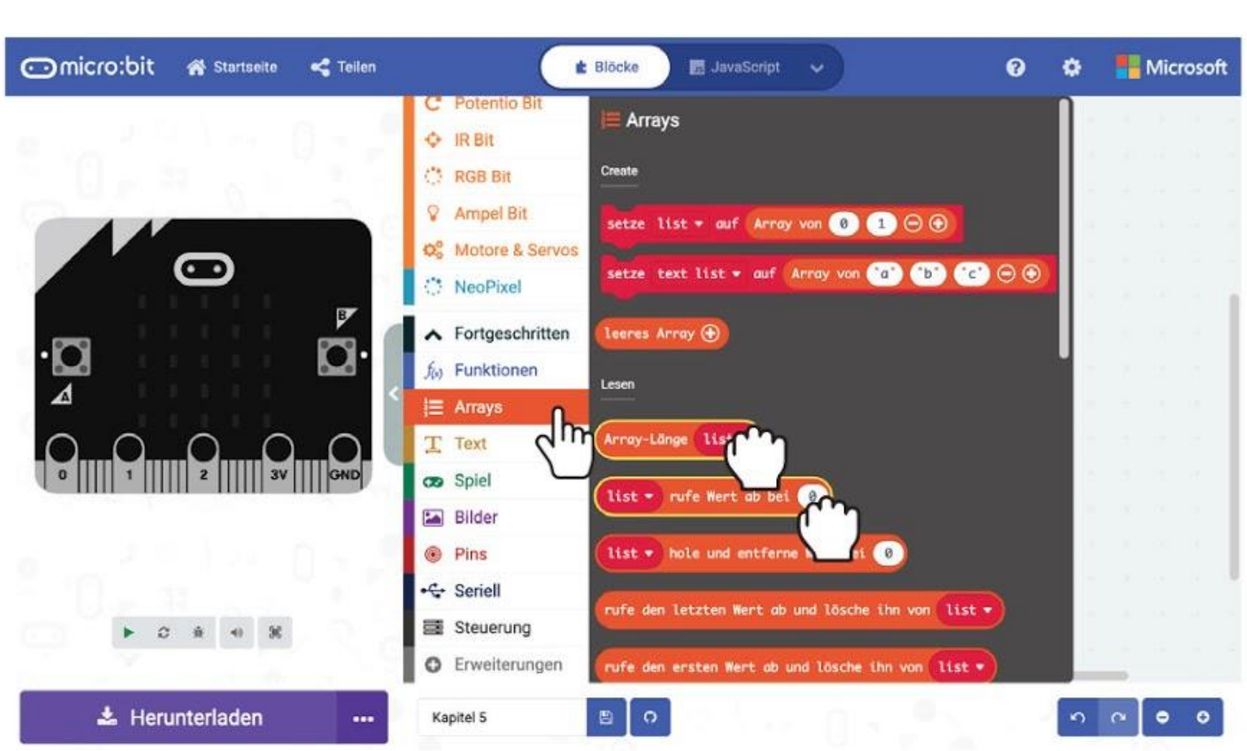


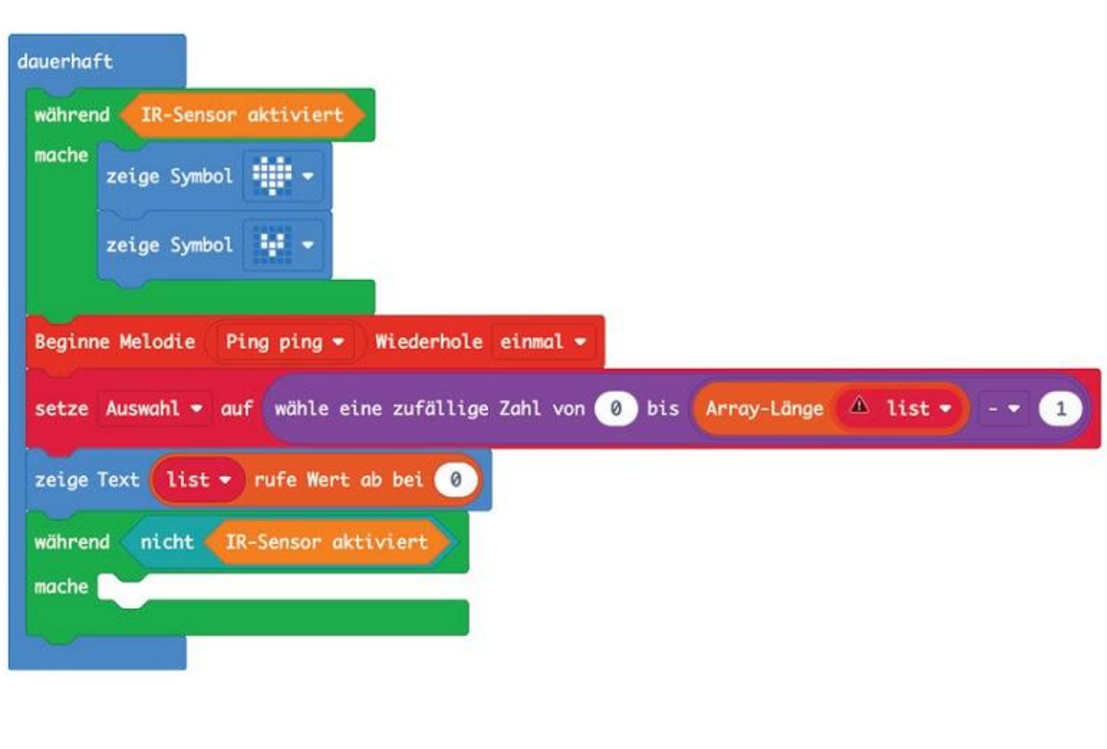
Schritt 18 Klicke auf **[Grundlagen]** und auf den Block **[zeige Text]**.



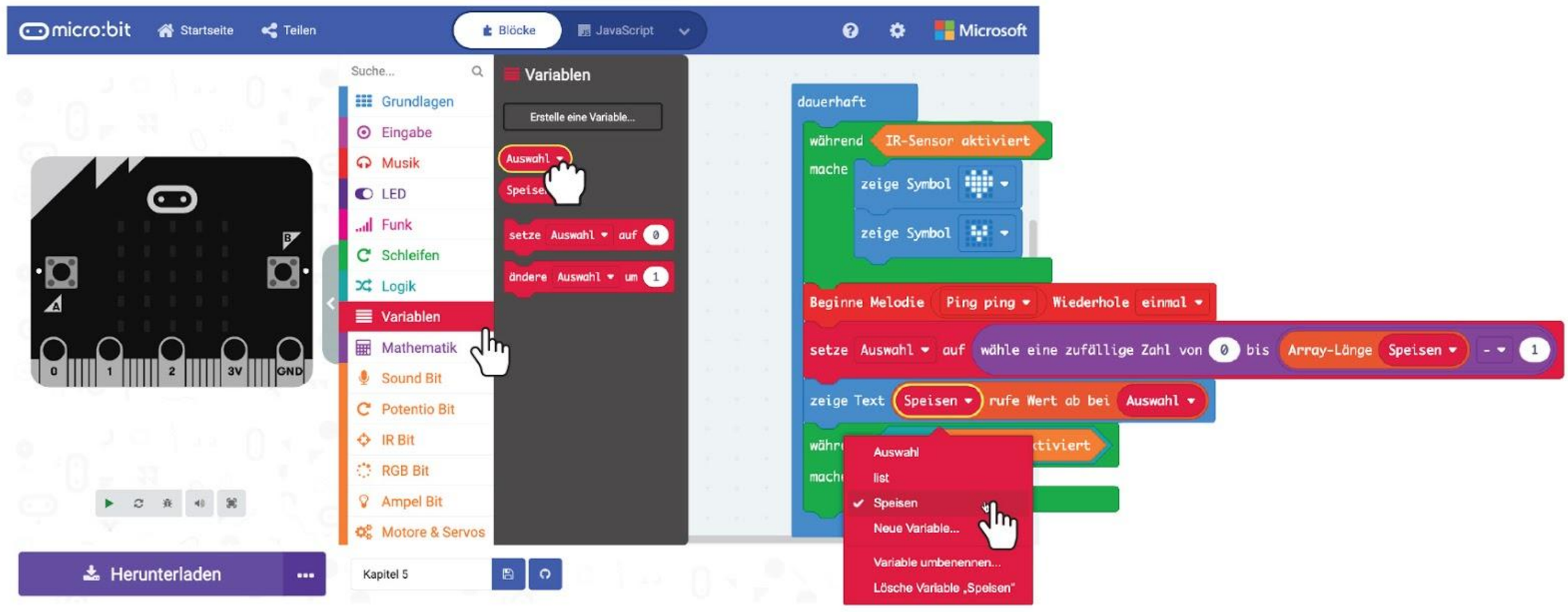


Schritt 19 Klicke auf die Gruppe **[Fortgeschritten] : [Array]** und füge die Blöcke **[Array-Länge]** und **[_ rufe Wert ab bei _]** zu deinem Code hinzu.

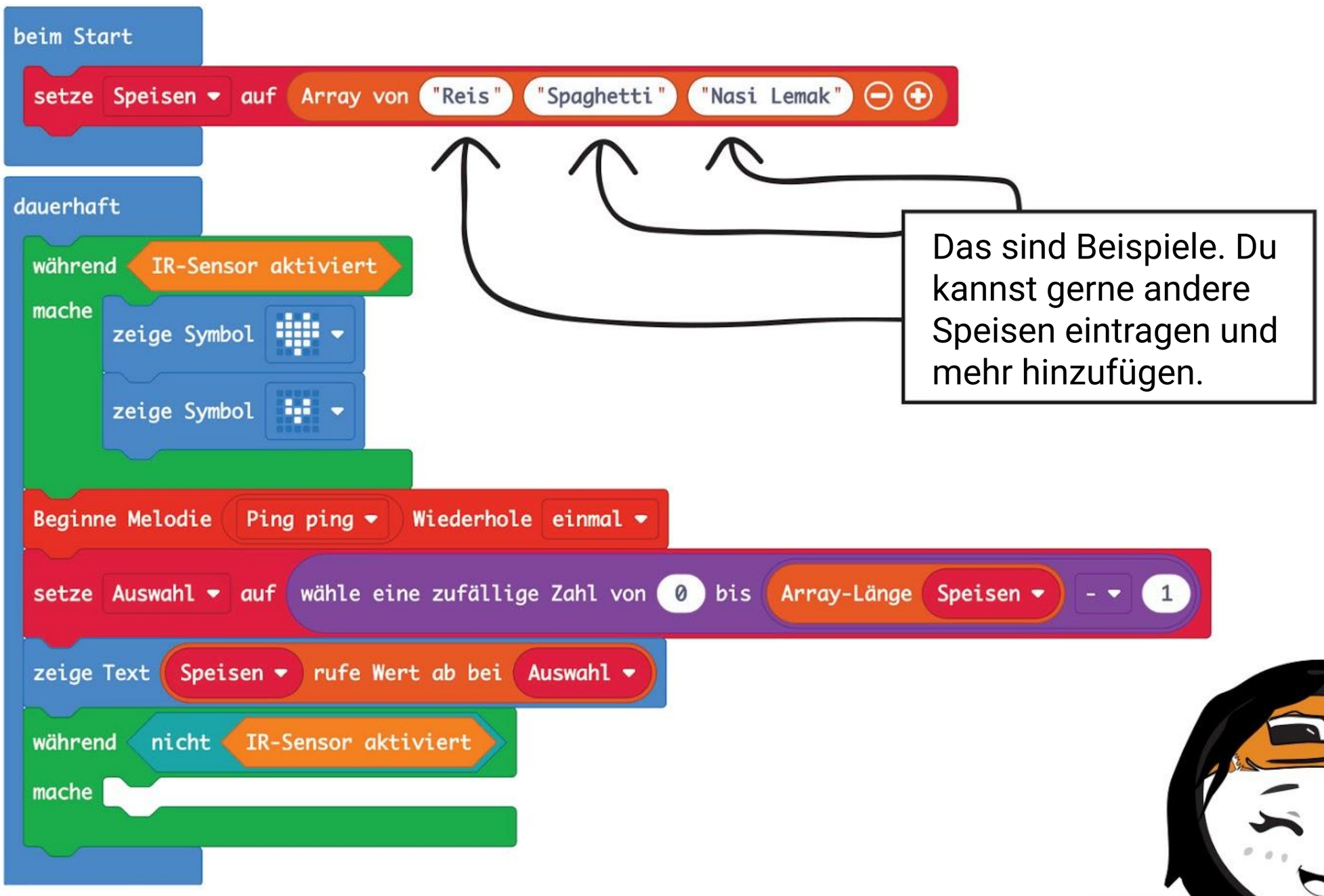




Schritt 20 Klicke auf **[list]** und ändere in beiden Blöcken die Variable zu '**Speisen**'. Zuletzt klicke auf **[Variablen]** und wähle den Block **[Auswahl]**. Ziehe ihn in den leeren Platz im Block **[_ rufe Wert ab bei _]**.



Hier ist der vollständige Code von "Was gibt's zu Mittag?":

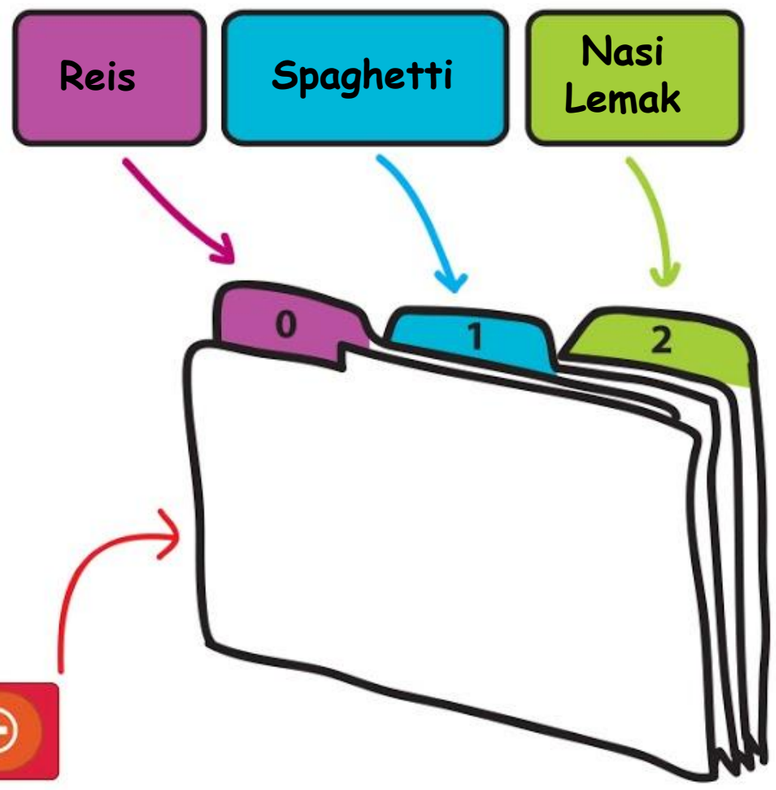


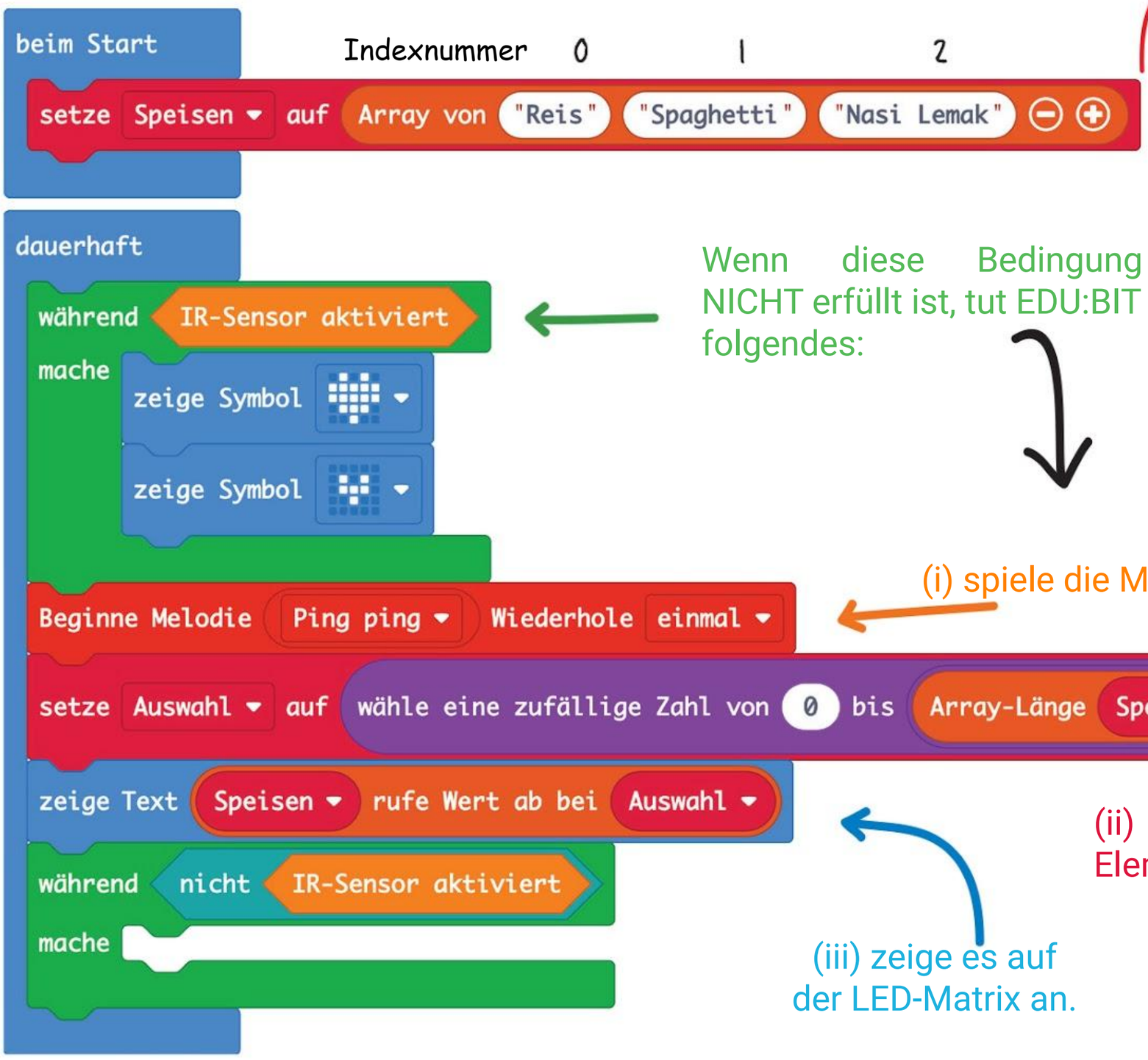
Das nächste Mal, wenn du nicht weißt, was du essen sollst, kann EDU:BIT dir bei der Entscheidung helfen, indem du IR Bit auslöst und dann deine Hand wegziehst. Du kannst dieses Programm auch so anpassen, dass es dir hilft zu entscheiden, was du nächstes Mal mit deinen Freundinnen und Freunden spielen willst. Was müsstest du dafür verändern?

KNACKE DEN CODE

Ein **Array** (auch Feld genannt) ist eine Liste bzw. Sammlung von gleichen Variablen. Du kannst es dir wie eine Mappe mit mehreren Unterteilungen vorstellen. In jeder Unterteilung kann eine Information gespeichert werden. Wir benutzen ein Array wenn wir in unserem Code unkompliziert eine Liste verwalten möchten.

In diesem Code haben wir ein Array mit drei Elementen erstellt und es “Speisen” genannt. Jedes Element steht dann für eine Speise und wir können es ganz einfach verändern. Wir können Elemente hinzufügen und löschen indem wir auf + und - klicken.





Wenn diese Bedingung NICHT erfüllt ist, tut EDU:BIT folgendes:

(i) spiele die Melodie ‘Ping ping’ einmal,

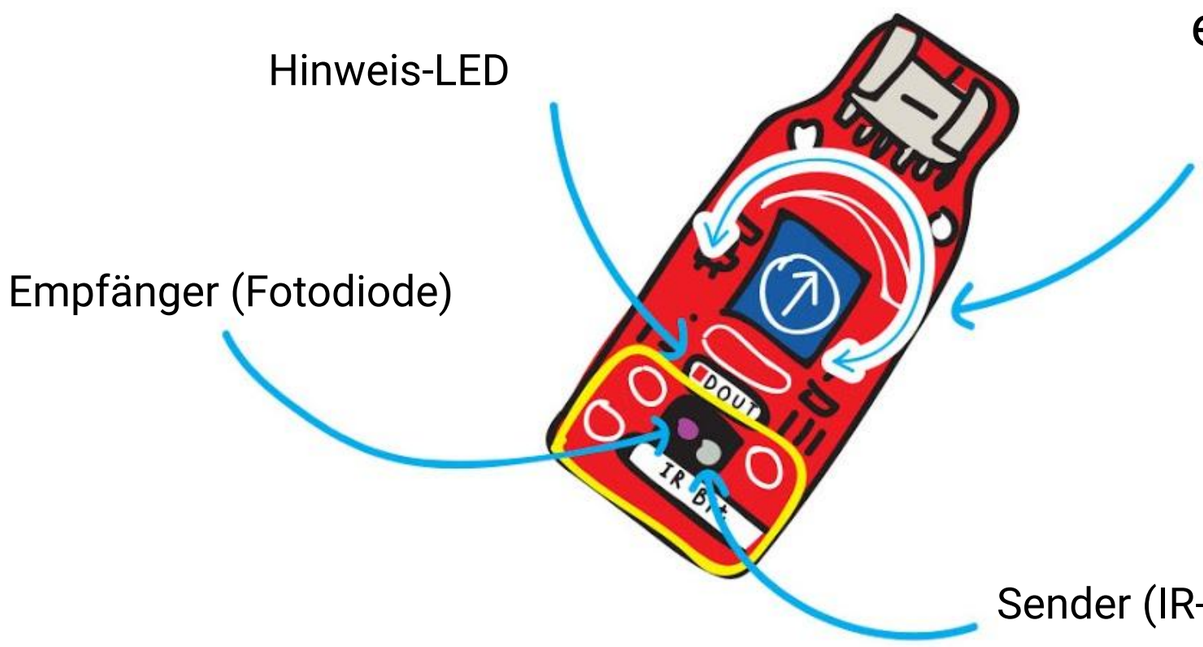
(ii) wähle zufällig per ‘Auswahl’ ein Element aus der Speisen-Liste und

(iii) zeige es auf der LED-Matrix an.

Beim Programmieren zählen wir von 0 aufwärts anstatt von 1. Also hat “Reis” die Indexnummer 0 und das letzte Element “Nasi Lemak” hat die Indexnummer 2, obwohl es das dritte Element in der Liste ist.

GUT ZU WISSEN!

Ein **Infrarotsensor (IR-Sensor)** ist ein elektronisches Gerät, das oft verwendet wird um Hindernisse zu erkennen. Er besteht aus zwei Bauteilen - einem Sender (IR-LED) und einem Empfänger (Fotodiode).

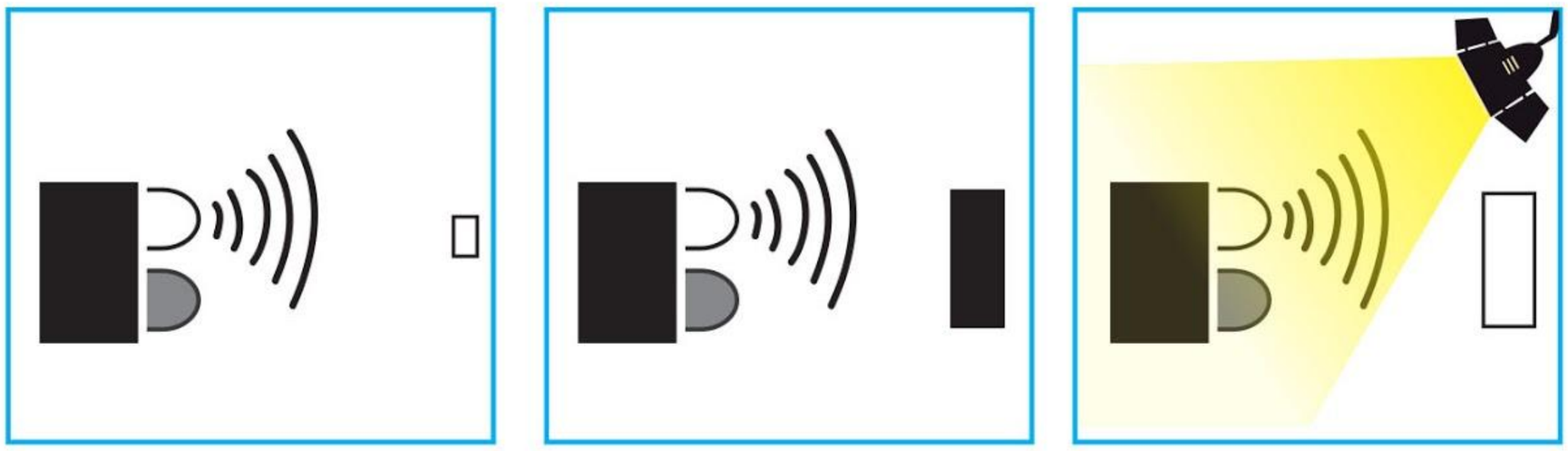


Wie funktioniert es?

Die IR-LED sendet Licht aus, das in den Empfänger zurückgeworfen wird, wenn sich etwas vor dem Sensor befindet. IR Bit wird “ausgelöst”, wenn die Menge an reflektiertem Licht höher ist als der eingestellte Schwellenwert. Wenn es ausgelöst wird, leuchtet die Hinweis-LED auf dem IR Bit.

Wenn sich nichts vor dem Sensor befindet oder das Objekt zu weit weg ist, gelangt nur wenig oder gar kein Licht in den Empfänger. Dann wird der Sensor nicht ausgelöst.

Unter den folgenden Bedingungen funktioniert der Infrarotsensor möglicherweise nicht richtig:



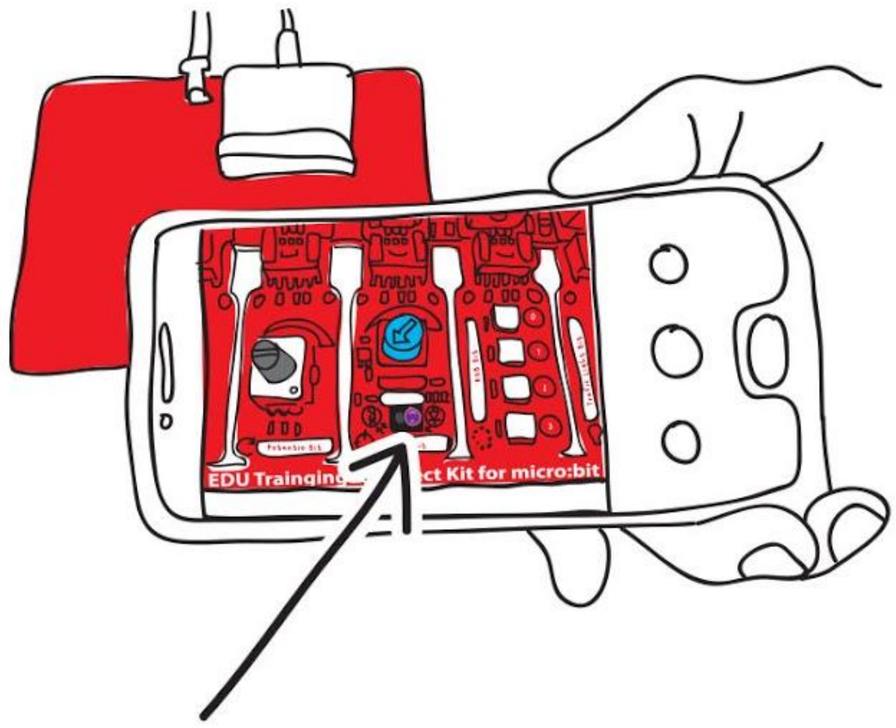
Das Objekt ist zu klein

Das Objekt hat eine dunkle oder schwarze Oberfläche

Zu viel direktes Licht

Wusstest du?

Infrarotlicht ist mit bloßem Auge unsichtbar, aber du kannst es sichtbar machen, indem du die IR-LED durch die Kamera eines Smartphones anschaust.

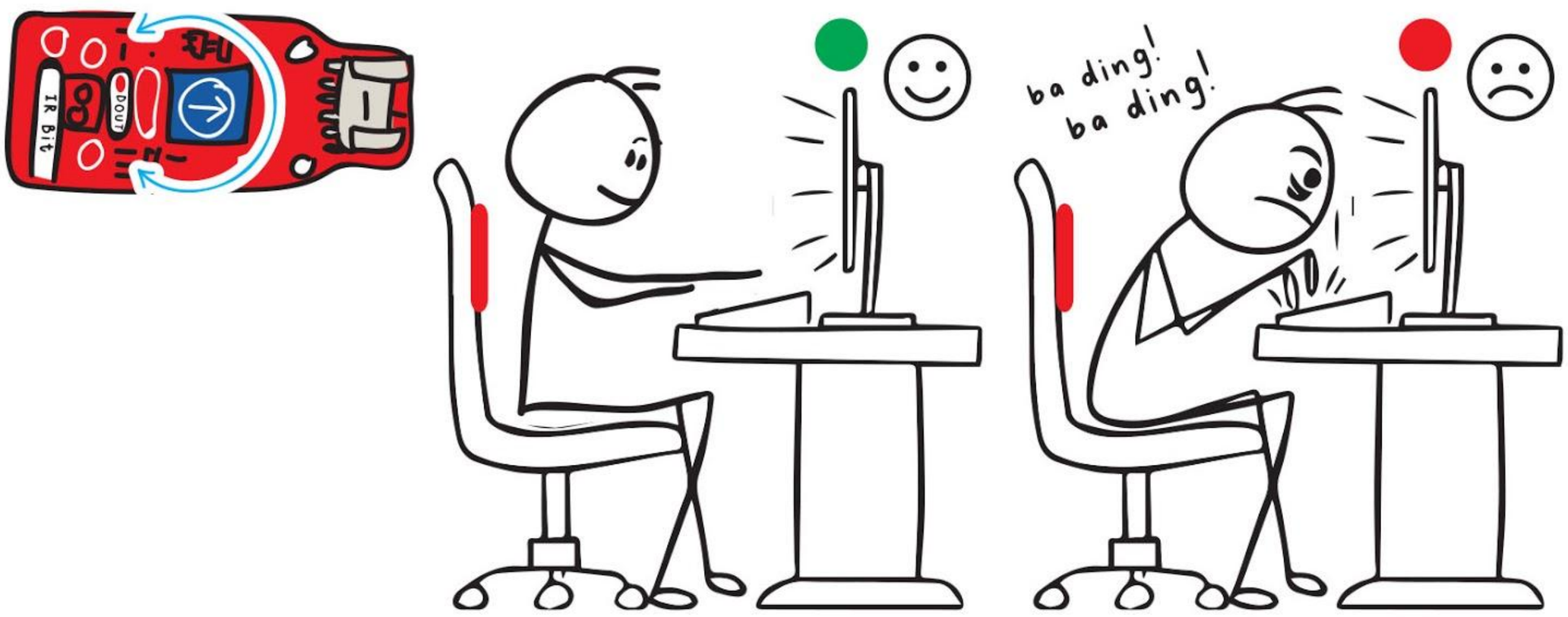


HERAUSFORDERUNG

Mache aus deinem EDU:BIT einen Anti-Schlapp-Detektor.	
beim Start	Scrolle “Achte auf deine Haltung”
IR-Sensor aktiviert	Zeige einen Smiley auf der LED-Matrix und aktiviere die grüne LED.
IR-Sensor nicht aktiviert	Spiele die Melodie “Ping ping” einmal. Zeige ein trauriges Gesicht auf der LED-Matrix und lass die rote LED blinken.

So funktioniert es:

Bringe EDU:BIT an deinem Stuhl an wie gezeigt. Sitze gemütlich mit einer guten Körperhaltung. Drehe am blauen Knopf auf dem IR Bit bis die Hinweis-LED aufleuchtet (IR Bit erkennt deinen Rücken). Dieser Vorgang heißt Kalibrierung.

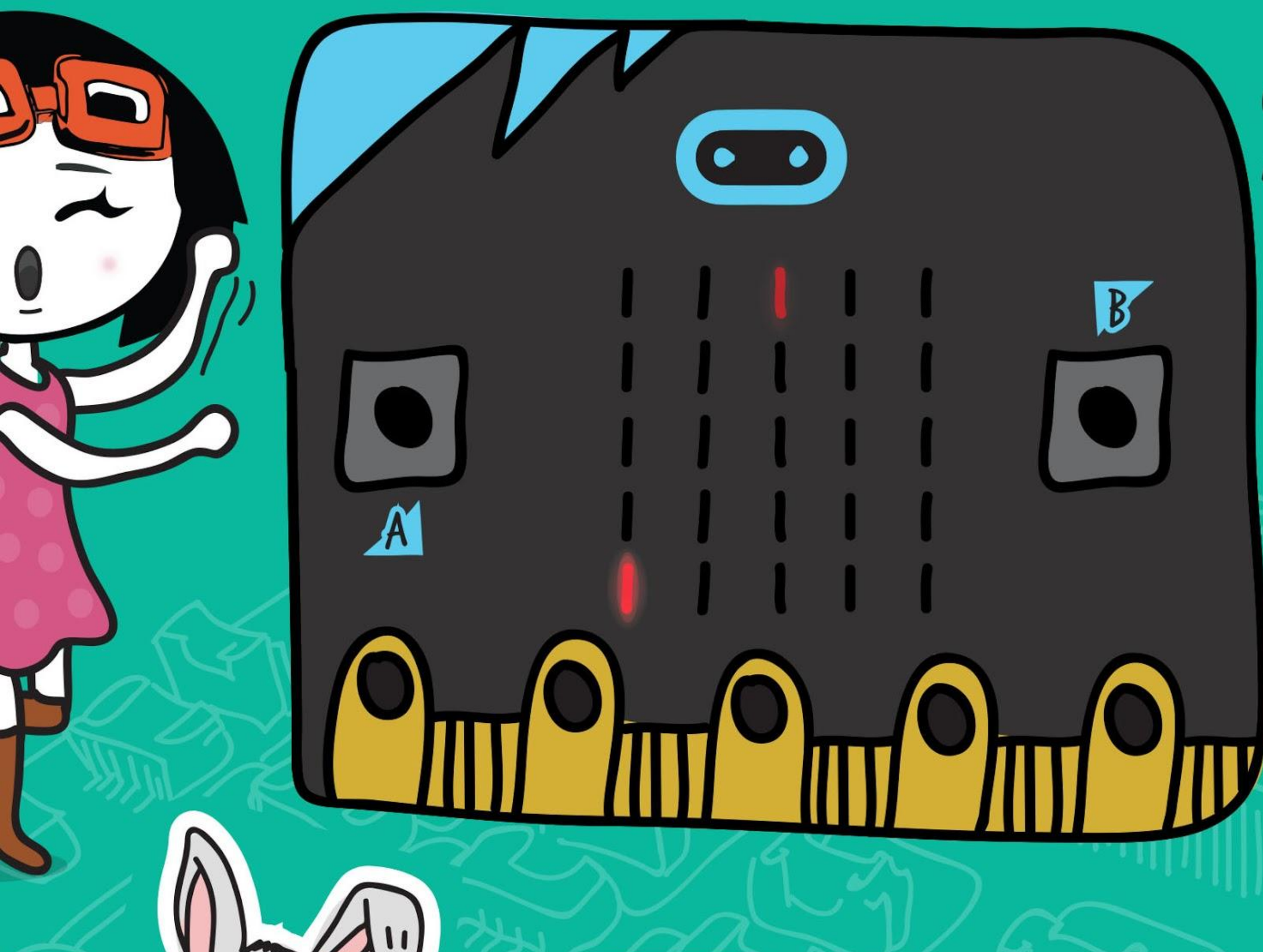


Du musst IR Bit neu kalibrieren, wenn du ein T-Shirt mit einer anderen Farbe anziehst. Kannst du erraten warum?

KAPITEL 6

Fangen spielen - “Ich hab dich!”

Potentio Bit

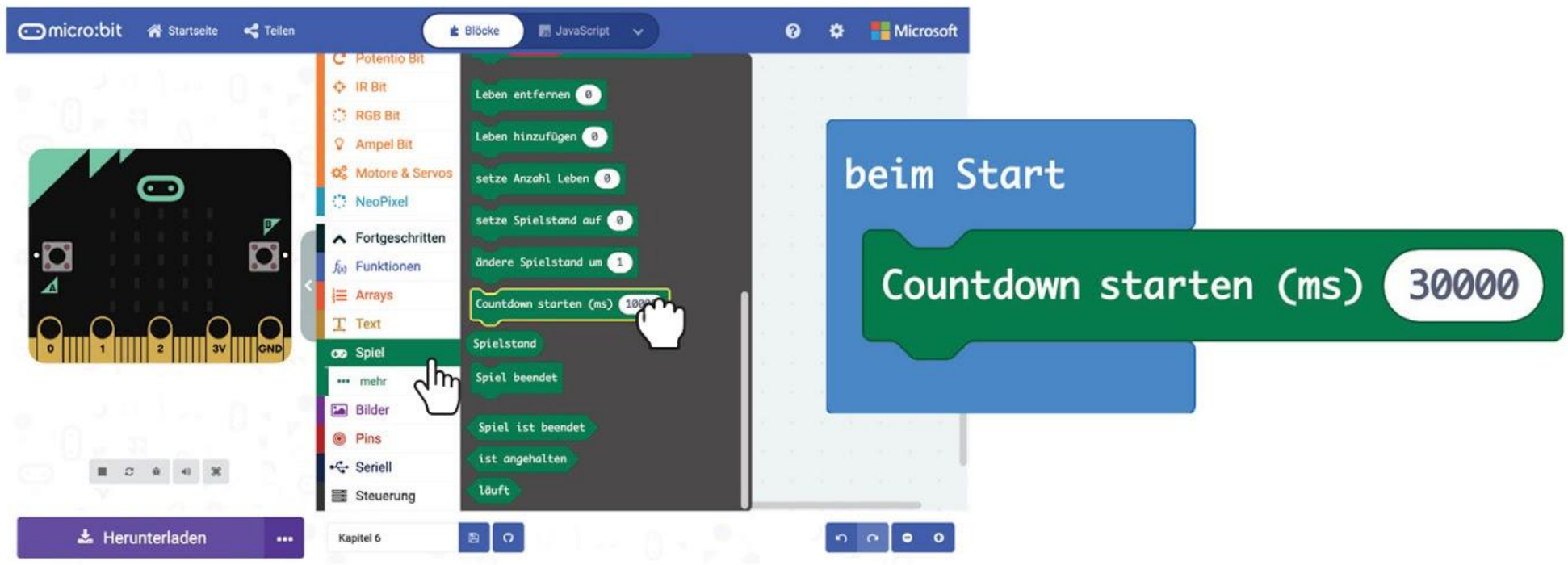


Scanne mich!

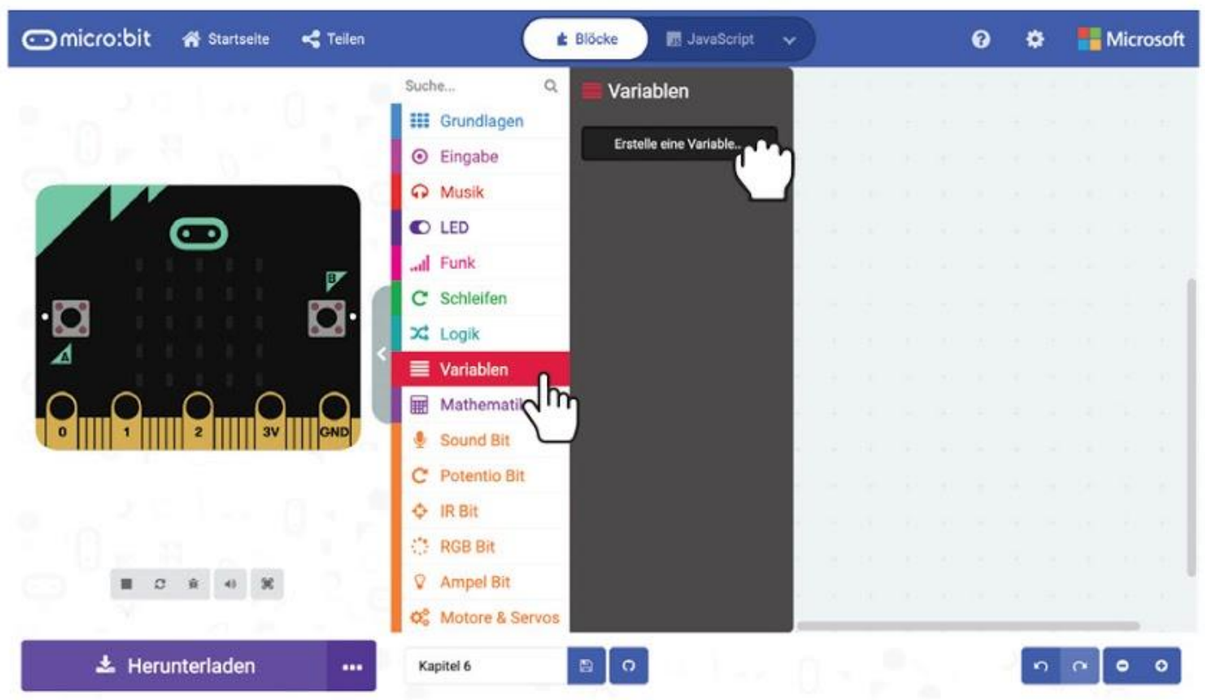
link.cytron.io/edubit-chapter-6

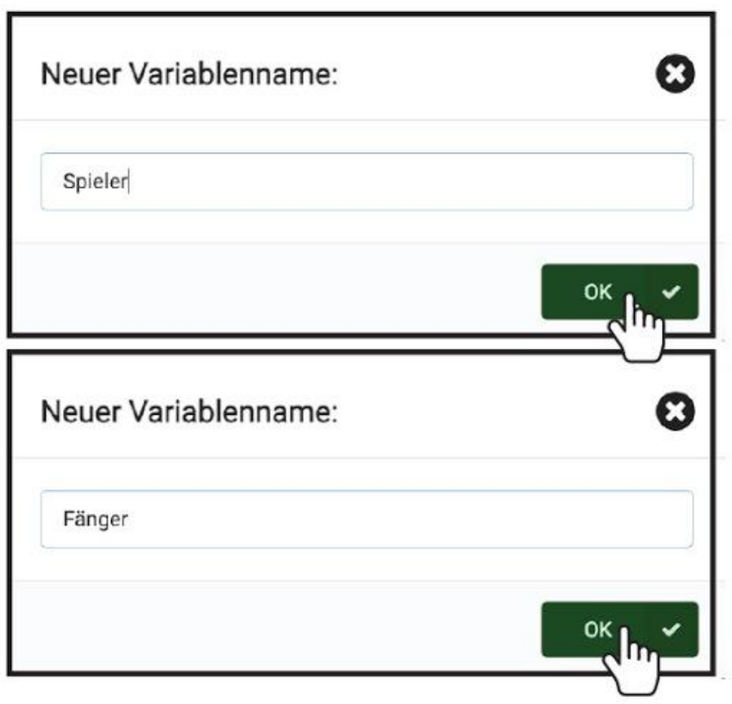
LASS UNS PROGRAMMIEREN!

Schritt 1 Erstelle im MakeCode Editor ein neues Projekt und füge die Erweiterung für EDU:BIT hinzu (siehe Seite 40). Klicke die Gruppe **[Fortgeschritten]** und **[Spiel]** an. Ziehe den Block **[Countdown starten (ms) _]** in den Block **[beim Start]** und ändere den Wert zu **30000**.

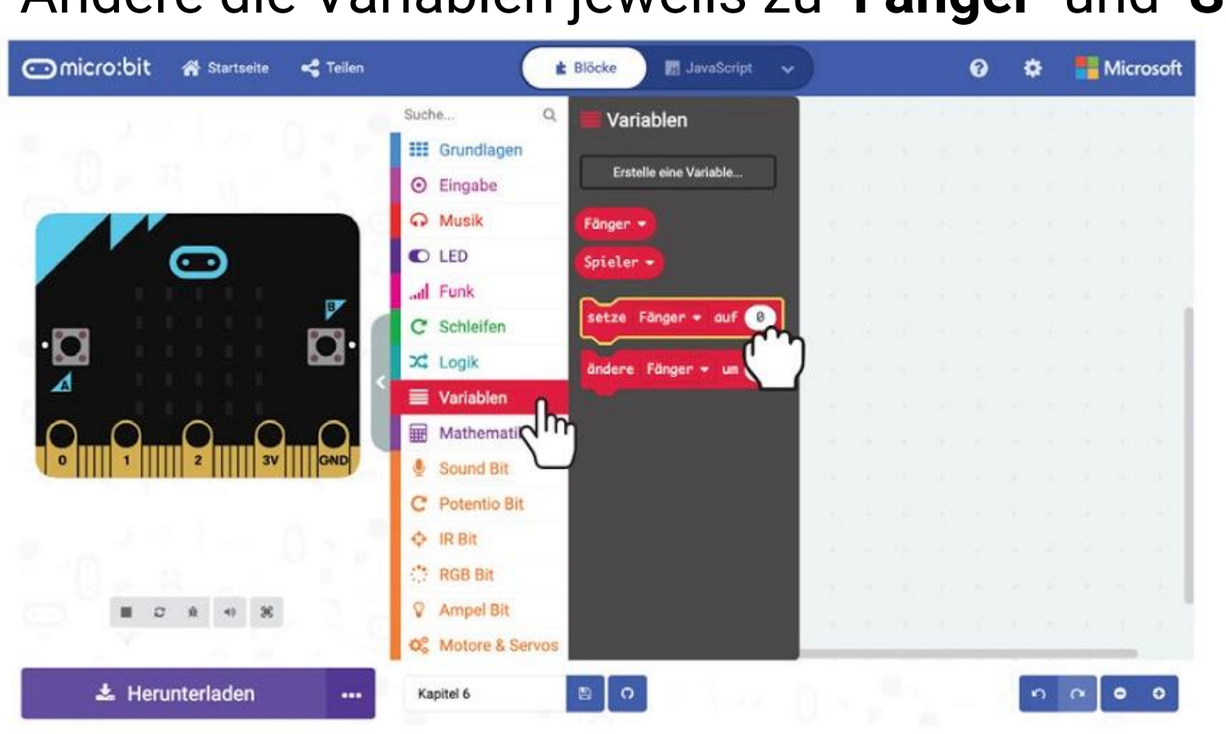


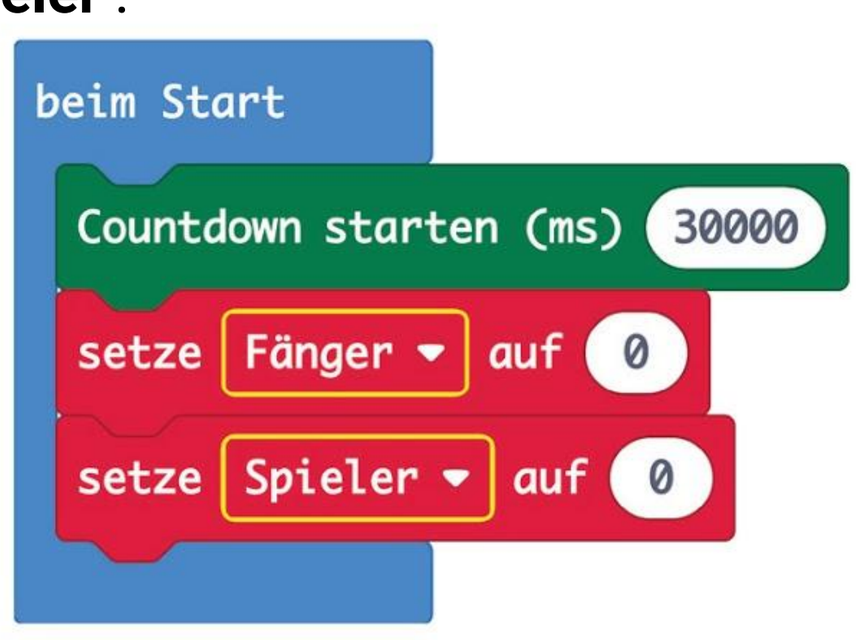
Schritt 2 Klicke auf **[Variablen]** und auf **[Erstelle eine Variable]**. Nenne sie ‘ **Spieler’** und klicke auf OK. Erstelle eine zweiter Variable namens **‘Fänger’**.



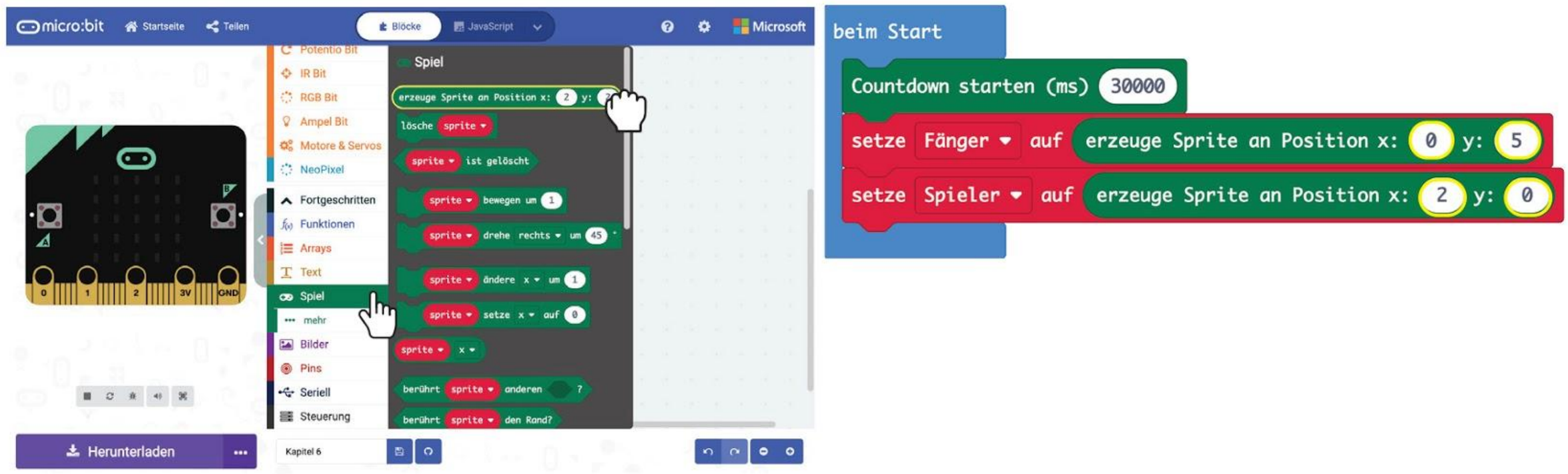


Schritt 3 Klicke auf **[Variablen]** und klicke auf den Block **[setze _ auf _]**. Dupliziere diesen Block und lege beide Blöcke in den **[beim Start]**-Block.
Ändere die Variablen jeweils zu **‘Fänger’** und **‘Spieler’**.

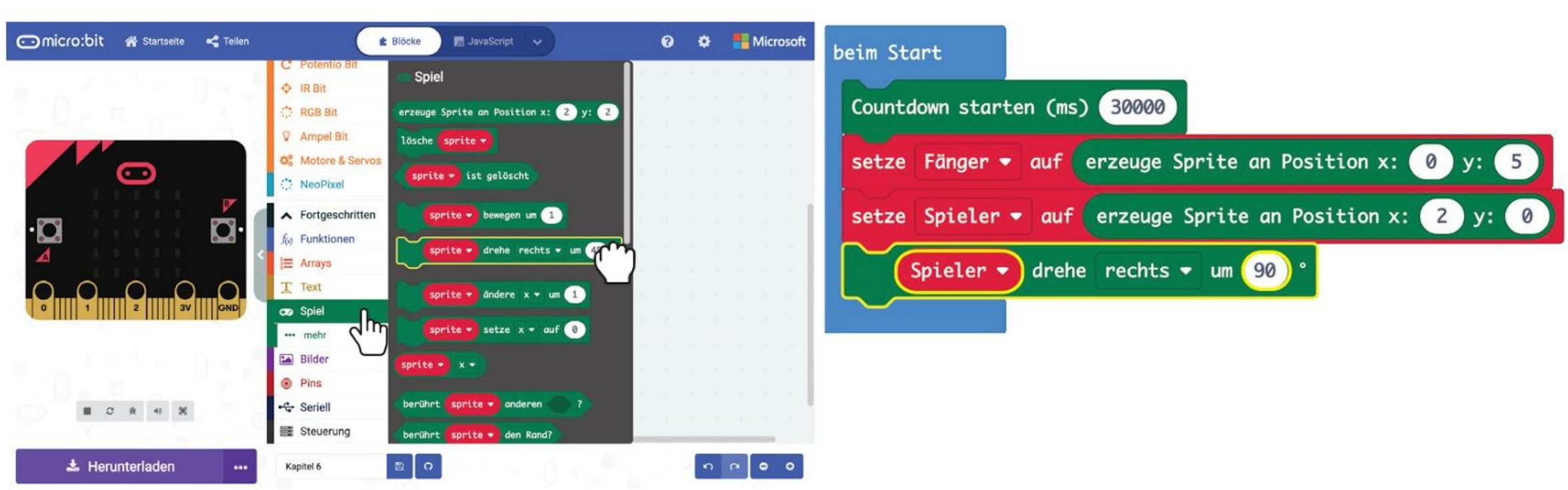




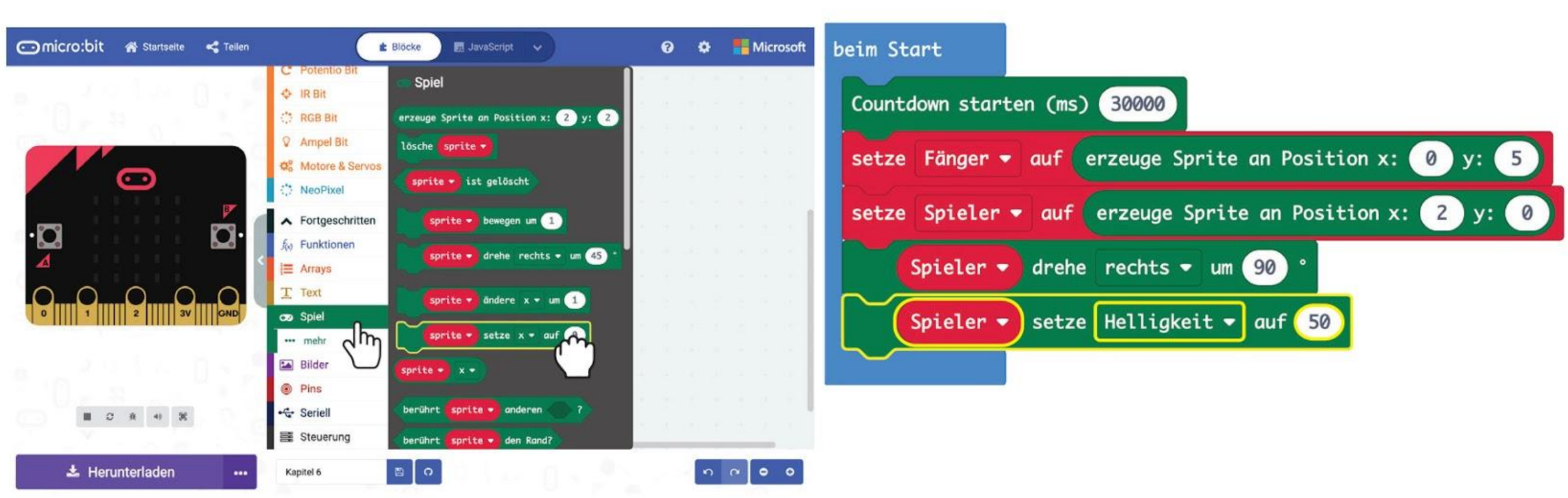
Schritt 4 Klicke in **[Fortgeschritten]** auf **[Spiel]**. Klicke auf den Block **[erzeuge Sprite an Position x:_ y:_]**. Dupliziere die Blöcke und lege sie in die **[setze _ auf _]**-Blöcke. Ändere die Werte zu **x:0 y:5** für den **‘Fänger’** und **x:2 y:0** für den **‘Spieler’**.



Schritt 5 Klicke auf **[Fortgeschritten]** : **[Spiel]** und wähle den Block **[_ drehe _ um _ °]**. Lege ihn in den **[beim Start]**-Block. Wähle die Variable **‘Spieler’** und setze den Wert auf **90**.

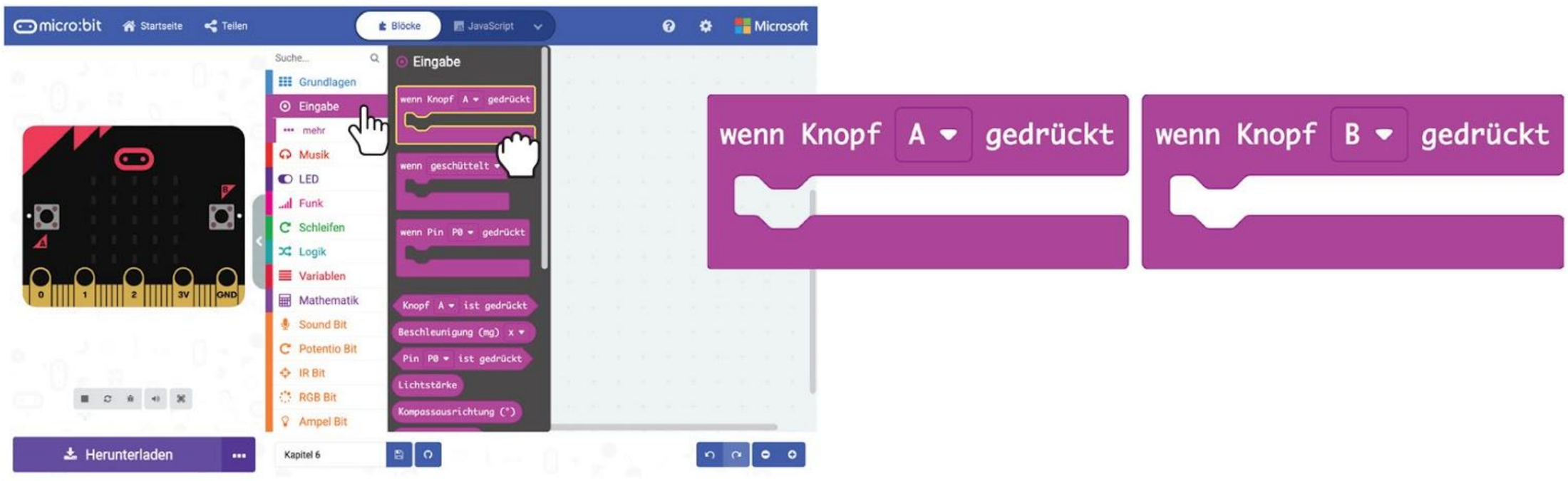


Schritt 6 Klicke in der Gruppe **[Fortgeschritten]** : **[Spiel]** auf den Block **[_ setze _ auf _]**. Wähle die Variable **‘Spieler’** , ändere **‘x’** zu **‘Helligkeit’** und gib **50** ein.

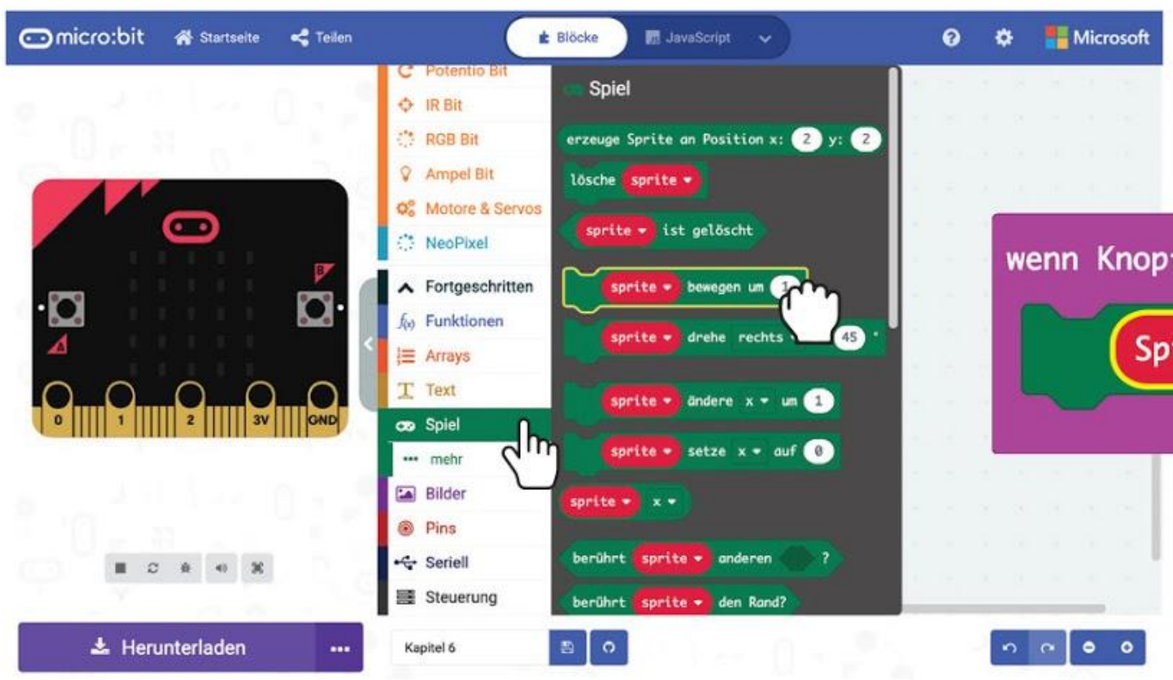


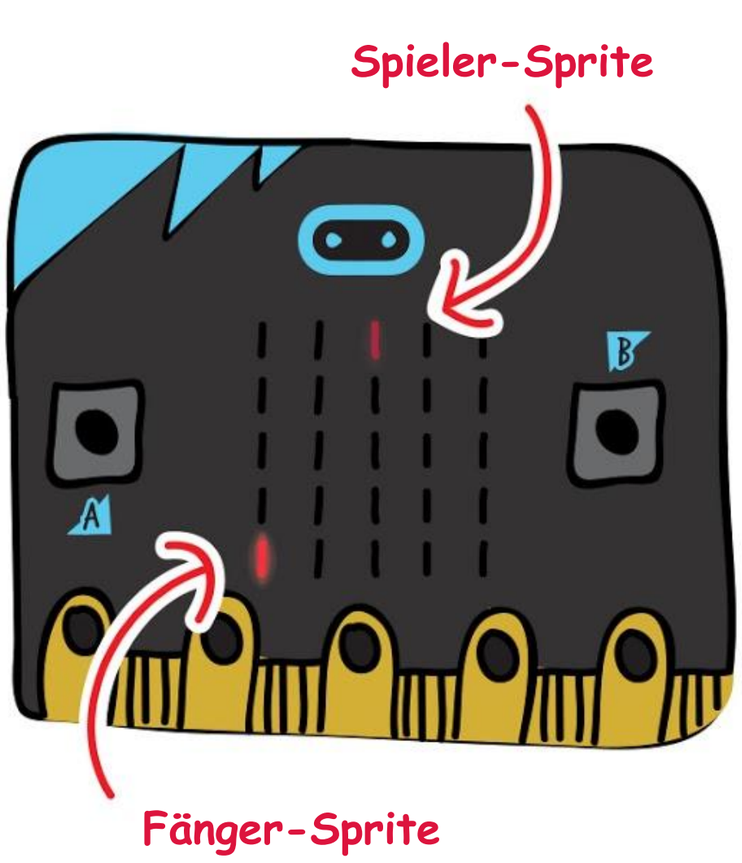
KAPITEL 6 : Fangen spielen

Schritt 7 Klicke auf **[Eingabe]** und wähle den Block **[wenn Knopf _ gedrückt]** aus. Dupliziere den Block und wähle **‘B’** für den zweiten **[wenn Knopf _ gedrückt]**-Block.



Schritt 8 Klicke in der Gruppe **[Spiel]** auf den Block **[_ bewegen um _]**. Dupliziere ihn und lege die Blöcke jeweils in die **[wenn Knopf _ gedrückt]**-Blöcke. Wähle die Variable **‘Spieler’** für beide Blöcke und ändere die Werte zu **-1** (Knopf A gedrückt) und **1** (Knopf B gedrückt).

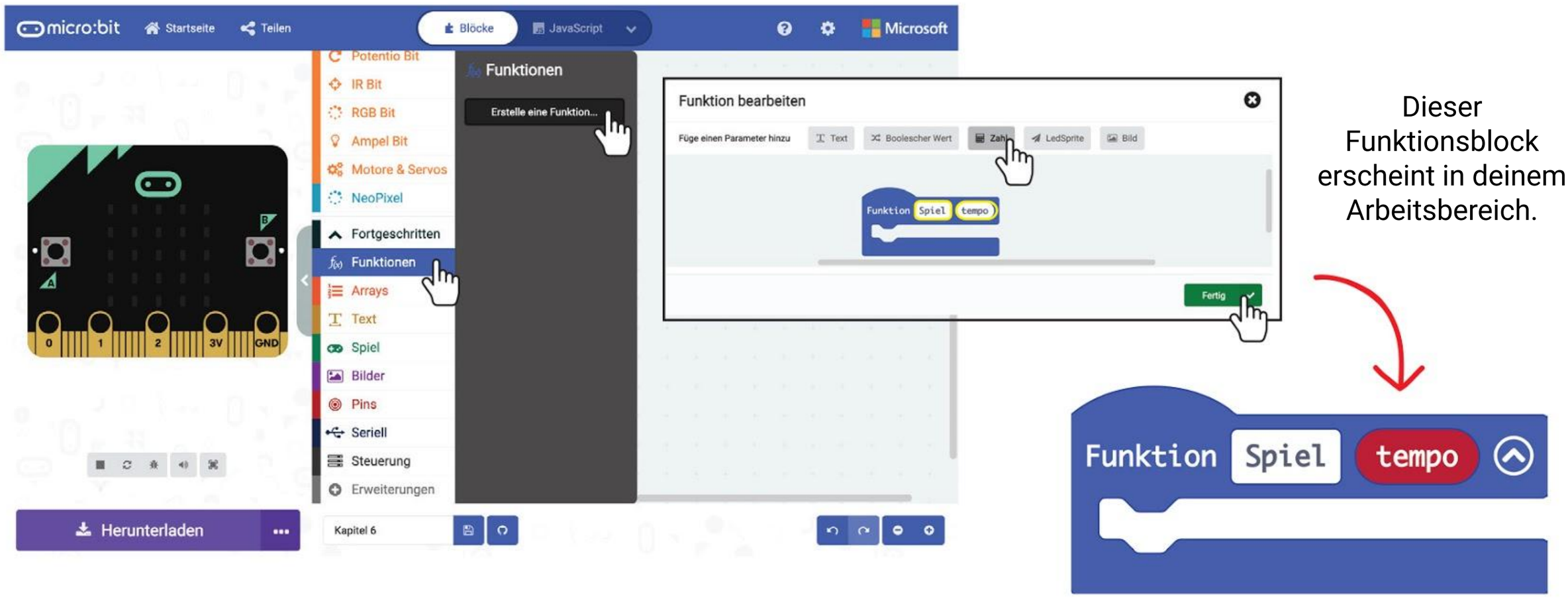




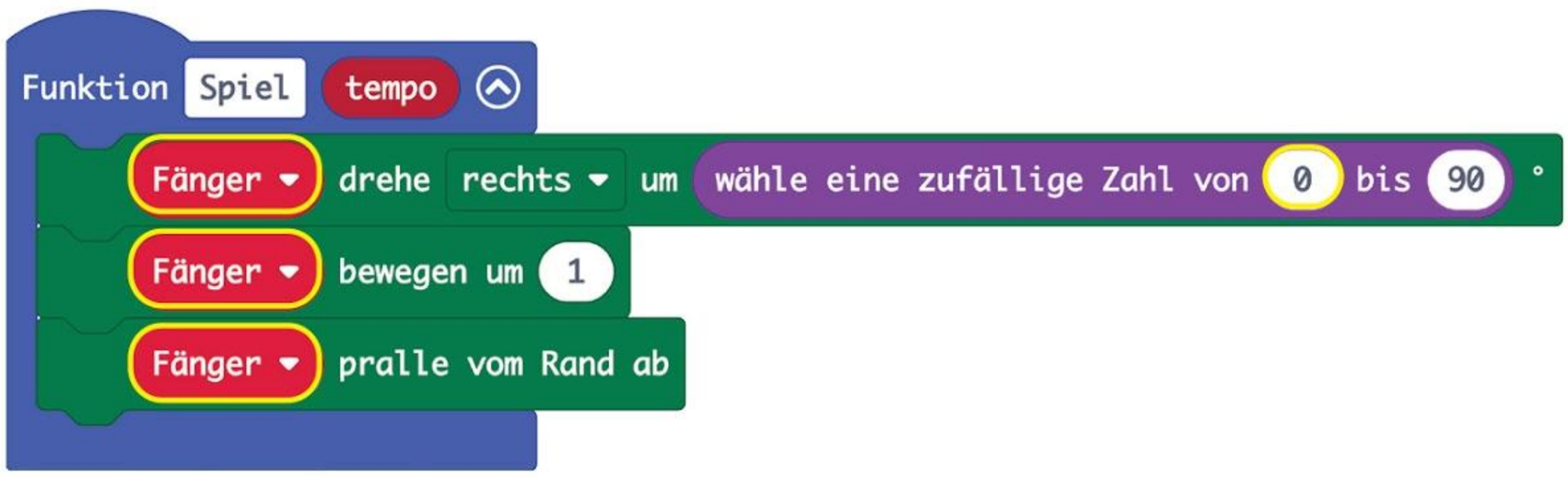
Flashe den Code auf deinen EDU:BIT. Wenn du den blauen Knopf (B) drückst, bewegt sich die dunklere LED nach unten. Das ist das Spieler-Sprite.

Ein Sprite ist wie ein kleines “LED-Gespenst”, das du steuern kannst.
Was passiert, wenn du den gelben Knopf (A) drückst?

Schritt 9 Klicke in der Kategorie **[Fortgeschritten]** auf **[Funktionen]** und auf **[Erstelle eine Funktion…]**. Im Fenster “Funktion bearbeiten” ändere ‘macheEtwas’ zu **‘Spiel’**. Klicke auf **[Zahl]** um einen Parameter hinzuzufügen und ändere ‘num’ zu **‘tempo’** Klicke dann auf **‘Fertig’**.

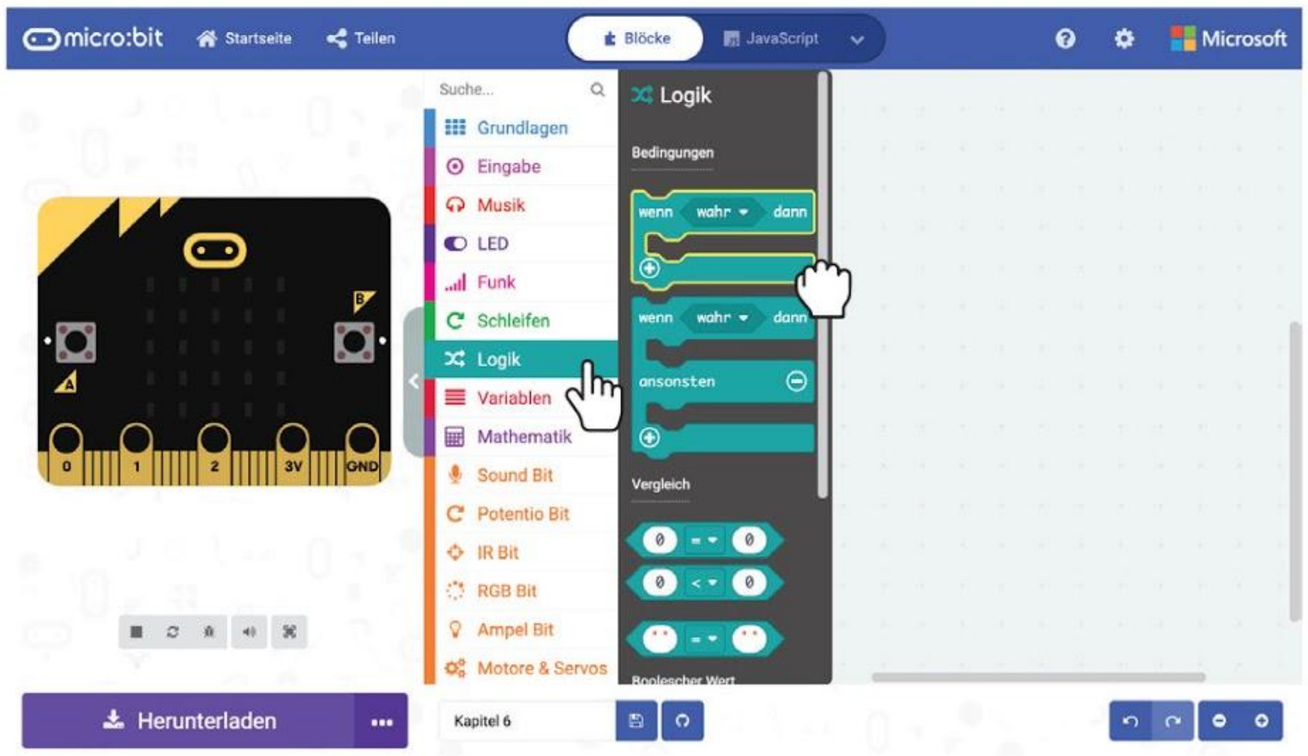


Schritt 10 Baue deinen Code zusammen, indem du Blöcke aus **[Fortgeschritten] : [Spiel]** und **[Mathematik]** wie unten gezeigt verwendest. Denke daran, die Variable in **‘Fänger’** und den Wert auf **90** zu ändern.



KAPITEL 6 : Fangen spielen

Schritt 11 Füge zwei **[wenn _ dann]**-Blöcke aus der Gruppe **[Logik]** hinzu.



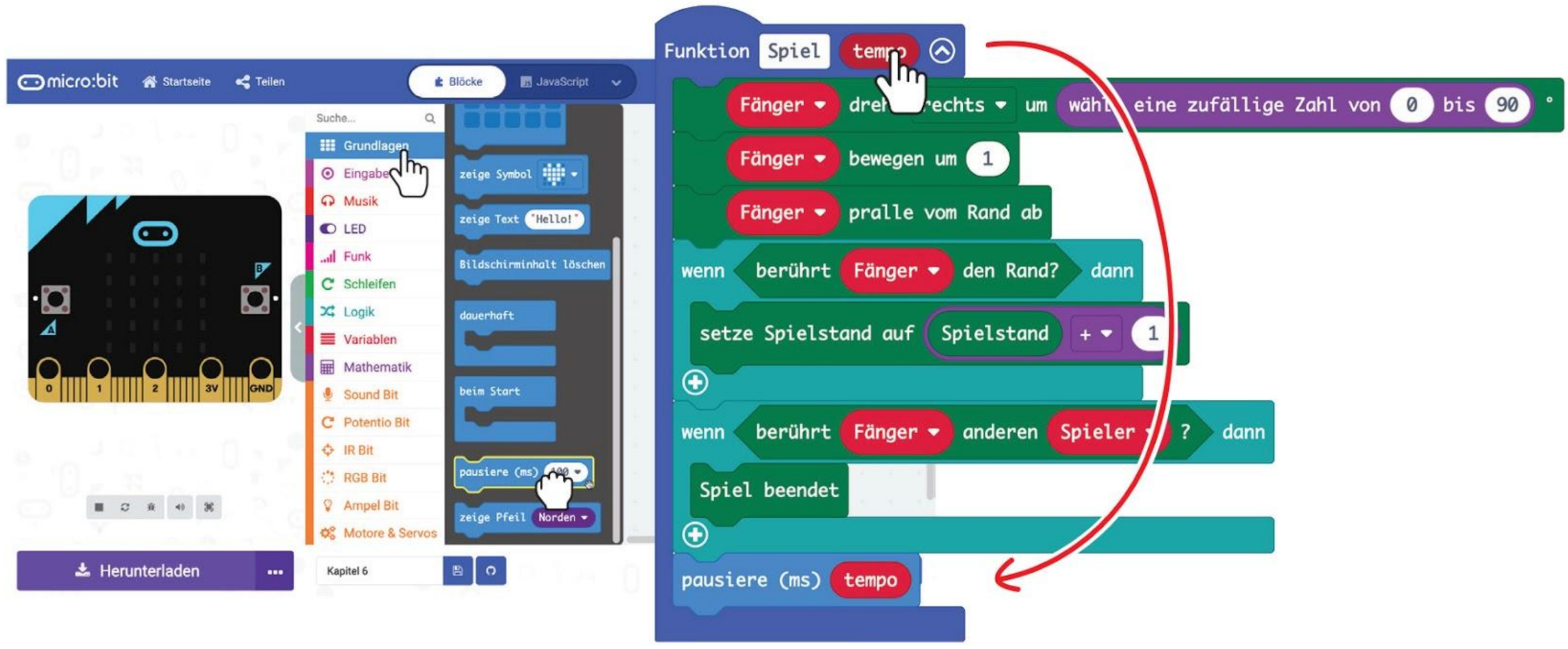
Schritt 12 Mache weiter mit Blöcken aus den Gruppen **[Fortgeschritten] : [Spiel]** und **[Mathematik]** so wie hier gezeigt.

Weiter so!

KAPITEL 6 : Fangen spielen

Schritt 13 Ändere beide **[sprite]**-Blöcke zu **‘Fänger’** indem du sie klickst und **‘Fänger’** auswählst. Klicke auf **[Variablen]** und wähle den Block **[Spieler]**. Lege ihn in den leeren Bereich im Block **[berührt _ anderen _]**.

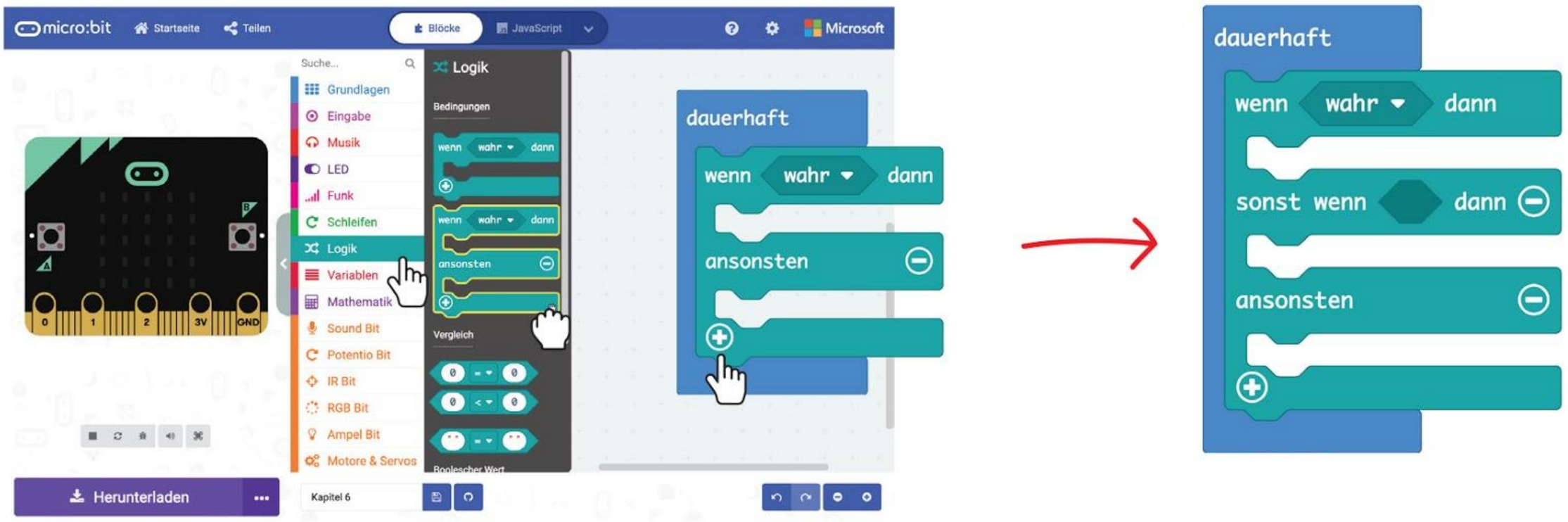
Schritt 14 Klicke in **[Grundlagen]** auf den Block **[pausiere (ms) _]**. Füge ihn zu deinem Code hinzu. Klicke auf den Block **[tempo]** oben in deinem Funktions-Block und ziehe ihn in den Wert-Bereich im Block **[pausiere (ms) _]**.



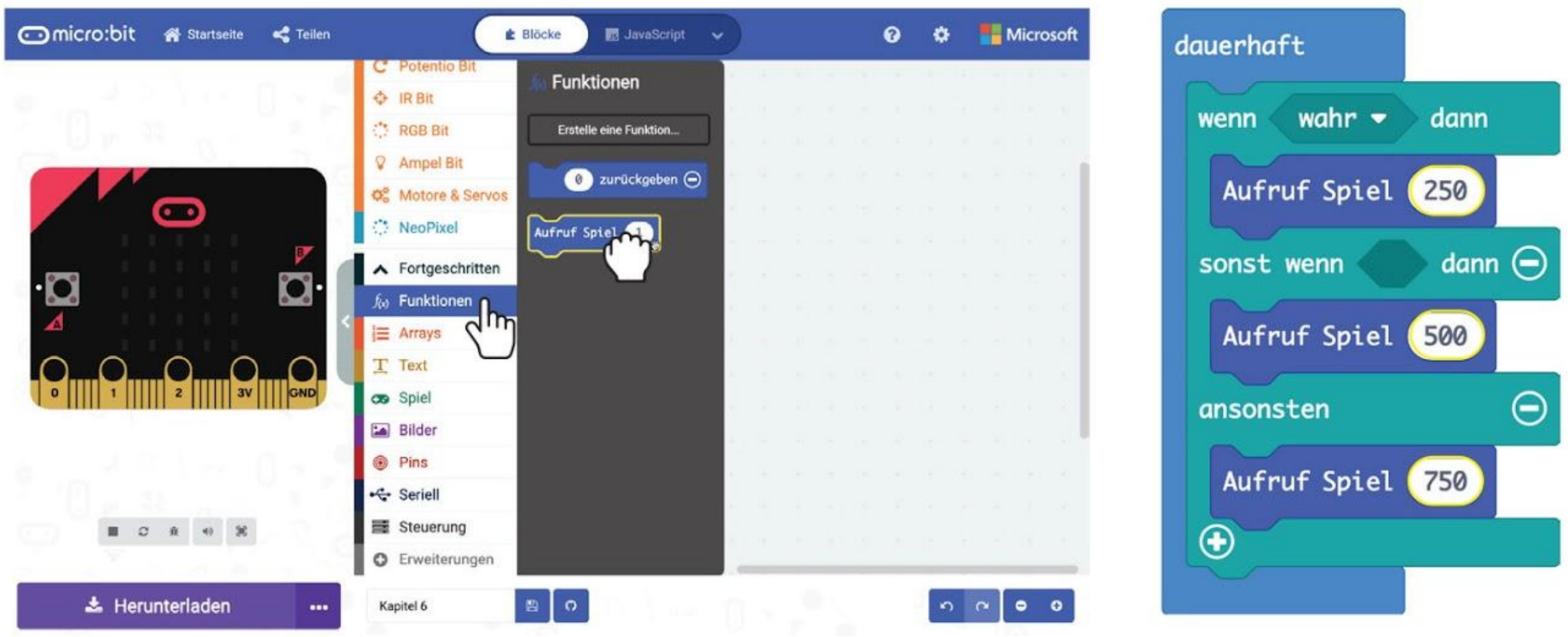
Du kannst dein Spiel aufregender machen, indem du eine Melodie spielst, wenn der Fänger den Spieler berührt. Findest du heraus, welchen Block du wo hinzufügen musst?

Jetzt fügen wir unterschiedlich schwierige Level zum Spiel hinzu!

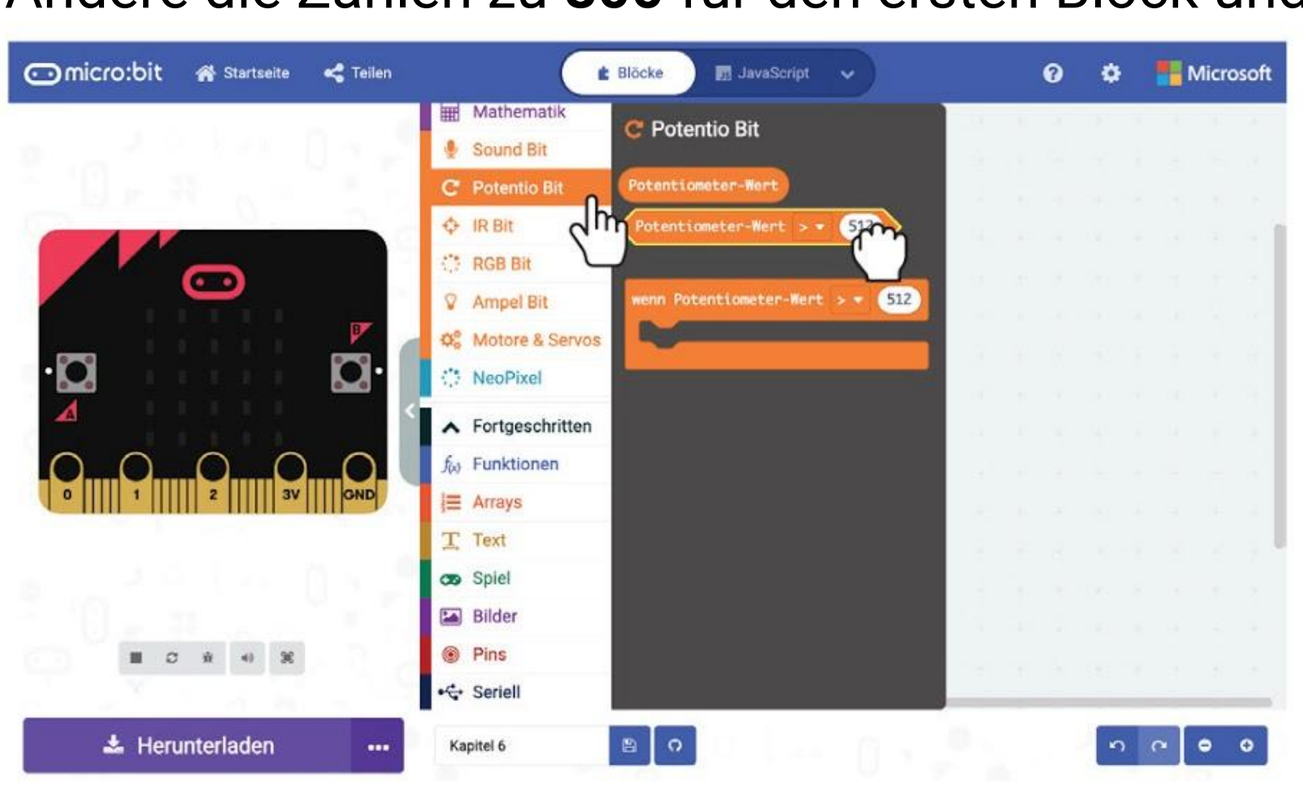
Schritt 15 Klicke unter **[Logik]** auf **[wenn-dann-ansonsten]**. Lege den Block in den Bereich **[dauerhaft]**. Klicke auf das (+) um eine weitere Bedingung hinzuzufügen.



Schritt 16 Klicke in der Gruppe **[Funktionen]** auf **[Aufruf Spiel _]**. Dupliziere den Block und lege jeweils einen Block in jeden **[wenn-dann-ansonsten]**-Bereich. Ändere die Zahl in den **[Aufruf Spiel _]**-Blöcken jeweils auf **250**, **500** und **750.**



Schritt 17 Klicke auf **[Potentio Bit]** und **[Potentiometer-Wert > _]**. Dupliziere und lege die Blöcke in die Bedingungsfelder in den **[wenn _ dann _ ansonsten]**-Blöcken. Ändere die Zahlen zu **800** für den ersten Block und **400** für den zweiten.



Das ist der fertige Code:

Zeige eine Animation und starte einen Countdown.

Erstelle Sprites für Fänger und Spieler an den gegebenen Koordinaten.

Drehe das Spieler-Sprite um 90 Grad (damit es sich rauf und runter bewegt -> vertikal)

Mache das Spieler-Sprite dunkler.

```
beim Start
    Countdown starten (ms) 30000
    setze Fänger auf erzeuge Sprite an Position x: 0 y: 5
    setze Spieler auf erzeuge Sprite an Position x: 2 y: 0
    Spieler drehe rechts um 90 °
    Spieler setze Helligkeit auf 50
```

Verwende die Knöpfe A und B um das Spieler-Sprite aufwärts und abwärts zu bewegen.

```
wenn Knopf A gedrückt
    Spieler bewegen um -1
wenn Knopf B gedrückt
    Spieler bewegen um 1
```

Die Funktion “Spiel” steuert das Fänger-Sprite.

Das Fänger-Sprite dreht sich um einen zufälligen Winkel zwischen 0 und 90 Grad und macht dann einen Schritt. Am Rand prallt es ab.

Immer wenn das fänger-Sprite den Rand berührt, wird zur Variable Spielstand 1 Punkt hinzugefügt.

Wenn das Fänger-Sprite das Spieler-Sprite berührt, spielt die Melodie wawawawaa und der Bildschirm zeigt “Game Over” und den Spielstand.

Pausiere das Programm für die Anzahl an Millisekunden, die in der Variable “tempo” steht.

```
Funktion Spiel tempo
    Fänger drehe rechts um wähle eine zufällige Zahl von 0 bis 90 °
    Fänger bewegen um 1
    Fänger pralle vom Rand ab
    wenn berührt Fänger den Rand? dann
        setze Spielstand auf Spielstand + 1
    wenn berührt Fänger anderen Spieler ? dann
        Beginne Melodie wawawawaa Wiederhole einmal
        Spiel beendet
    pausiere (ms) tempo
```

Die Potentiometer-Einstellung wird laufend überprüft.

Wenn der Potentiometer-Wert größer als 800 ist, wird die Funktion “Spiel” mit dem Wert “250” aufgerufen.

Wenn der Potentiometer-Wert größer als 400 ist, wird “Spiel” mit dem Wert “500” aufgerufen.

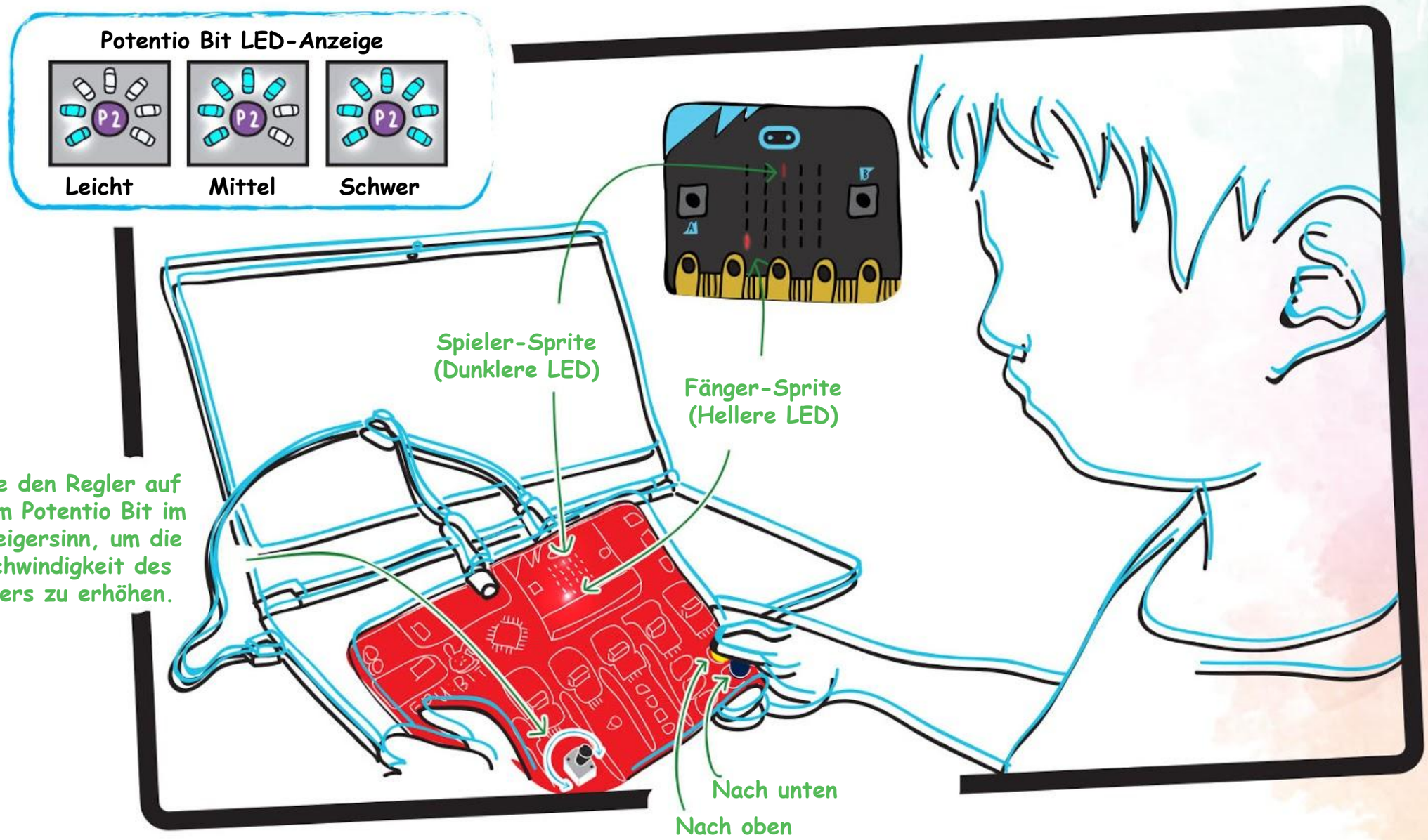
Ansonsten wird die Funktion “Spiel” mit dem Wert “750” aufgerufen.

```
dauerhaft
    wenn Potentiometer-Wert > 800 dann
        Aufruf Spiel 250
    sonst wenn Potentiometer-Wert > 400 dann
        Aufruf Spiel 500
    ansonsten
        Aufruf Spiel 750
```

Schritt 18 Lade den fertigen Code auf deinen EDU:BIT. Viel Spaß beim Fangenspielen!

Spielen wir!

"Ich hab dich!"



SPIELANLEITUNG:

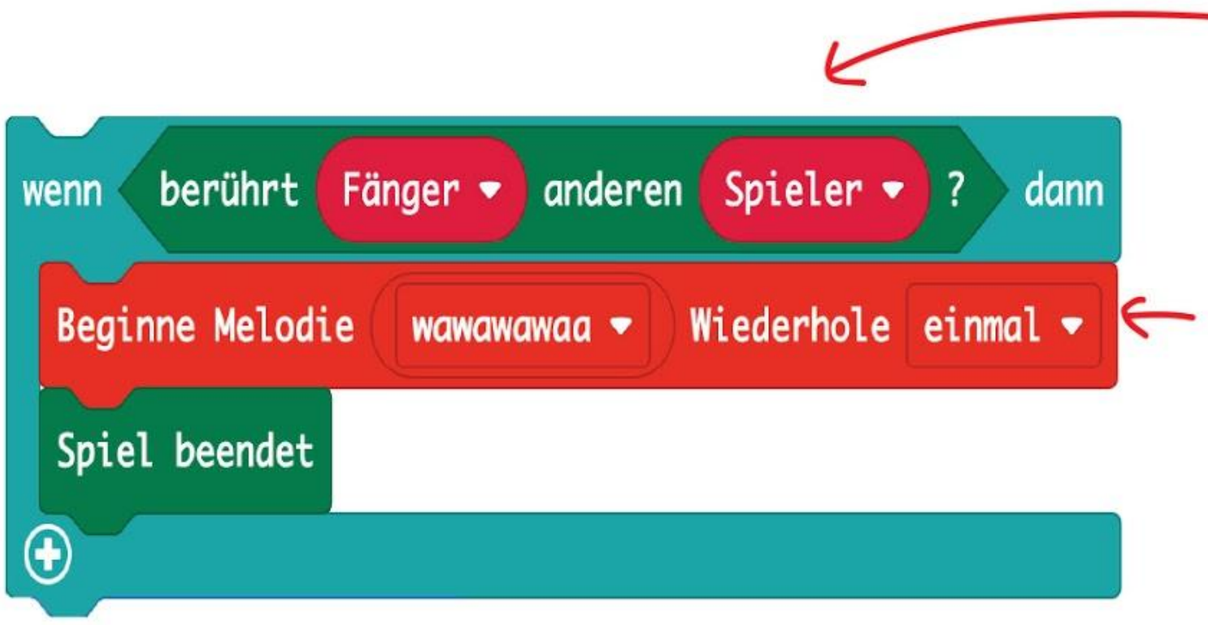
- Nach dem Einschalten bewegt sich der Fänger in eine zufällige Richtung.
- Bewege das Spieler-Sprite nach oben oder nach unten um dem Fänger auszuweichen. Drücke den gelben Knopf (A) um nach oben zu springen und den blauen Knopf (B) um nach unten zu springen.
- Das Spiel ist zu Ende, wenn das Spieler-Sprite vom Fänger-Sprite berührt wird oder wenn 30 Sekunden vergangen sind.
- Jedesmal wenn das Fänger-Sprite den Rand des Bildschirms "berührt", bekommst du 1 Punkt. Wer den höchsten Punktestand erreicht, gewinnt! Viel Spaß~

TIPPS!

#1 Um innerhalb der 30 Sekunden mehr Punkte zu erhalten, kannst du die Geschwindigkeit des Fängers erhöhen, damit er den Rand öfter "berührt".
#2 Nach dem Spiel kannst du A+B gleichzeitig drücken um eine neue Runde zu starten. Diese Funktion ist automatisch in die [Spiel]-Blöcke eingebaut.

KNACKE DEN CODE

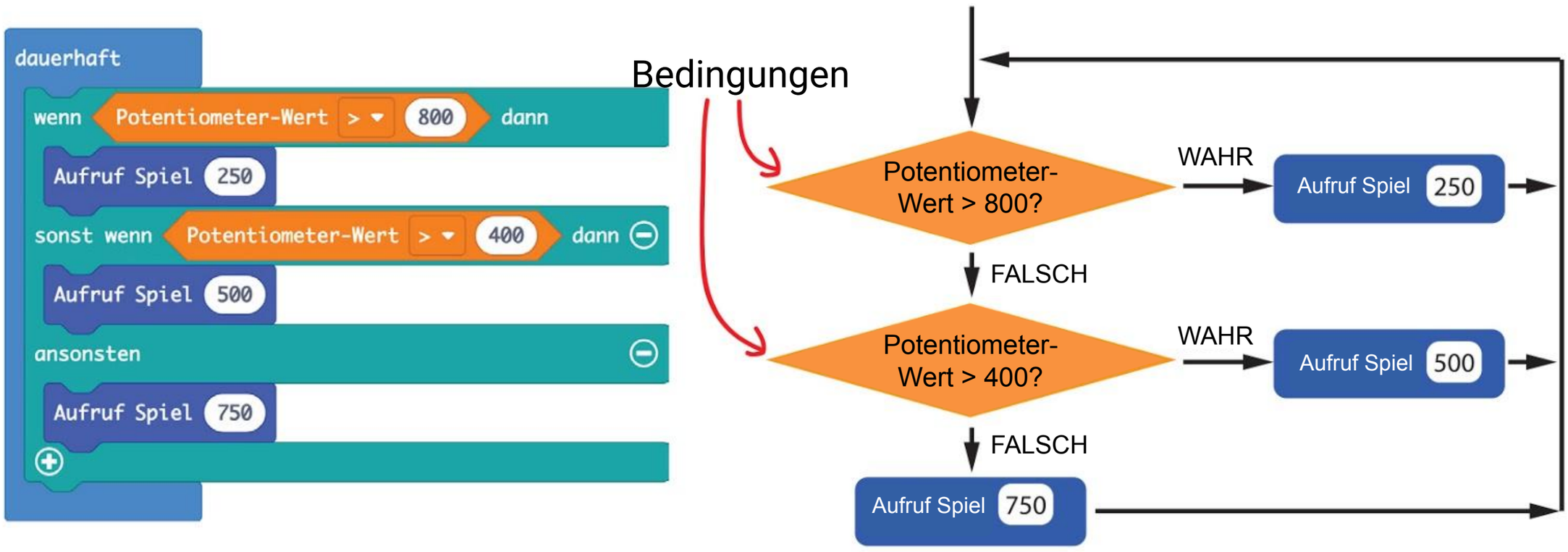
Beim Programmieren benutzen wir **bedingte Anweisungen** um Entscheidungen zu treffen. In MakeCode benutzen wir die Blöcke **[wenn-dann]** oder **[wenn-dann-ansonsten]** aus der Gruppe **[Logik]** dafür. Das Programm überprüft die Bedingung, und wenn sie WAHR ist, führt es den Code in dem Block aus. Wenn sie FALSCH ist, wird der Code im nächsten Block ausgeführt.



wenn diese Bedingung wahr ist
(z.B. das Fänger-Sprite berührt das Spieler-Sprite),
dann mache das
(Beginne die Melodie wawawawaa und zeige die “Game Over”-Animation.

Wenn wir mehrere Bedingungen haben, überprüft das Programm sie der Reihe nach von oben nach unten, und führt den Code der ersten Bedingung aus, die WAHR ist. Also hat eine “obere” Bedingung Vorrang vor einer “unteren”.

Ein Beispiel: Dieser Code aus dem Spiel legt die Geschwindigkeit des Fängers fest, indem er den Potentiometer-Wert mit vorgegebenen Schwellenwerten vergleicht.



wenn Potentiometer-Wert > 800, rufe die Funktion Spiel auf (mit der Variable tempo = 250 ms),
sonst wenn Potentiometer-Wert > 400, rufe die Funktion Spiel auf (mit tempo = 500 ms),
ansonsten rufe die Funktion Spiel auf (mit tempo = 750 ms)

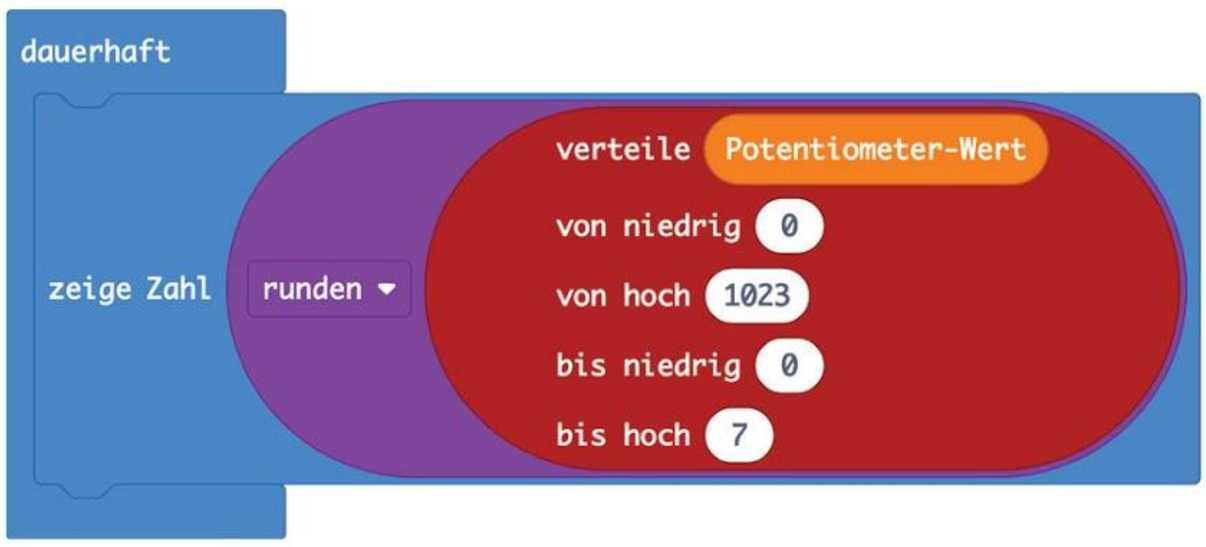
ENTDECKE NOCH MEHR

#1 Benutze einen **[Grundlagen] : [zeige Zahl]**-Block zusammen mit **[Potentio Bit] : [Potentiometer-Wert]** um die aktuelle Position des Potentiometers (auch Poti genannt) auszulesen und anzuzeigen.

#2 Das Poti kann einen Wert zwischen 0 und 1023 haben. Benutze **[verteile _ von niedrig _ von hoch _ bis niedrig _ bis hoch _]** aus **[Fortgeschritten] : [Pins]** um den Wert in einen besser nutzbaren Bereich umzurechnen.

#3 Der Verteile-Block liefert eine Dezimalzahl (z.B. 1.68, 3.998) als Ergebnis. Um die Zahl zu runden, benutze **[runden _]** aus der Gruppe **[Mathematik]**.

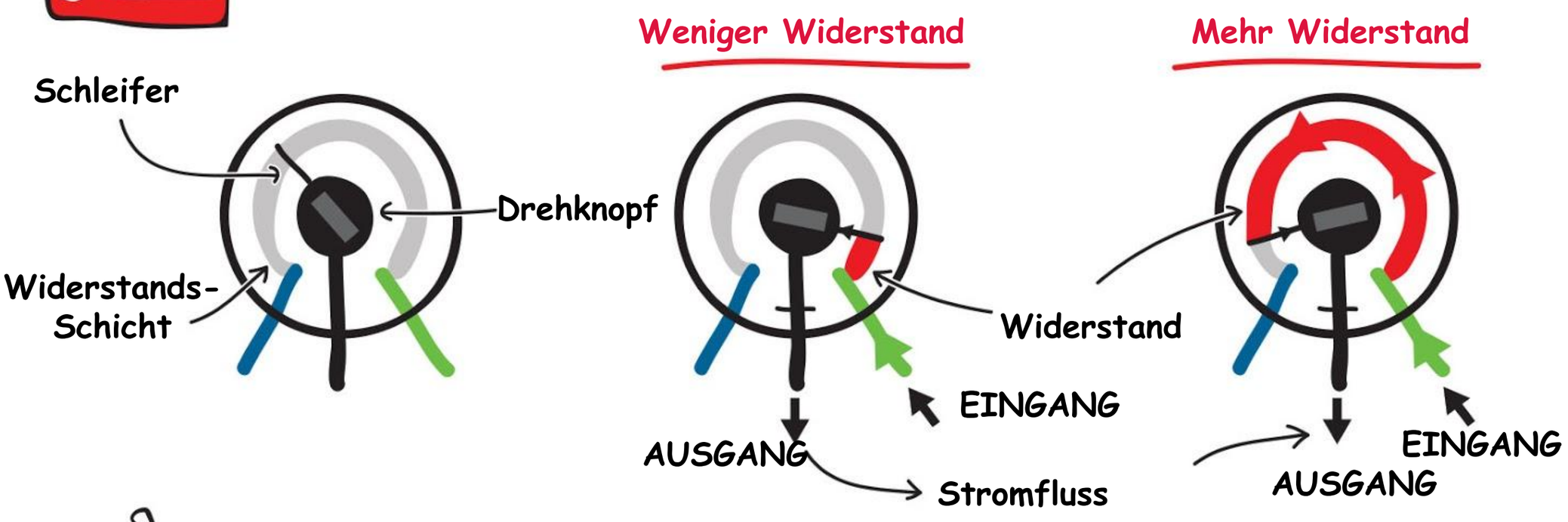
Hier ist ein Beispiel, in dem der Potentiometer-Wert in einen Bereich von 0 bis 7 neu verteilt wird. Der Wert wird gerundet und auf der LED-Matrix angezeigt.



GUT ZU WISSEN!

Potentiometer, auch Potis genannt, sind veränderbare elektrische Widerstände mit einem Dreh- oder Schieberegler.

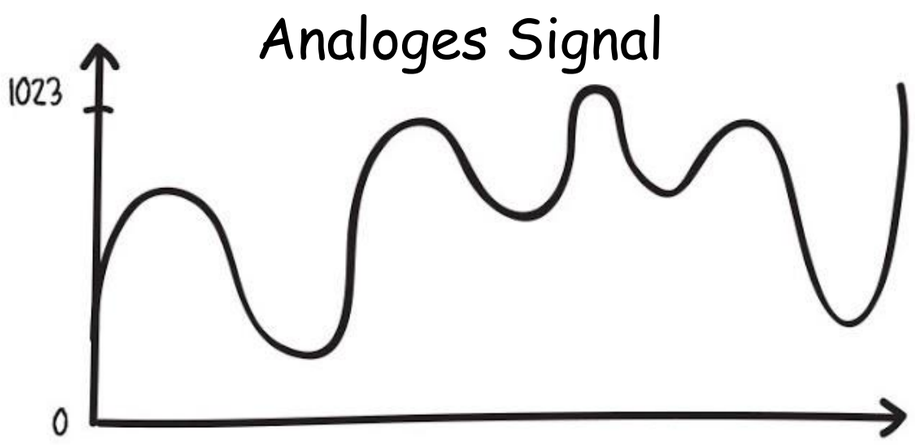
Bei einem Poti mit 10.000Ω kannst du einen Widerstand von 0Ω bis 10.000Ω einstellen, indem du die Position des Schleifers veränderst.

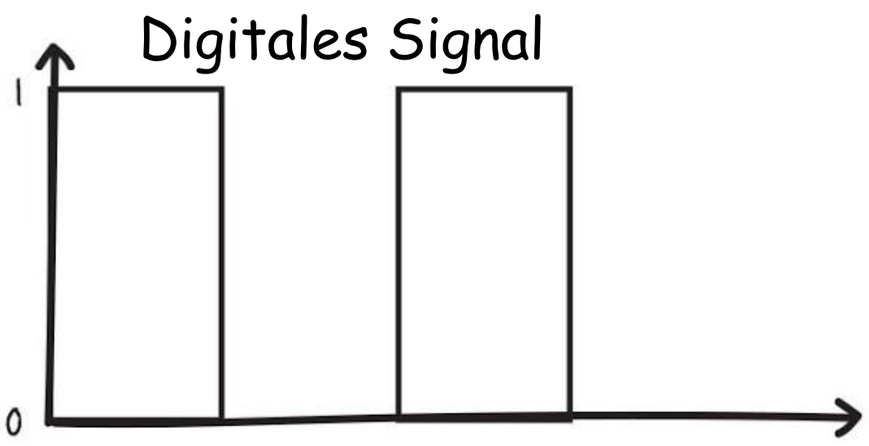


VERWENDUNGSMÖGLICHKEITEN:

- Als Lautstärkeregler für Lautsprecher
- Als Frequenzregler beim Radio
- Als Temperaturregler beim E-Herd

Der Poti des EDU:BIT ist ein analoges Eingabegerät. Es misst das elektrische Potential und wandelt die gemessene Spannung (zwischen 0 V und 3.3 V in einen ganzzahligen Wert zwischen 0 und 1023 um.





HERAUSFORDERUNG

Mache aus EDU:BIT einen Timer. Stelle die Dauer (zwischen 0 und 60 Sekunden) mit Potentio Bit ein. Knopf A aktiviert den Timer und Knopf B setzt ihn zurück.	
beim Start	Setze Modus auf 0
wenn Knopf A (gelb) gedrückt	Setze Modus auf 1 Setze Startzeit auf Laufzeit Zeige einen Smiley an
wenn Knopf B (blau) gedrückt	Setze Modus auf 0
dauerhaft	Überprüfe immer den Modus • WENN Modus=0, setze Dauer auf einen gerundeten Wert des Potentiometers, der auf Werte zwischen 0 und 60 neu verteilt wurde. Zeige die Dauer an. • SONST WENN Modus=1, überprüfe ob (Laufzeit - Startzeit) > (Dauer x 1000). Wenn das WAHR ist, spiele die Melodie “wawawawaa” und setze dann Modus auf 2. • ANSONSTEN zeige trauriges Gesicht

Hier sind noch ein paar Tipps:
#1 Du brauchst drei Variablen: Modus, Startzeit und Dauer.
#2 Du findest den Block [Laufzeit (ms)] in der Gruppe [Eingabe].
#3 Benutze diese Bedingung um zu prüfen, ob die Zeit um ist.

KAPITEL 7

Applaus, Applaus!

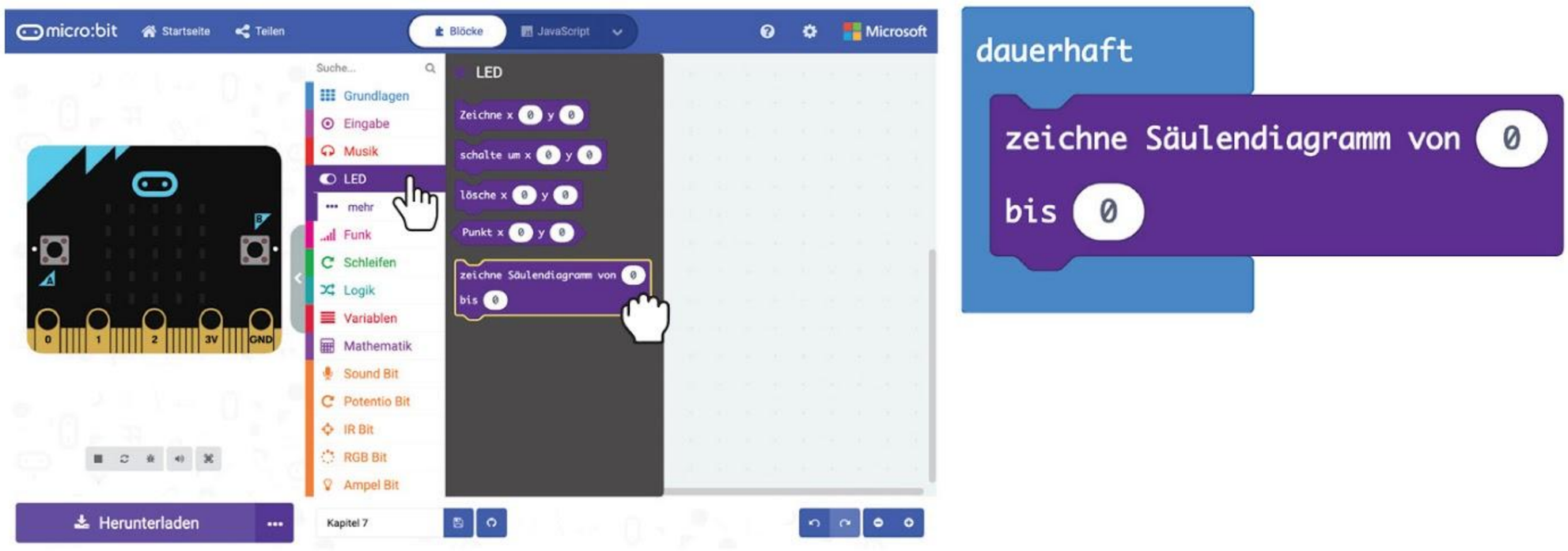
Sound Bit

Scanne mich !

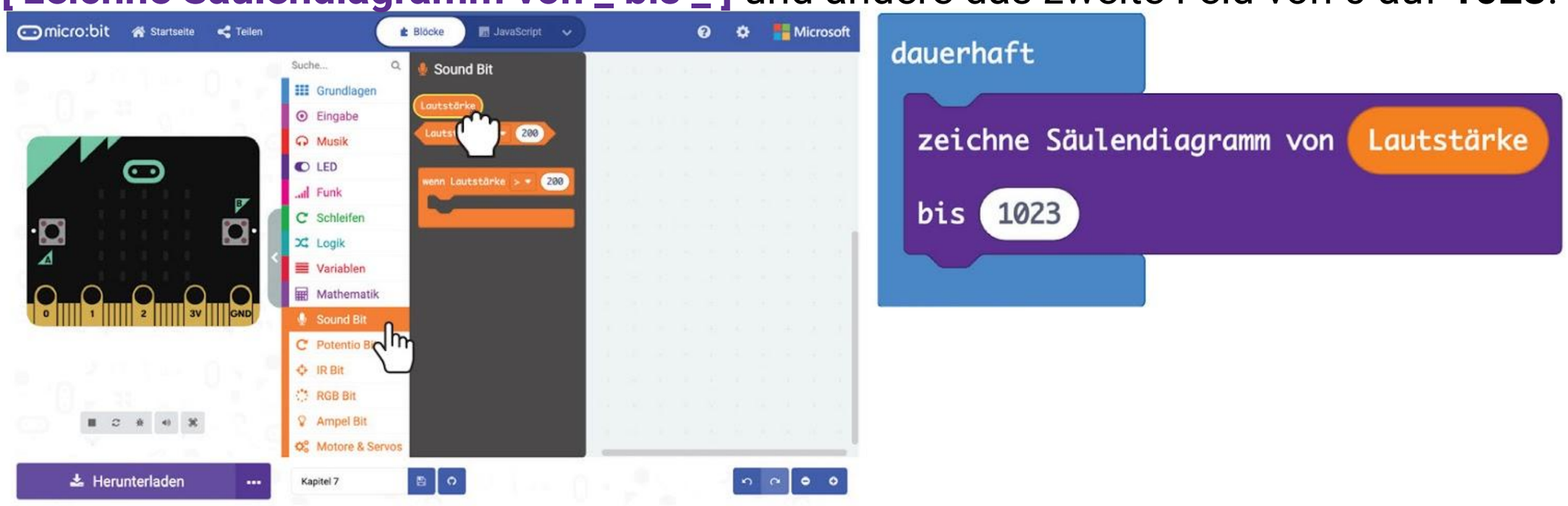
link.cytron.io/edubit-chapter-7

LASS UNS PROGRAMMIEREN!

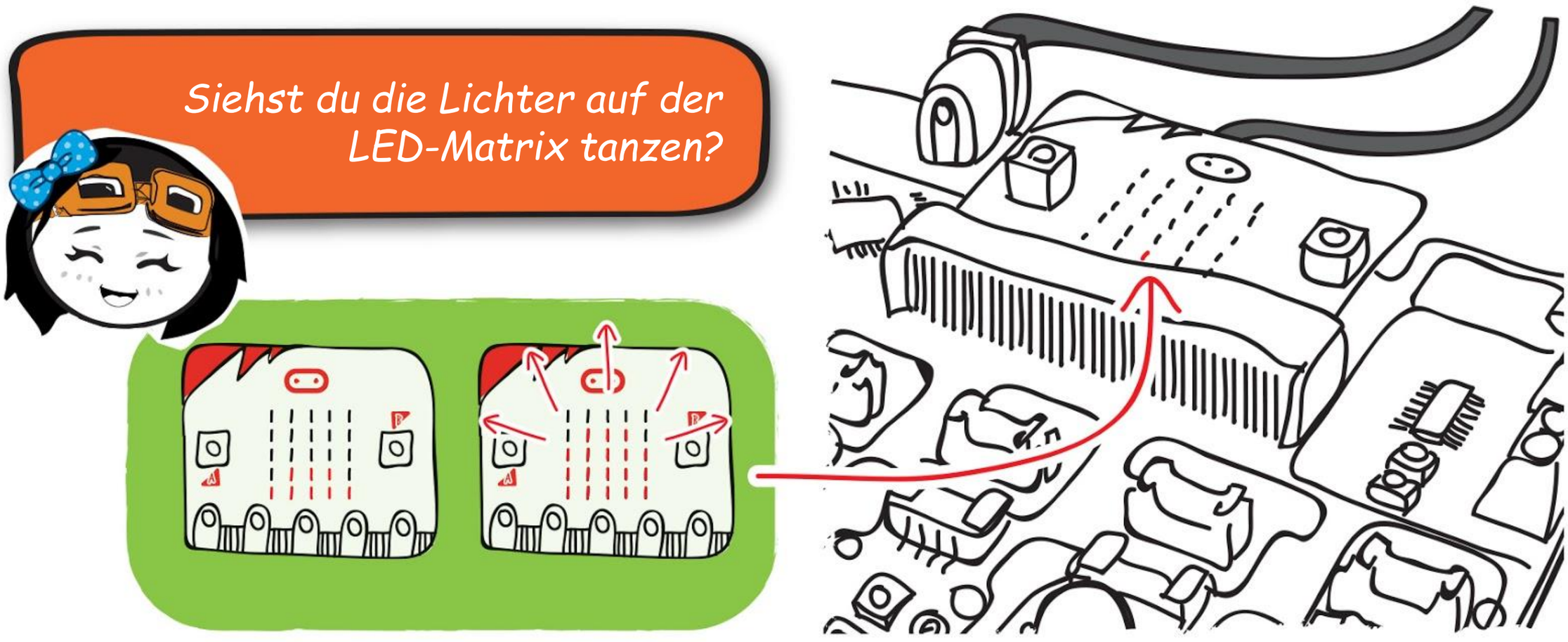
Schritt 1 Erstelle ein neues Projekt im MakeCode Editor und füge die EDU:BIT-Erweiterung hinzu (siehe Seite 40). Klicke auf **[LED]** und wähle den Block **[zeichne Säulendiagramm von _ bis _]**. Lege den Block in **[dauerhaft]**.



Schritt 2 Klicke in **[Sound Bit]** auf **[Lautstärke]**. Lege den Block in das erste Feld von **[zeichne Säulendiagramm von _ bis _]** und ändere das zweite Feld von 0 auf **1023**.

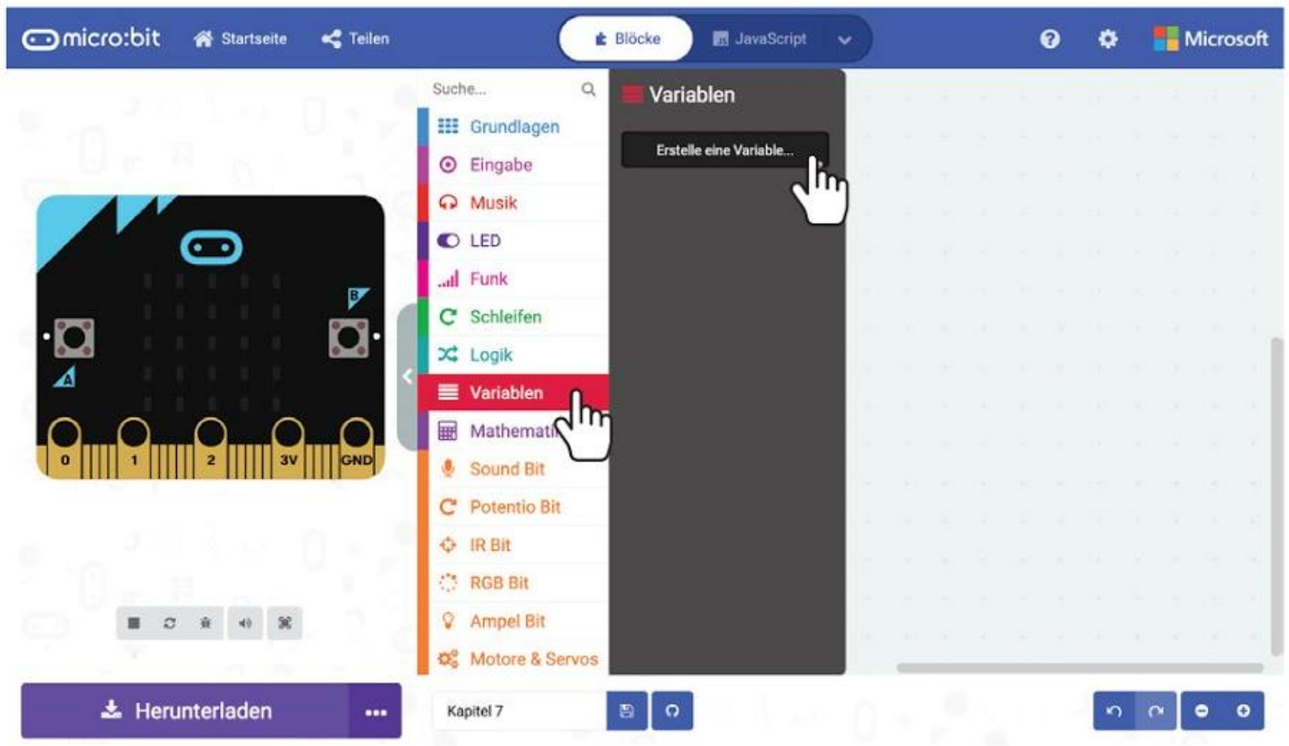


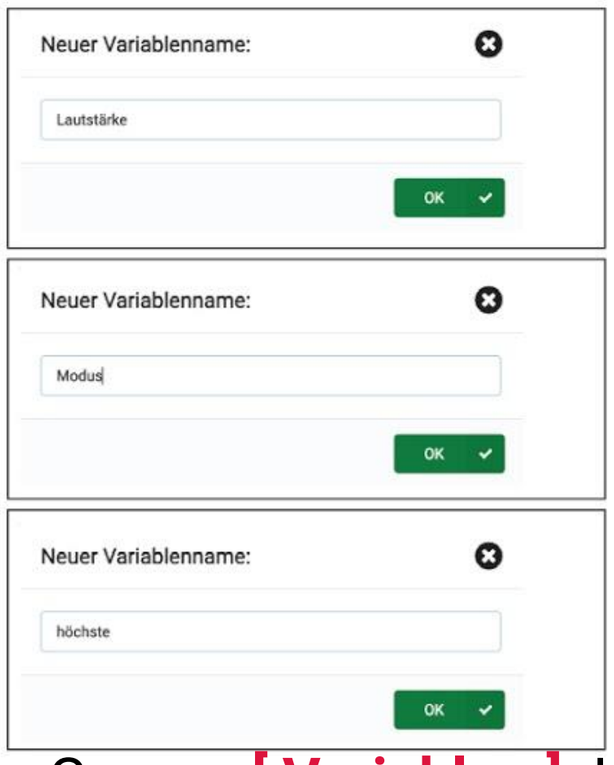
Schritt 3 Übertrage den Code auf EDU:BIT. Beobachte, was die LED-Matrix tut, wenn du in die Hände klatschst oder mit den Fingern auf dem Tisch trommelst.



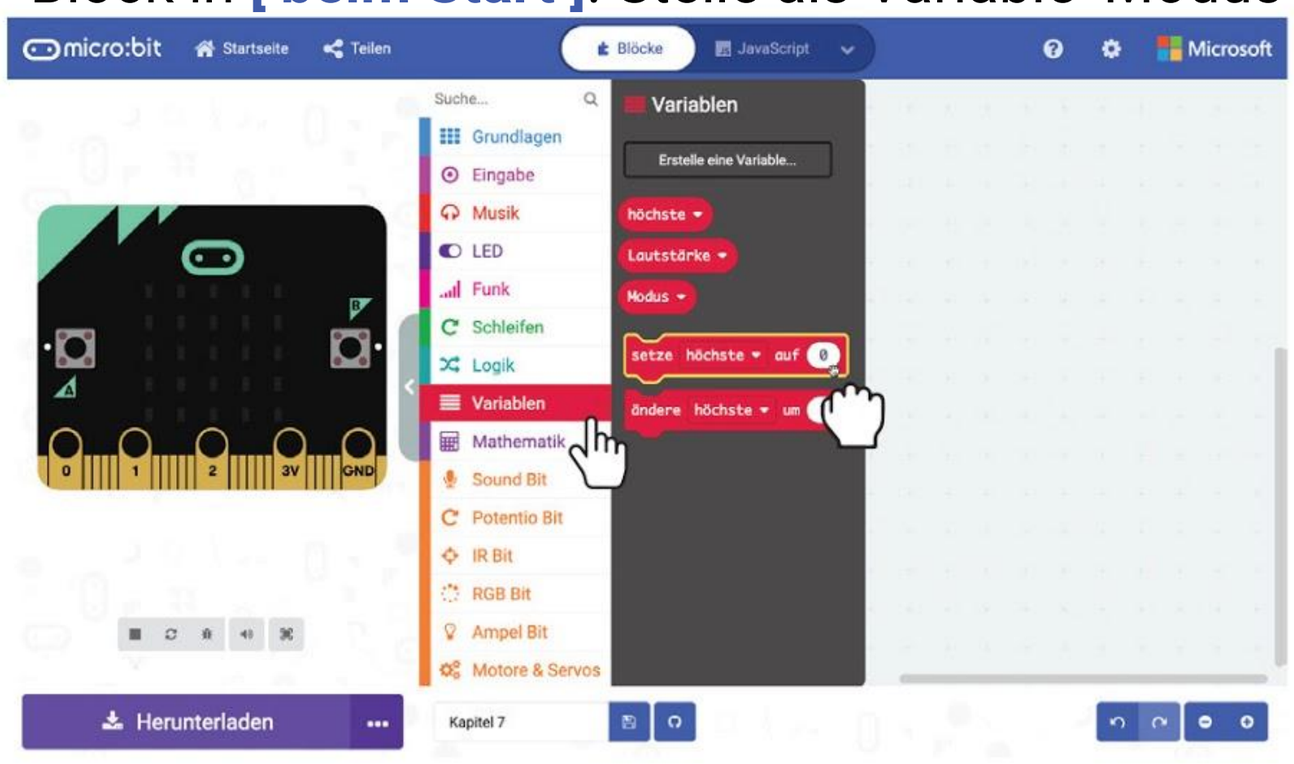
Jetzt wandeln wir EDU:BIT in ein Applaus-o-meter um.

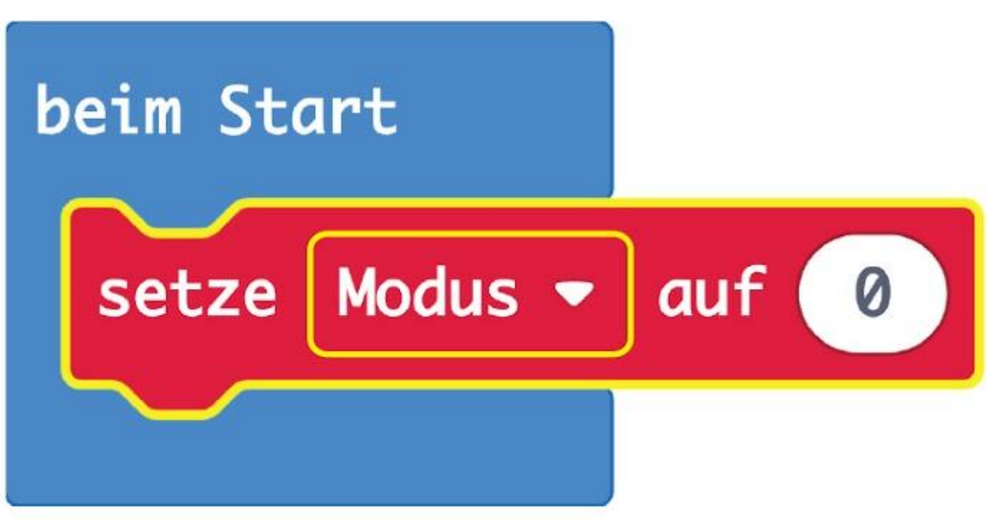
Schritt 4 Erstelle ein neues Projekt und füge die EDU:BIT-Erweiterung hinzu. Klicke in **[Variablen]** auf **[Erstelle eine Variable]**. Schreibe **‘Modus’** in das Dialogfeld und klicke OK. Erstelle noch zwei Variablen mit den Namen **‘Lautstärke’** und **‘höchste’**.



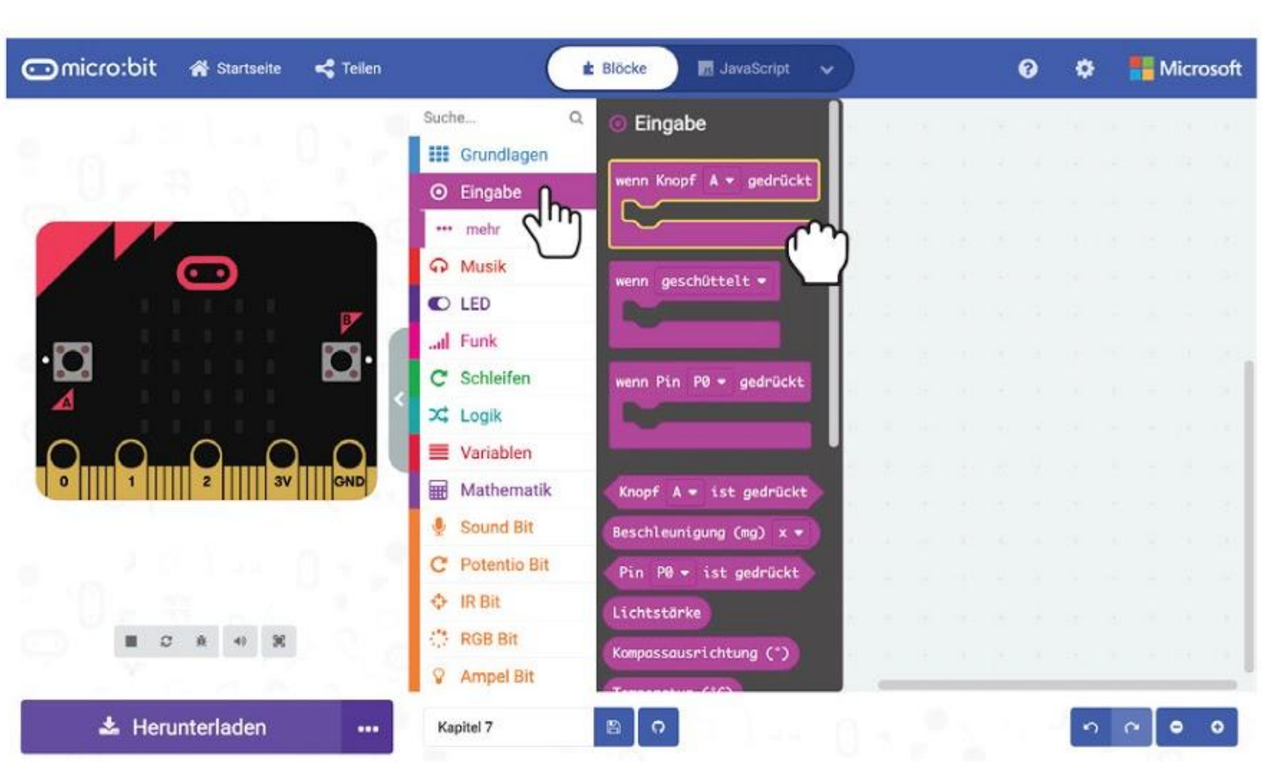


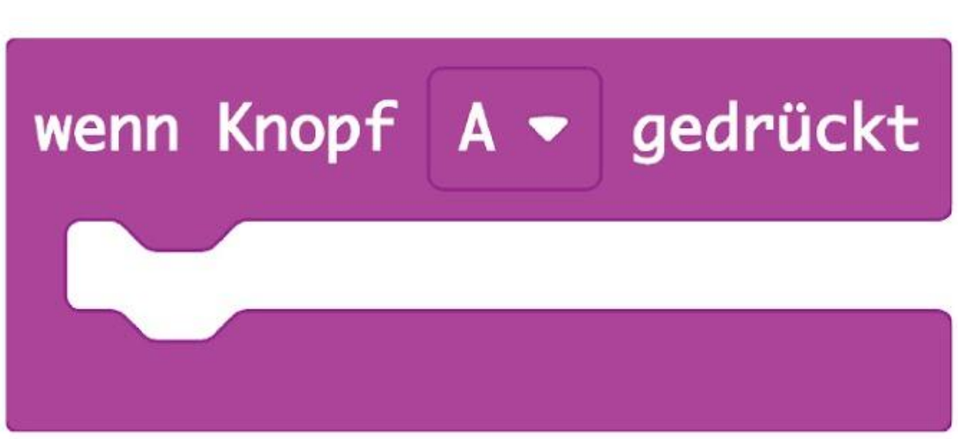
Schritt 5 Nimm einen Block **[setze _ auf _]** aus der Gruppe **[Variablen]**. Lege den Block in **[beim Start]**. Stelle als Variable ‘Modus’ ein.



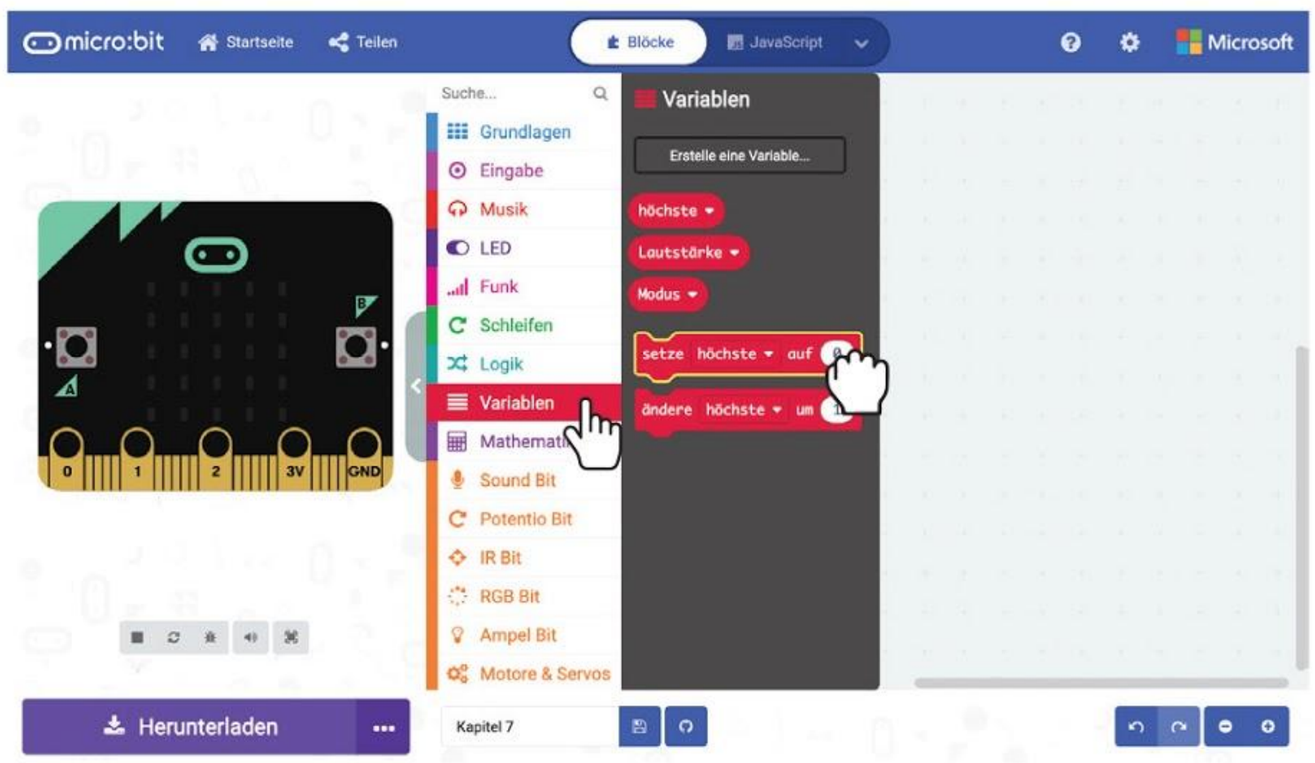


Schritt 6 Klicke unter **[Eingabe]** auf den Block **[wenn Knopf _ gedrückt]**.

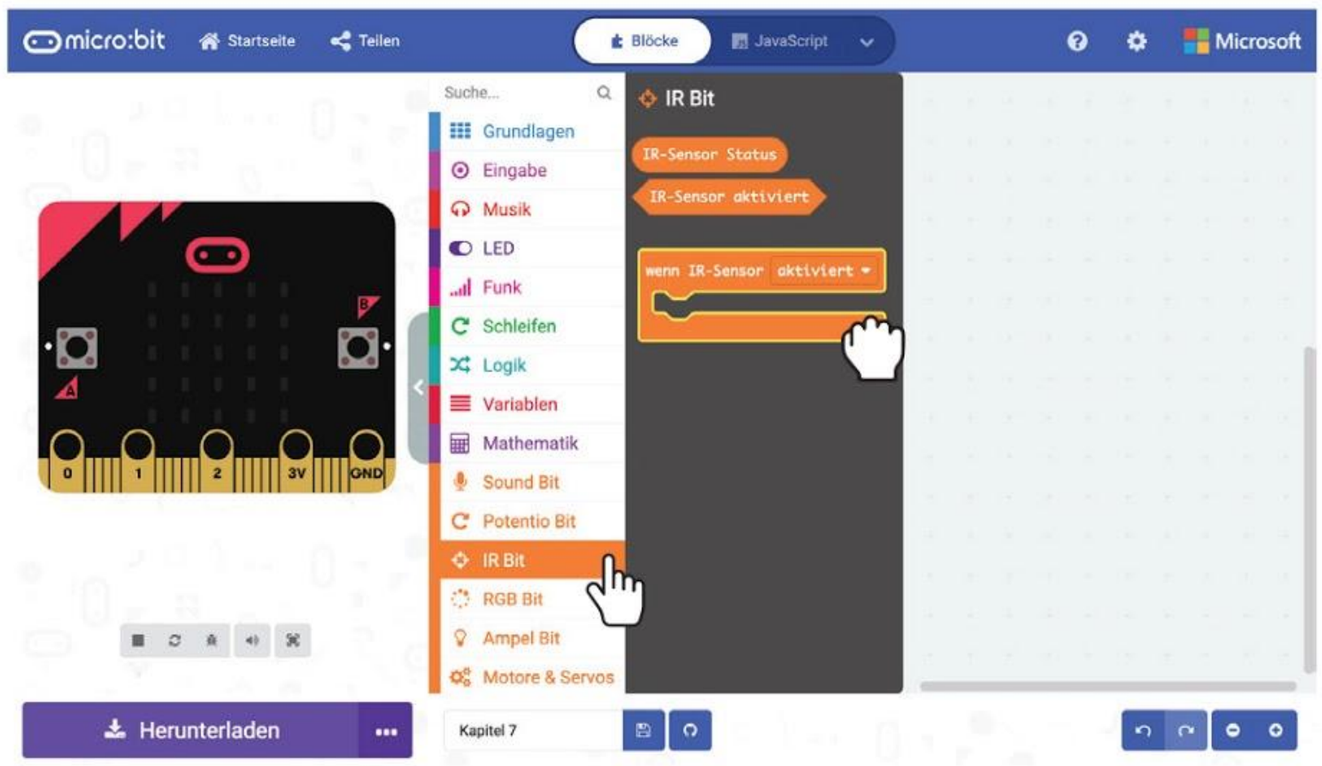




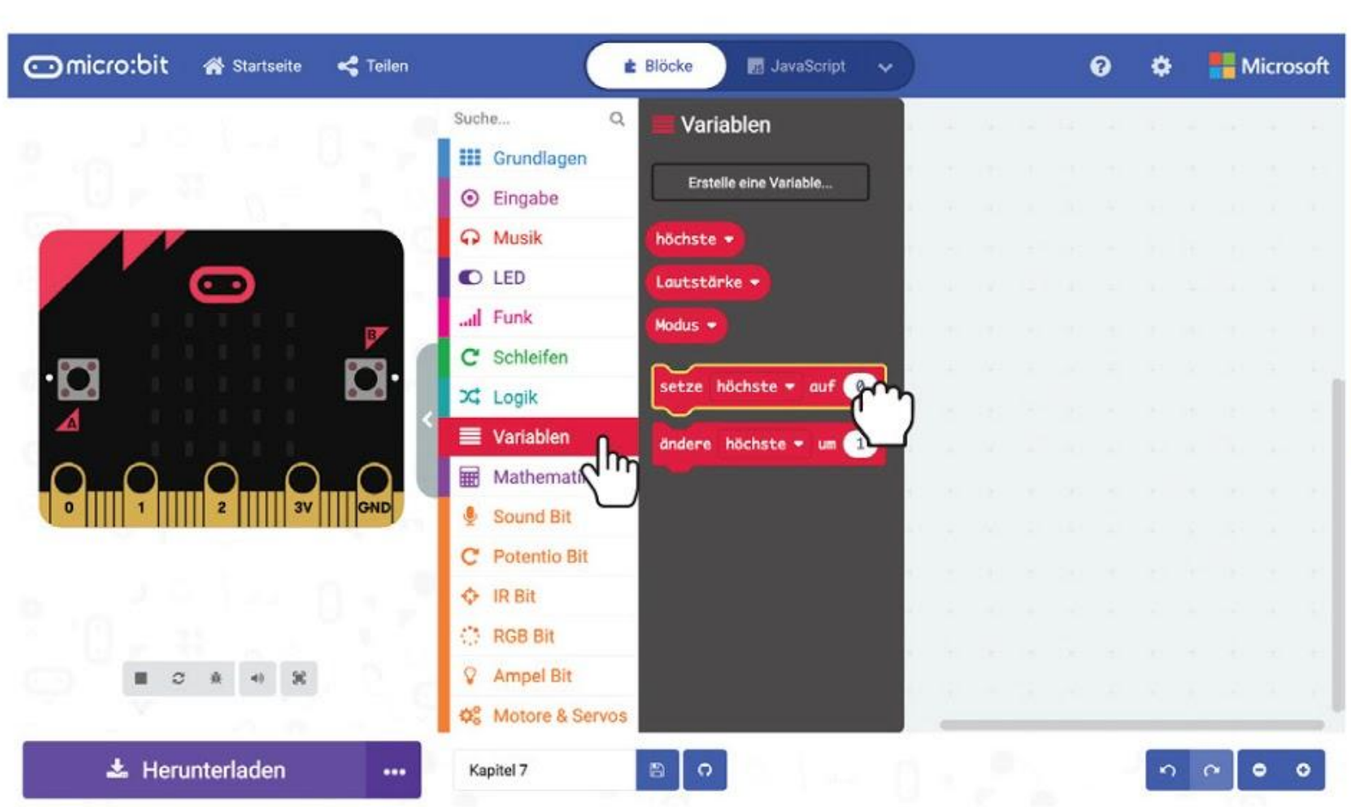
Schritt 7 Nimm zwei **[setze _ auf _]**-Blöcke aus der **[Variablen]**-Gruppe und lege sie in den Block **[wenn Knopf A gedrückt]**. Setze die erste Variable auf **‘Modus’** und den Wert auf 1, und die zweite Variable auf **‘höchste’** und den Wert auf 0.

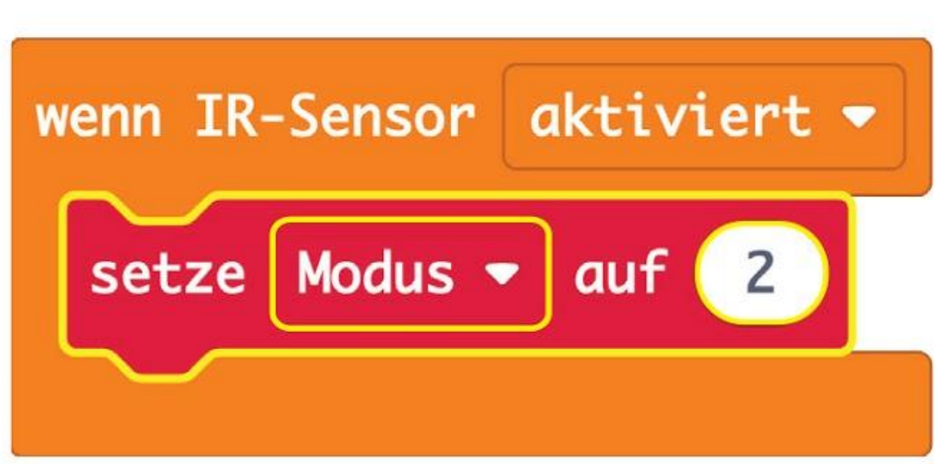


Schritt 8 Klicke auf die Gruppe **[IR Bit]** und den Block **[wenn IR-Sensor aktiviert]**.

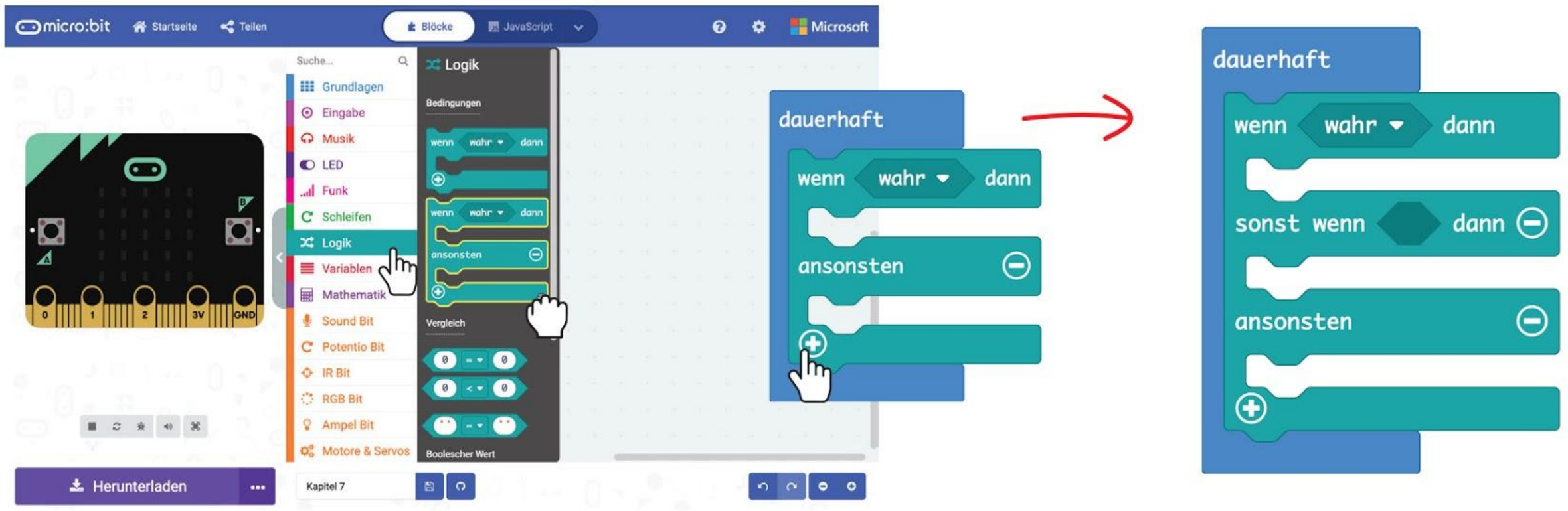


Schritt 9 Klicke auf **[Variablen]** und wähle **[setze _ auf _]**. Lege den Block in **[wenn IR-Sensor aktiviert]** und ändere die Variable auf **‘Modus’** und die Zahl auf **2**.

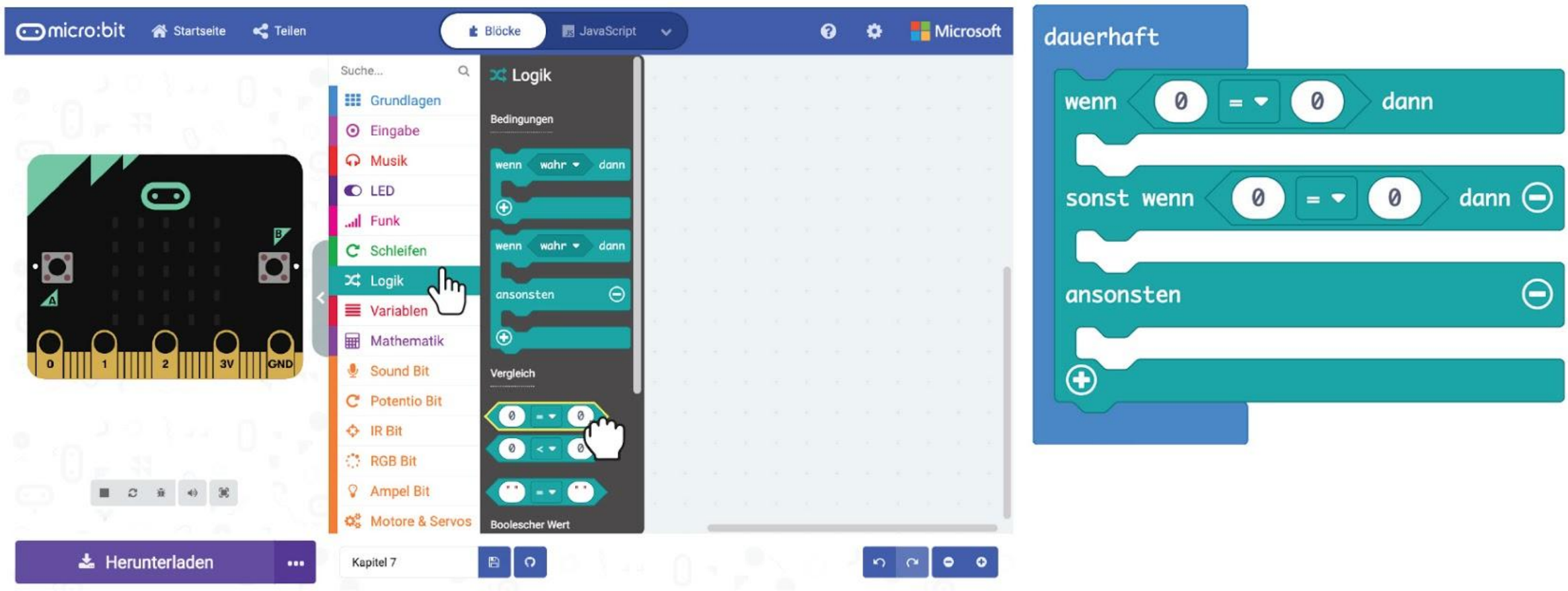




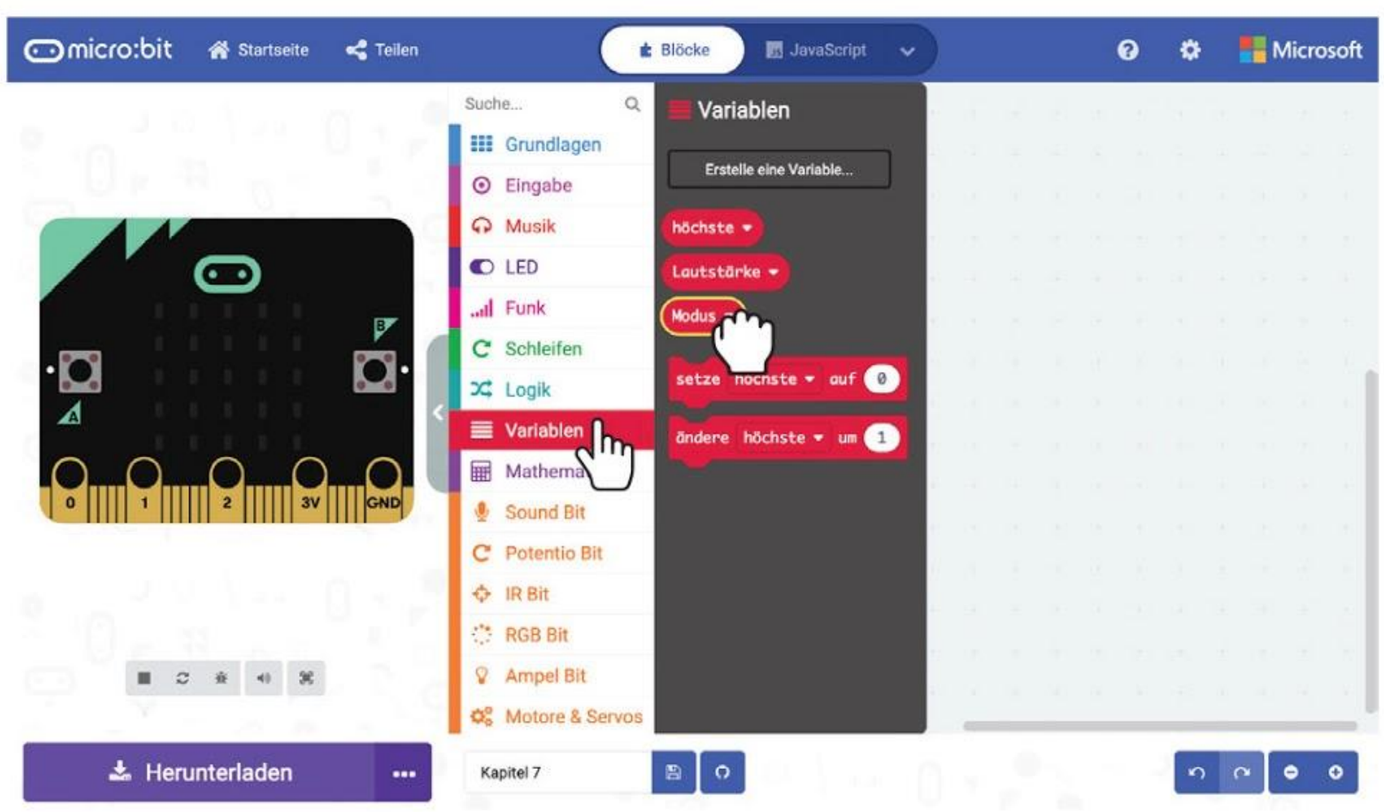
Schritt 10 Klicke auf die Gruppe **[Logik]** und auf den Block **[wenn-dann-ansonsten]**. Lege den Block in **[dauerhaft]**. Klicke auf das (+)-Symbol um eine Bedingung hinzuzufügen.



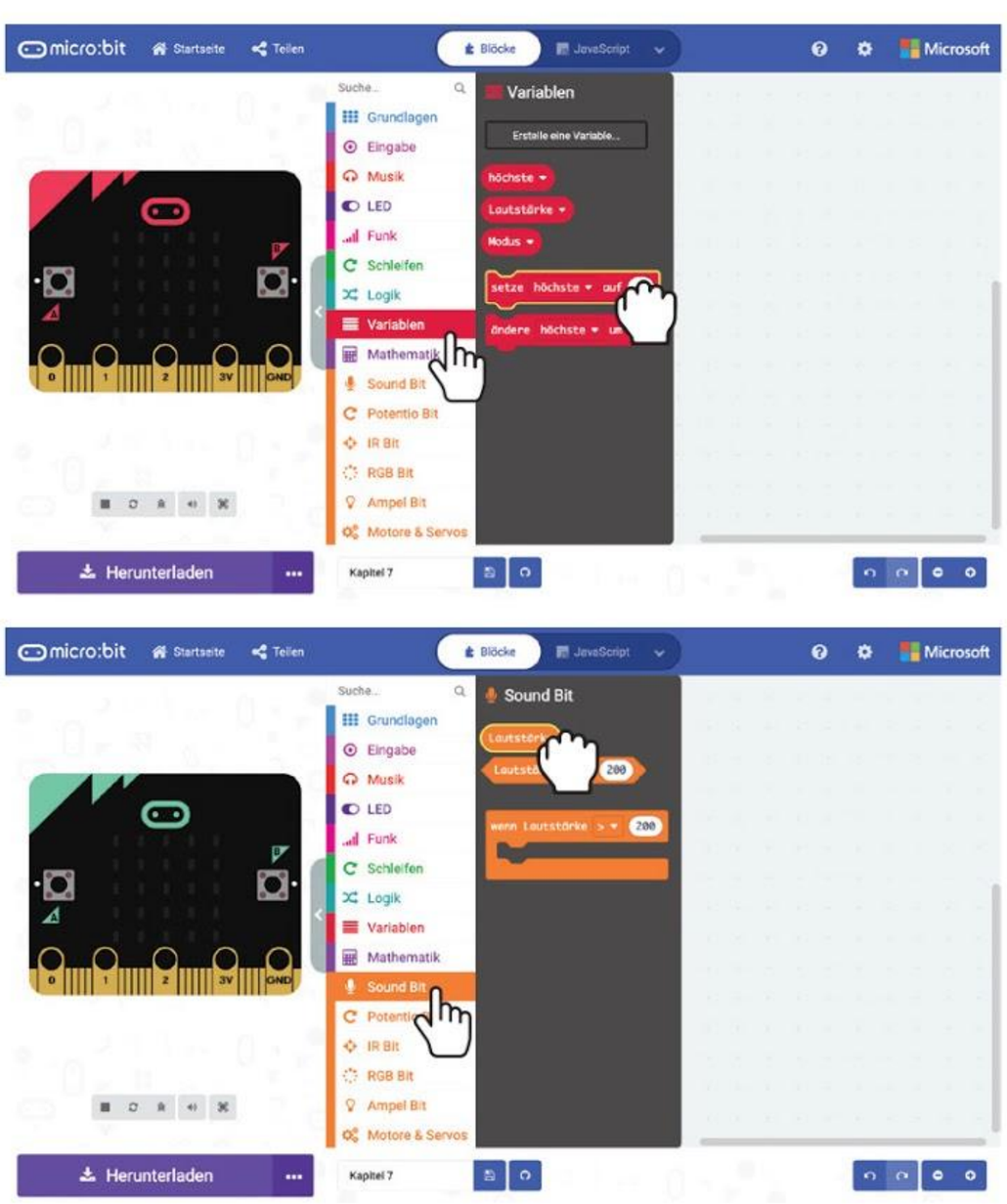
Schritt 11 Klicke unter **[Logik]** auf den Vergleichs-Block **[_ = _]**. Dupliziere den Block und ziehe die Blöcke in den **[wenn-dann-ansonsten]**-Block.

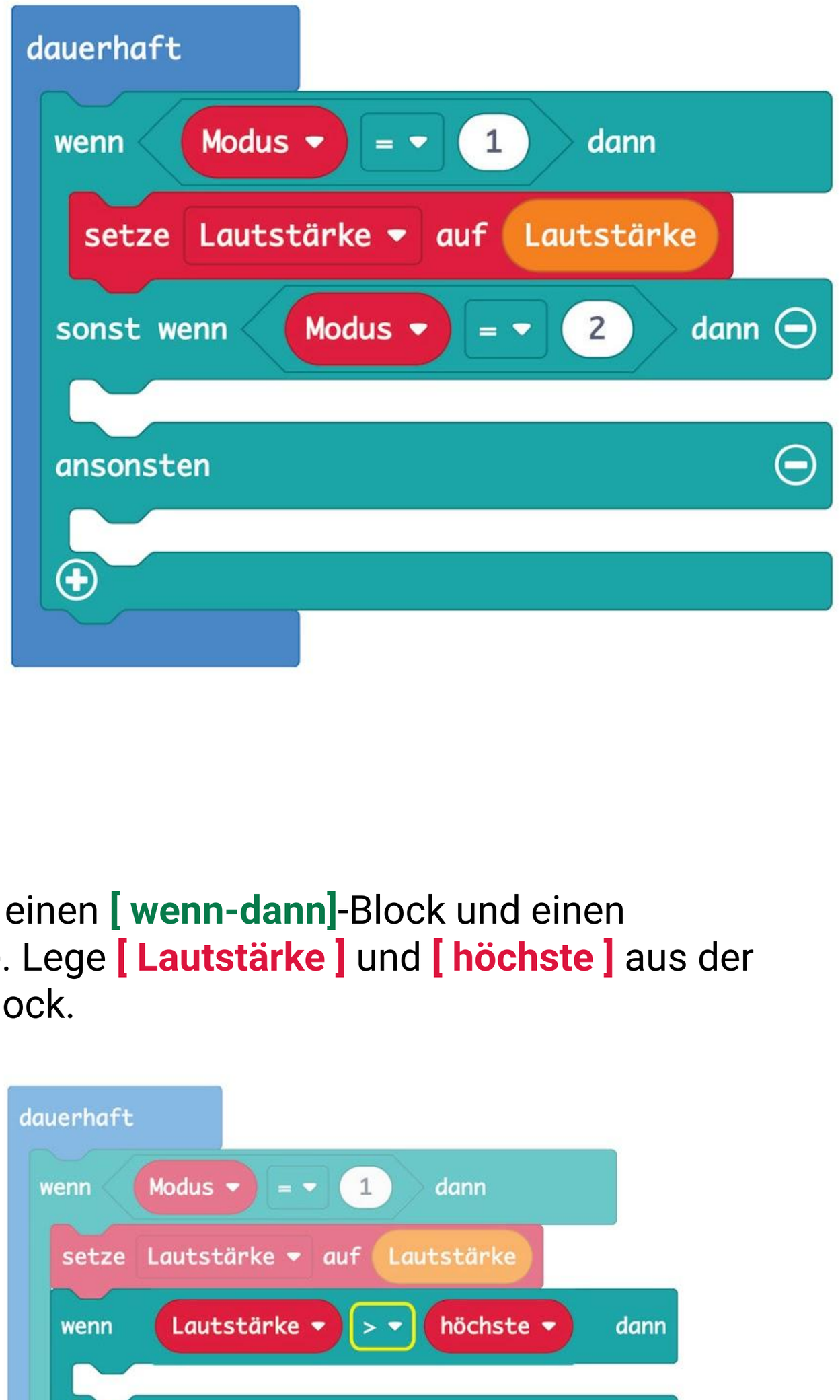


Schritt 12 Nimm zweimal **[Modus]** aus der Gruppe **[Variablen]** und lege die Blöcke in die linken Felder in den Vergleichs-Blöcken. Setze die Zahlen jeweils auf **1** und **2.**

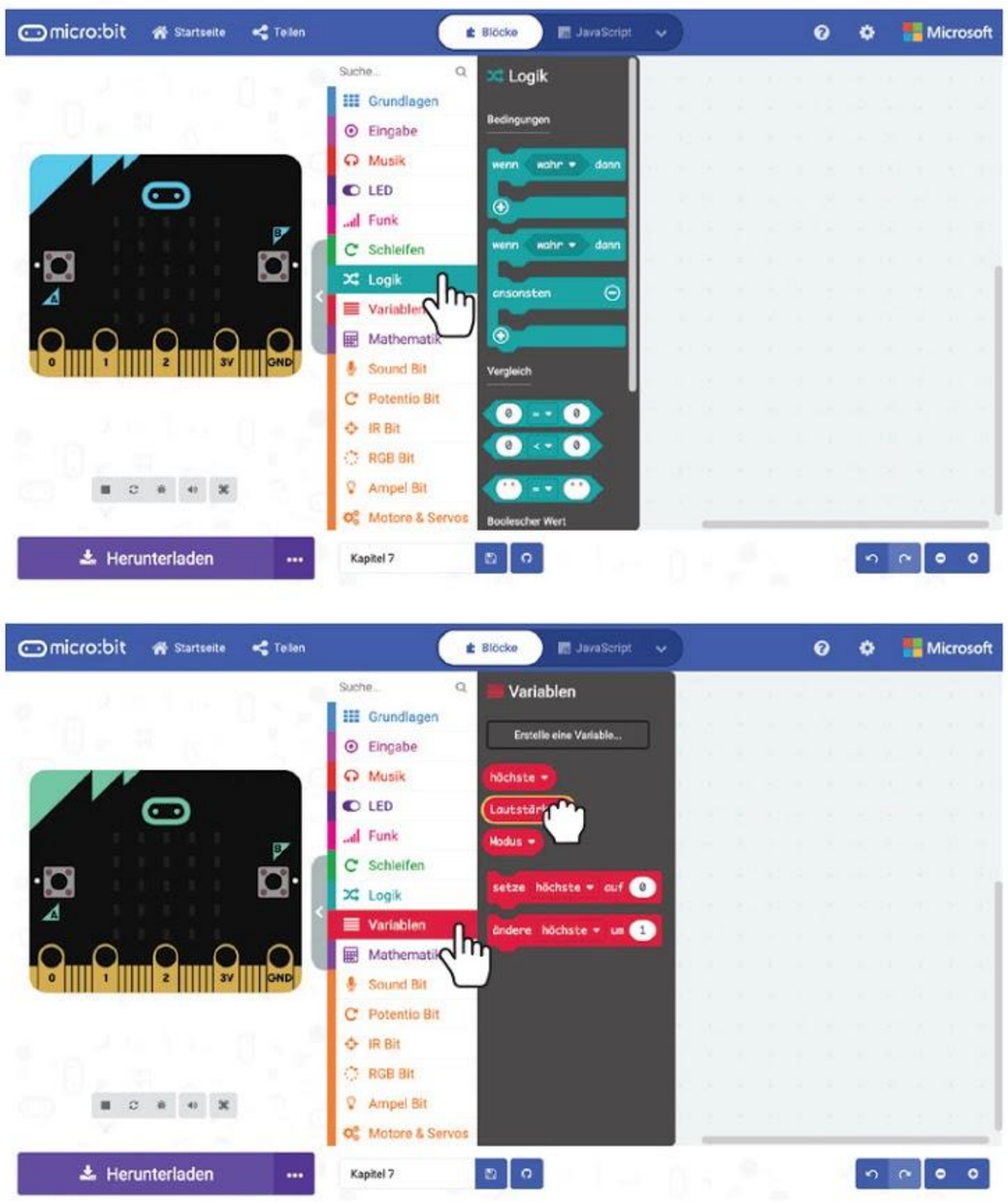


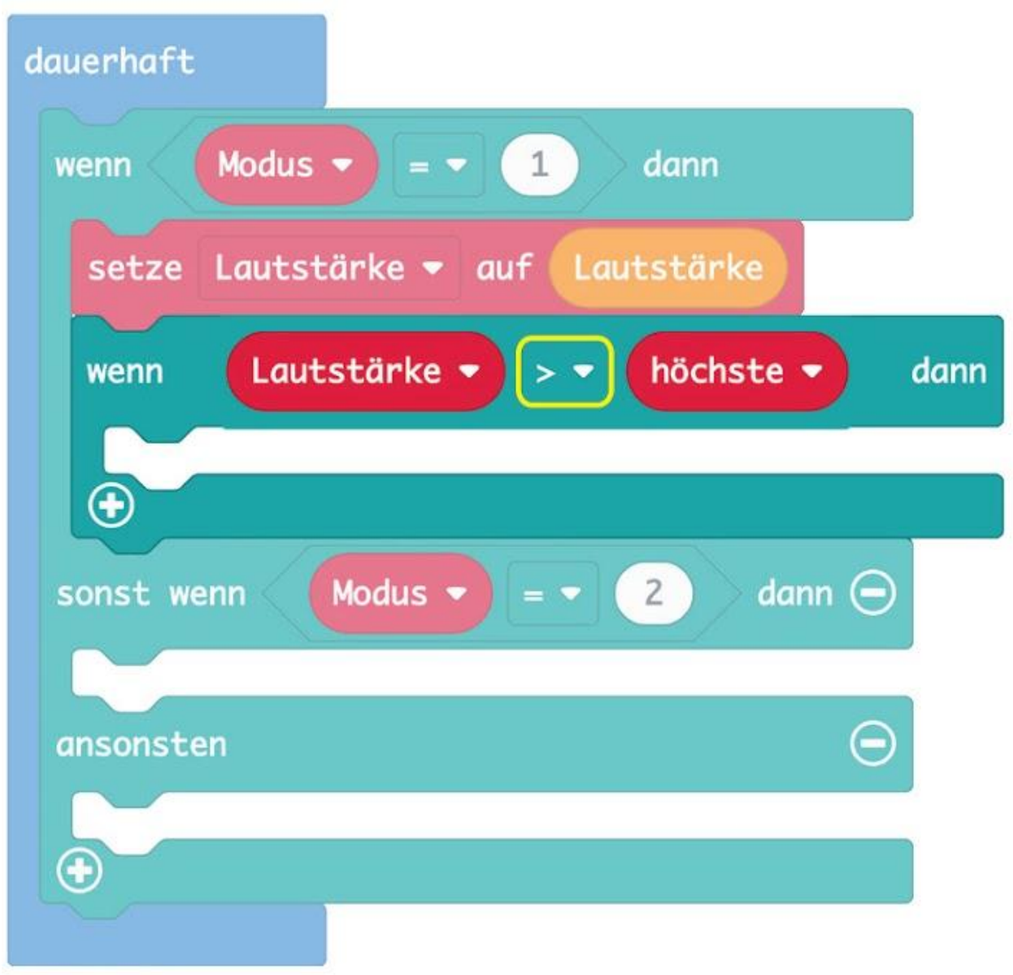
Schritt 13 Nimm einen **[setze _ auf _]**-Block aus der Gruppe **[Variablen]** und lege ihn in den Block **[wenn-dann-ansonsten]**. Setze die Variable auf **‘Lautstärke’** und lege den Block **[Lautstärke]** aus **[Sound Bit]** in den Zahlen-Slot.





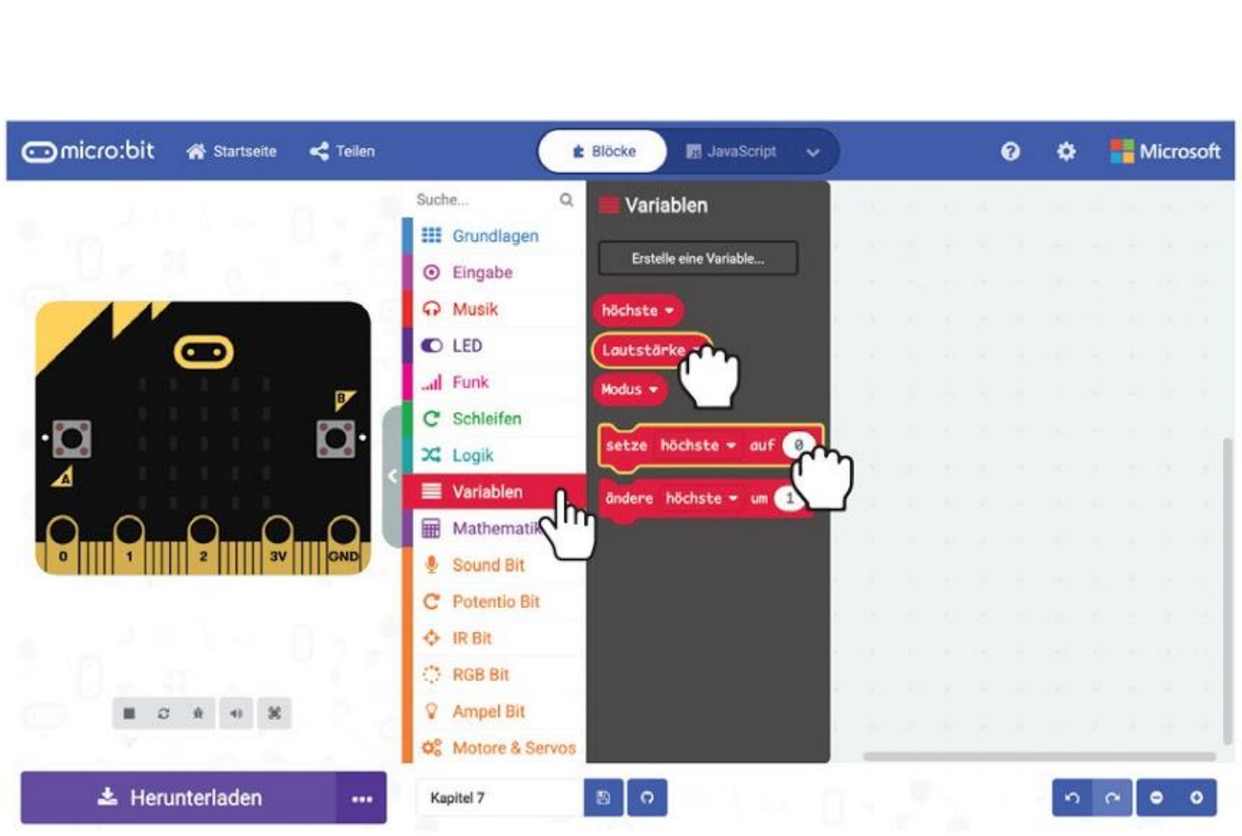
Schritt 14 Klicke auf **[Logik]** und wähle einen **[wenn-dann]**-Block und einen **[_ = _]**-Block. Ändere das Symbol = zu **>**. Lege **[Lautstärke]** und **[höchste]** aus der Gruppe **[Variablen]** in den Vergleichs-Block.

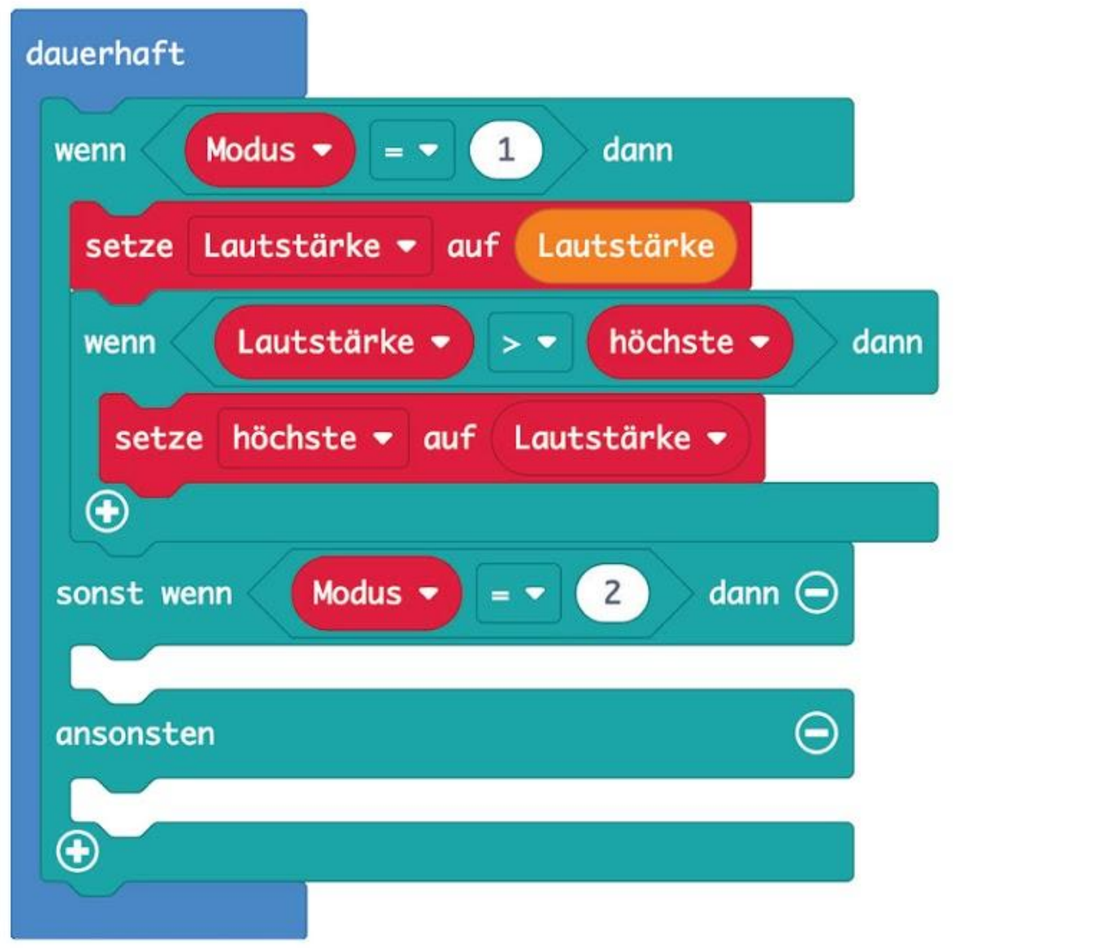




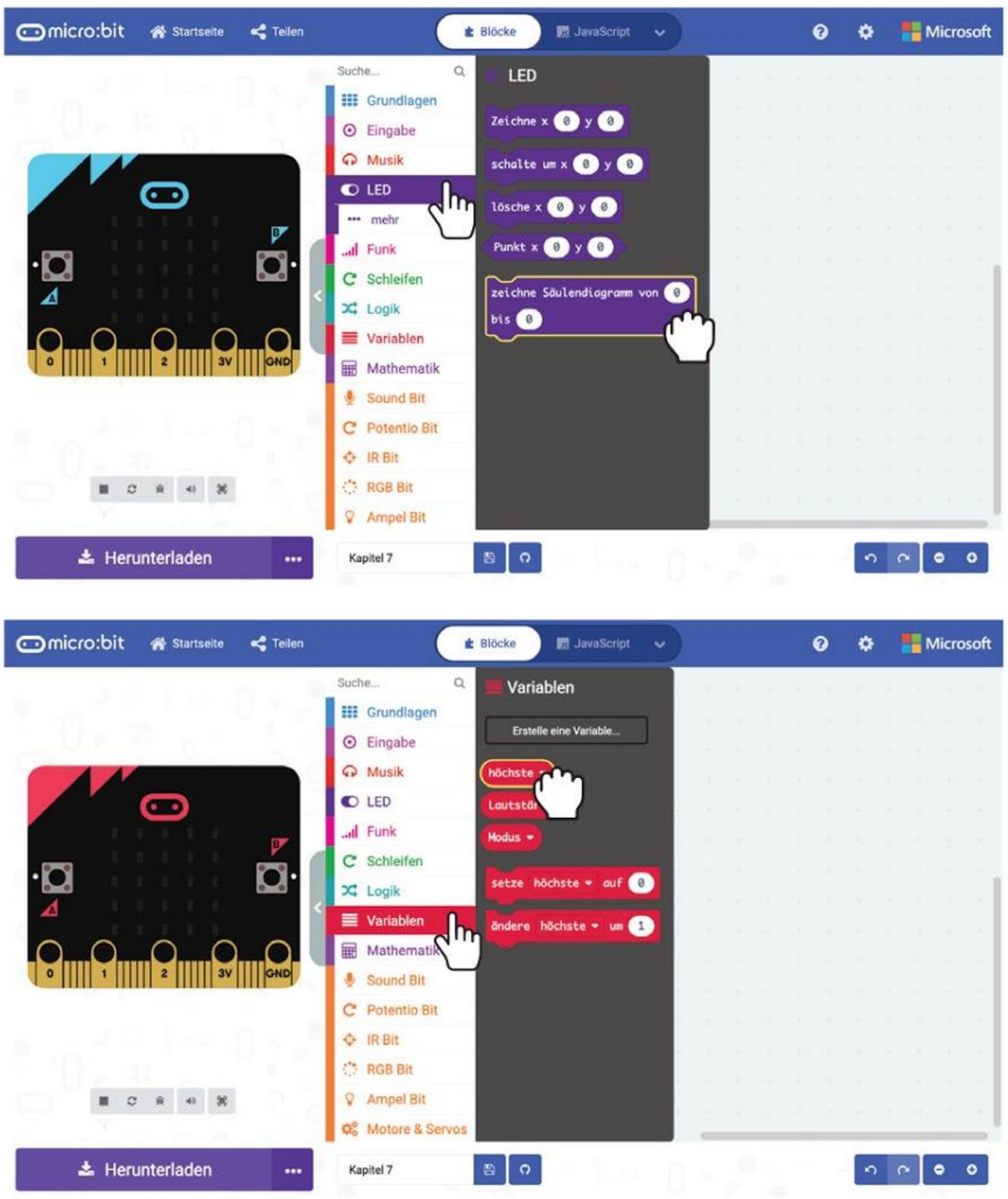
KAPITEL 7 : Applaus, Applaus!

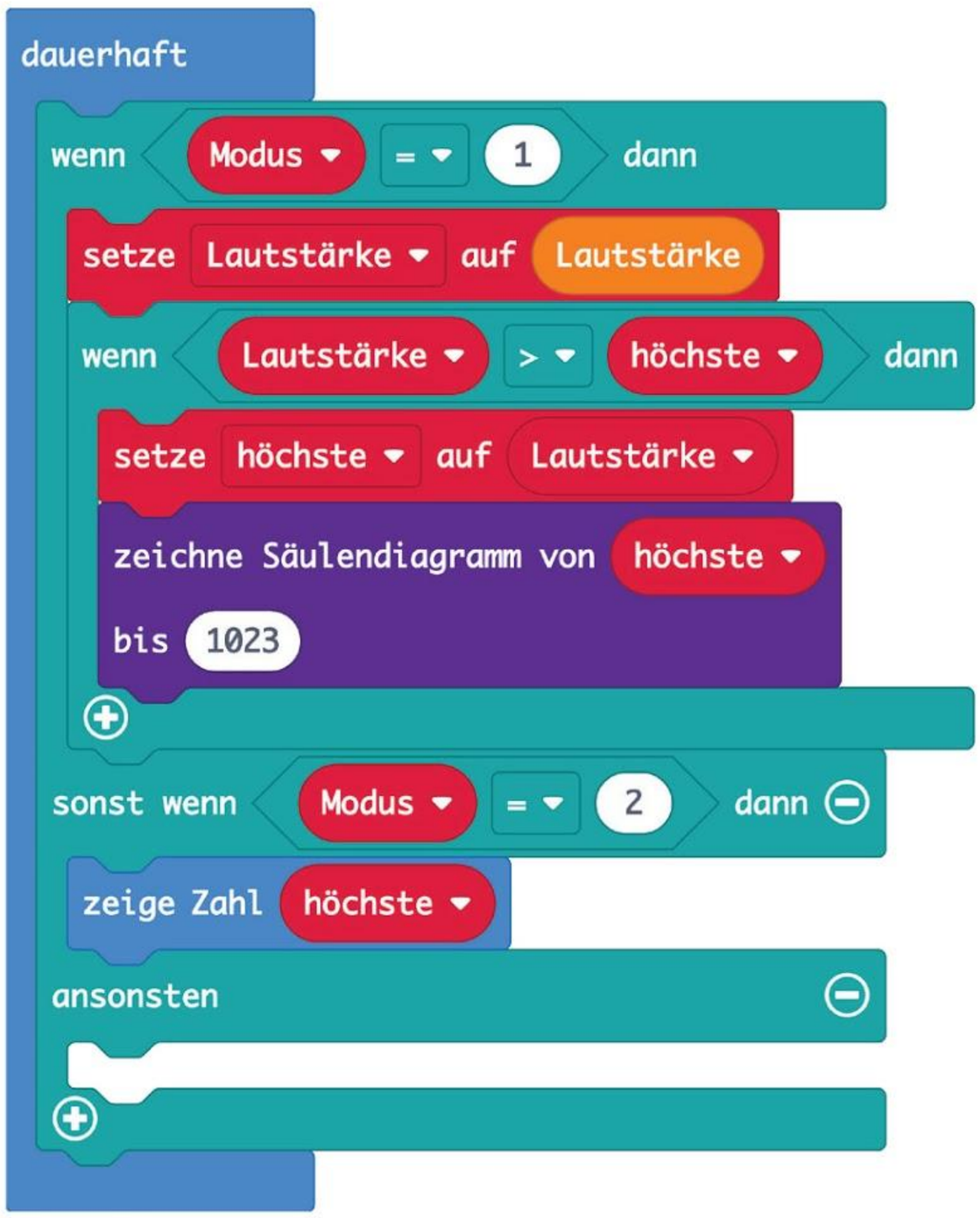
Schritt 15 Klicke auf **[Variablen]** und wähle **[setze _ auf _]** aus. Lege den Block in **[wenn-dann]**. Ziehe **[Lautstärke]** aus der Gruppe **[Variablen]** in den Wert-Bereich.



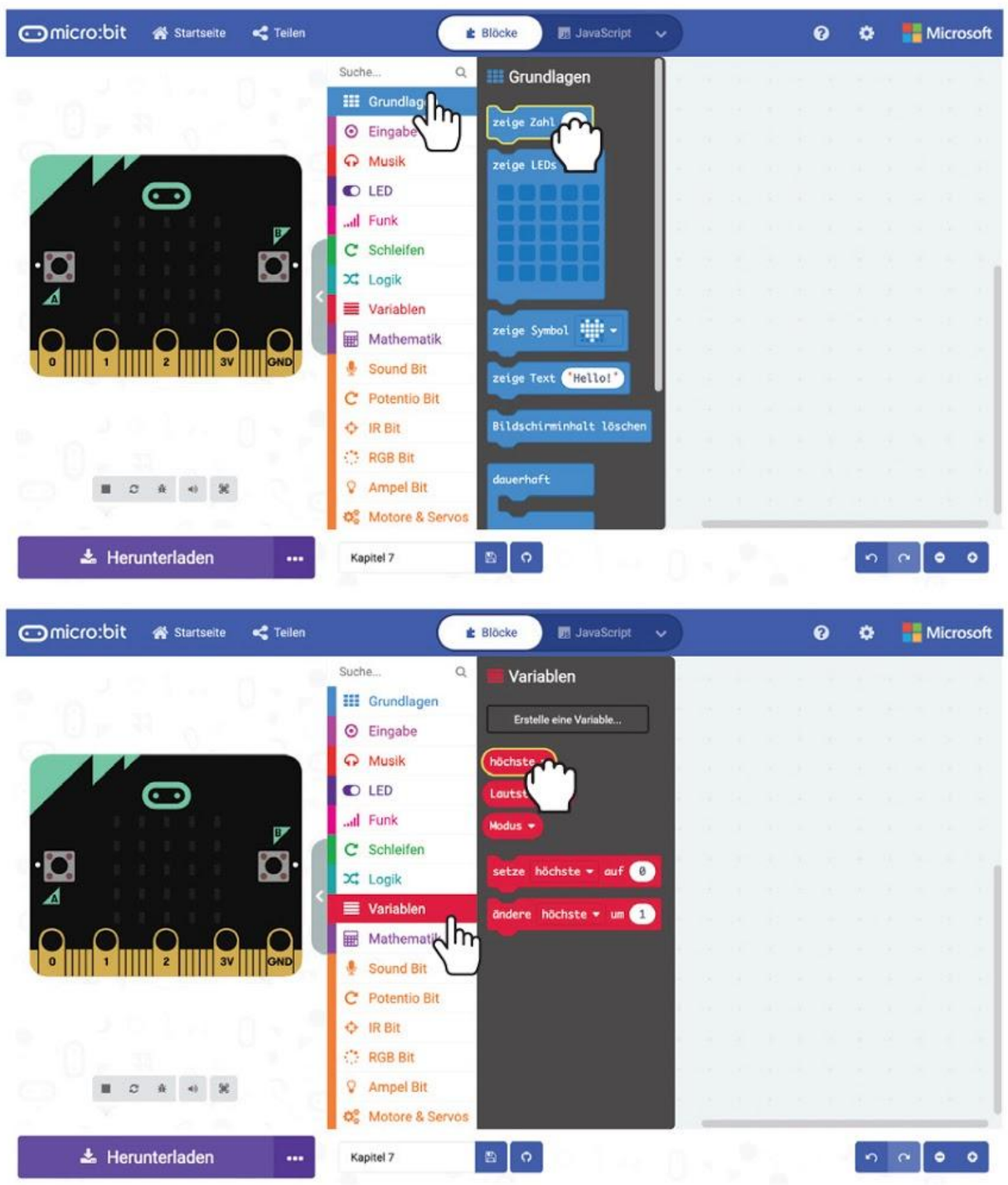


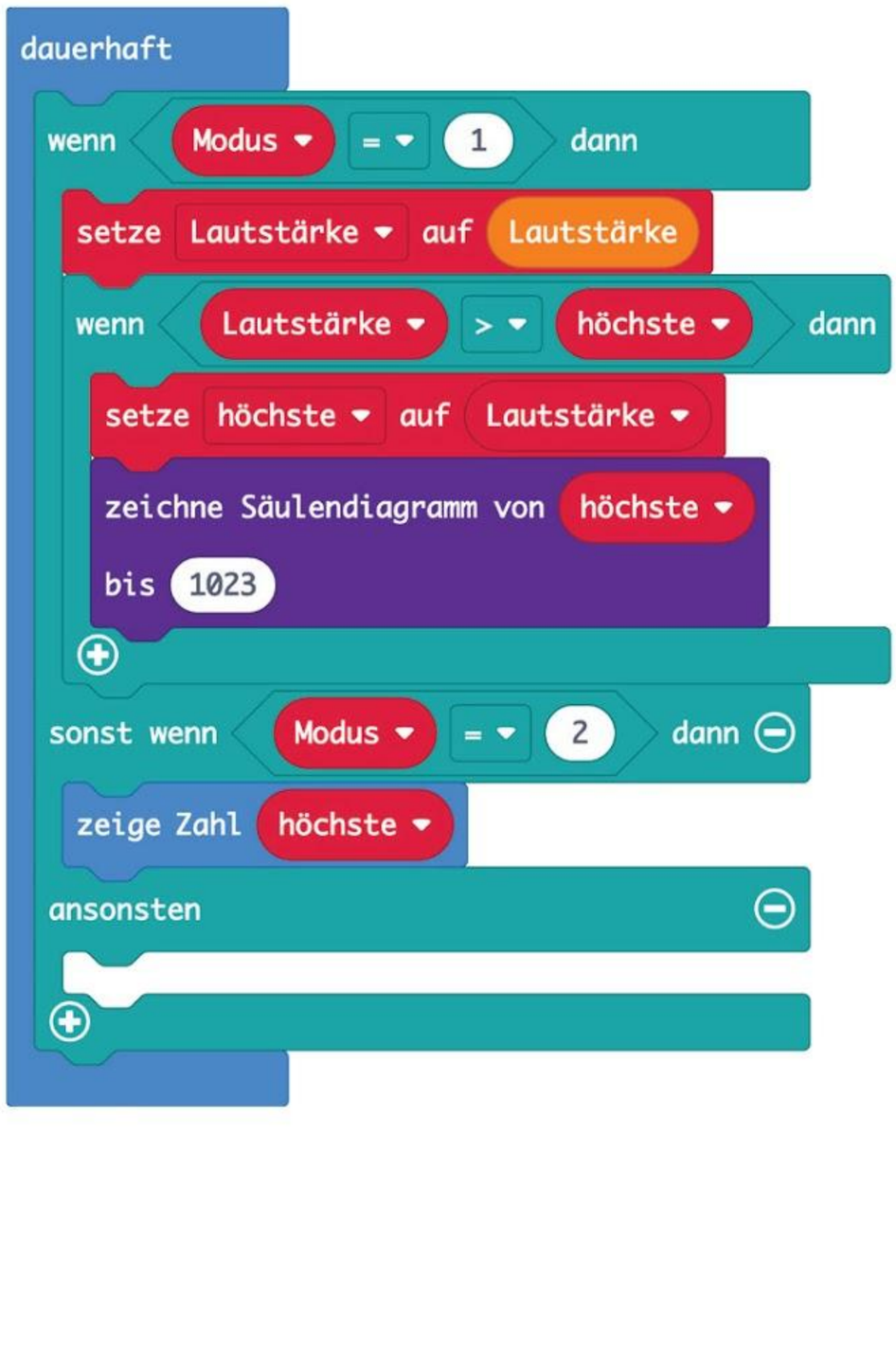
Schritt 16 Klicke auf **[LED]** und wähle den Block **[zeichne Säulendiagramm von _ bis _]**. Klicke in der Gruppe **[Variablen]** auf den Block **[höchste]**. Lege den Block in **[zeichne Säulendiagramm von _ bis _]** und ändere den Wert zu **1023**.



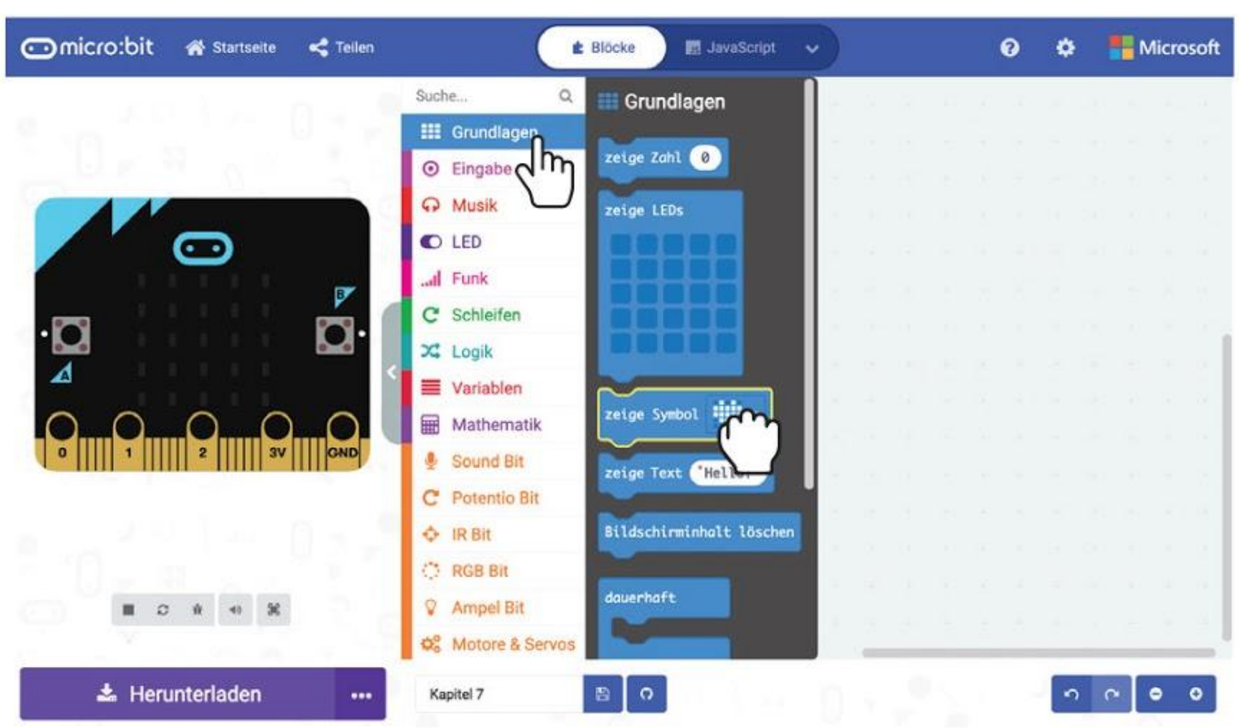


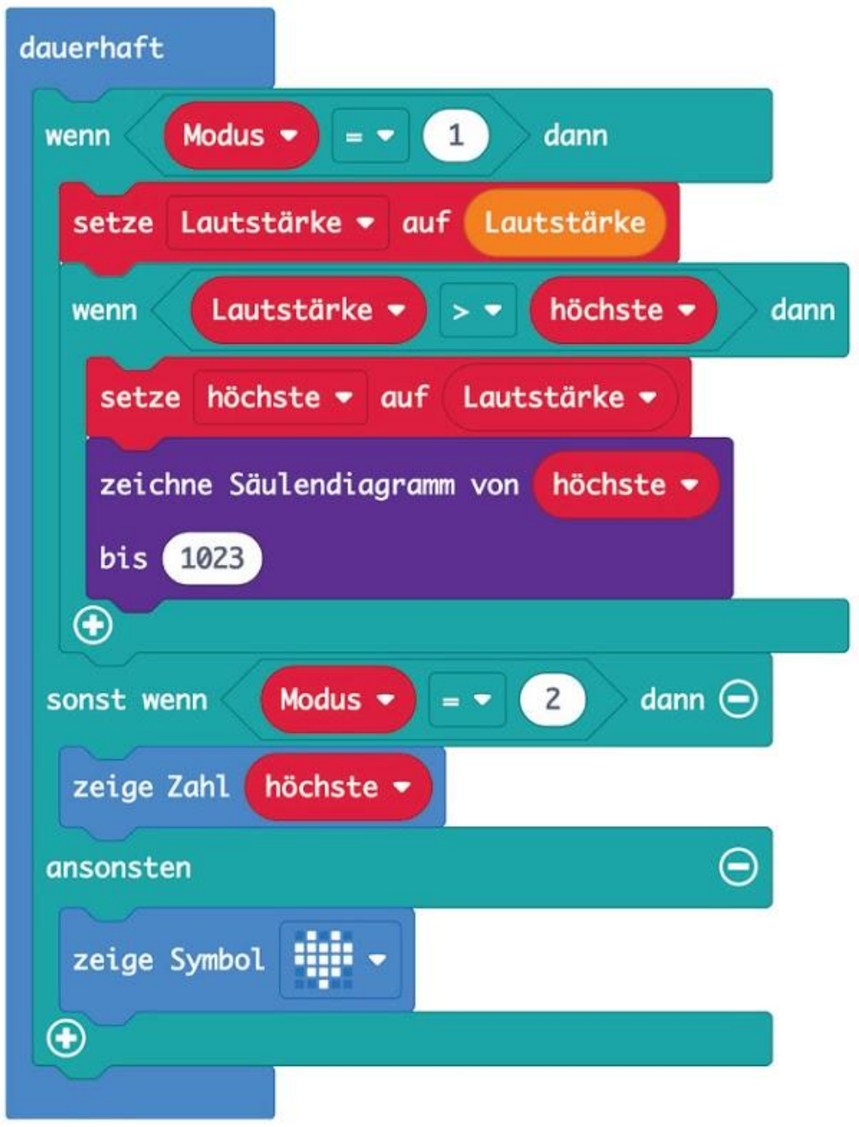
Schritt 17 Klicke unter **[Grundlagen]** auf den Block **[zeige Zahl]**. Lege ihn in den zweiten Code-Bereich des Blockes **[wenn-dann-ansonsten]**. Lege **[höchste]** aus **[Variablen]** in den Block **[zeige Zahl]**.



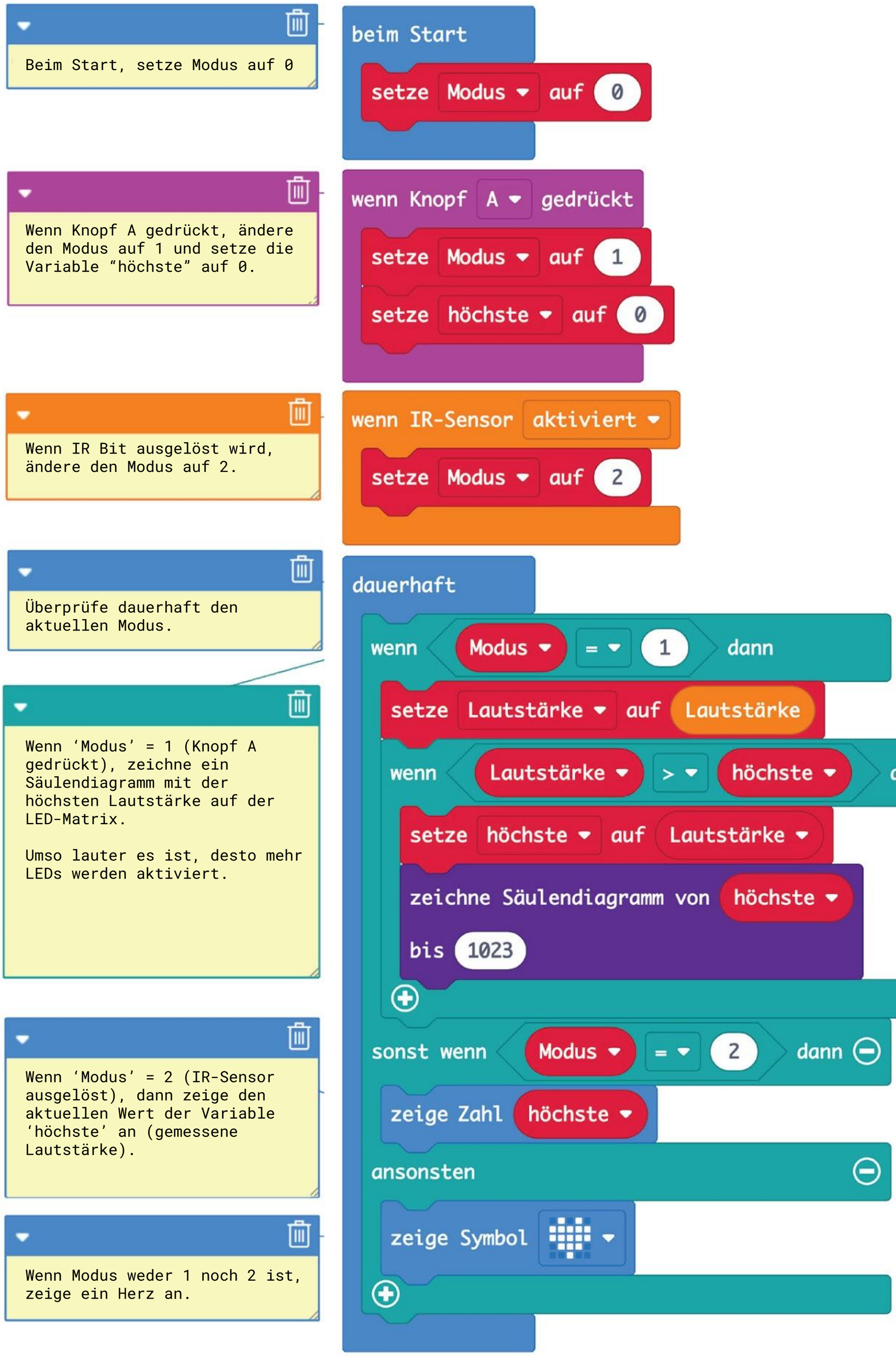


Schritt 18 Klicke in der Gruppe **[Grundlagen]** auf den Block **[zeige Symbol]**. Lege ihn in den letzten Bereich des Blocks **[wenn-dann-ansonsten]**.





Hier ist der ganze Code:

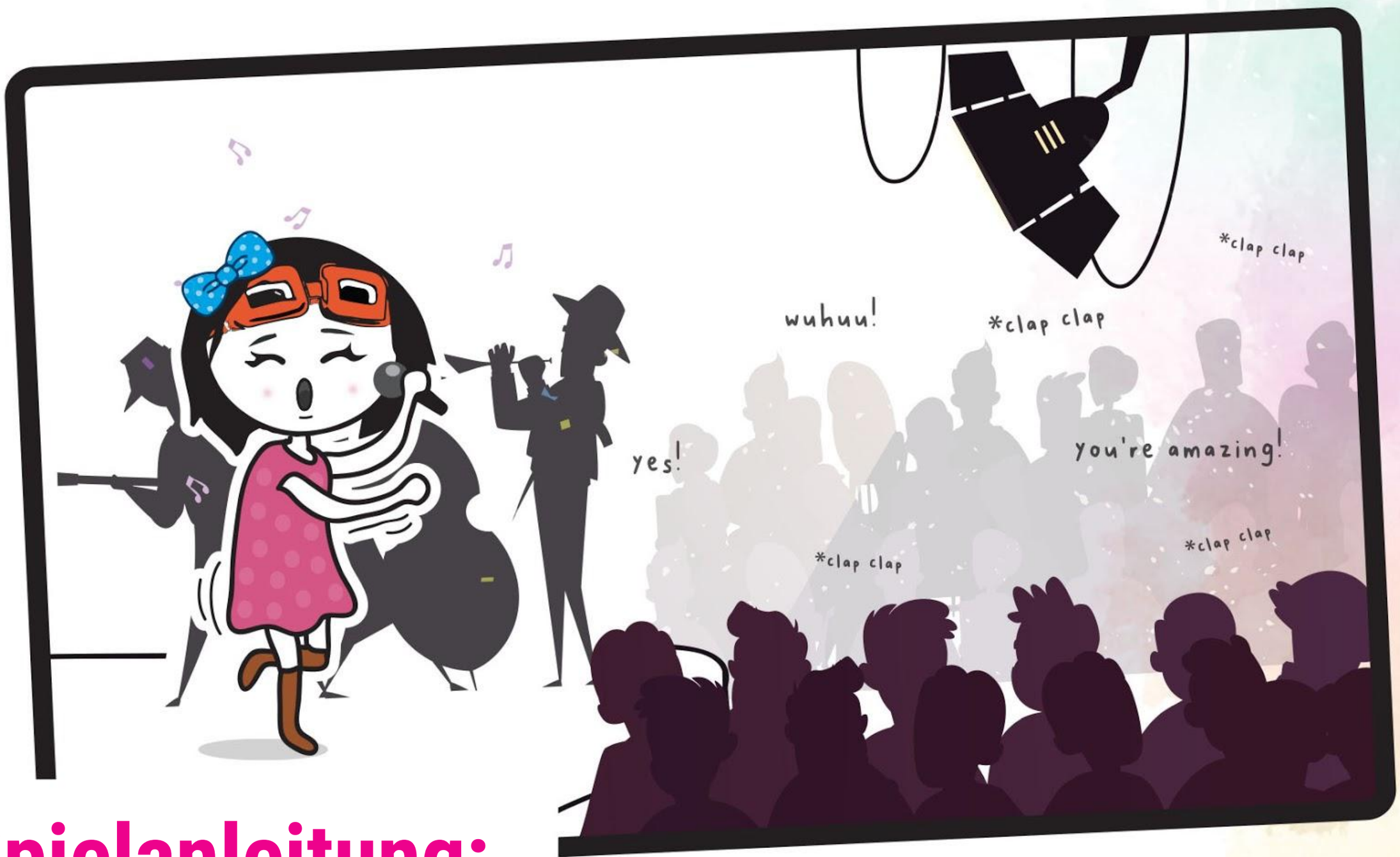


Ich hab's!

Schritt 19 Übertrage den Code auf deinen EDU:BIT und schon hast du ein Applaus-o-meter für deine Talentshow.

Spielen wir!

Applaus, Applaus!



Spielanleitung:

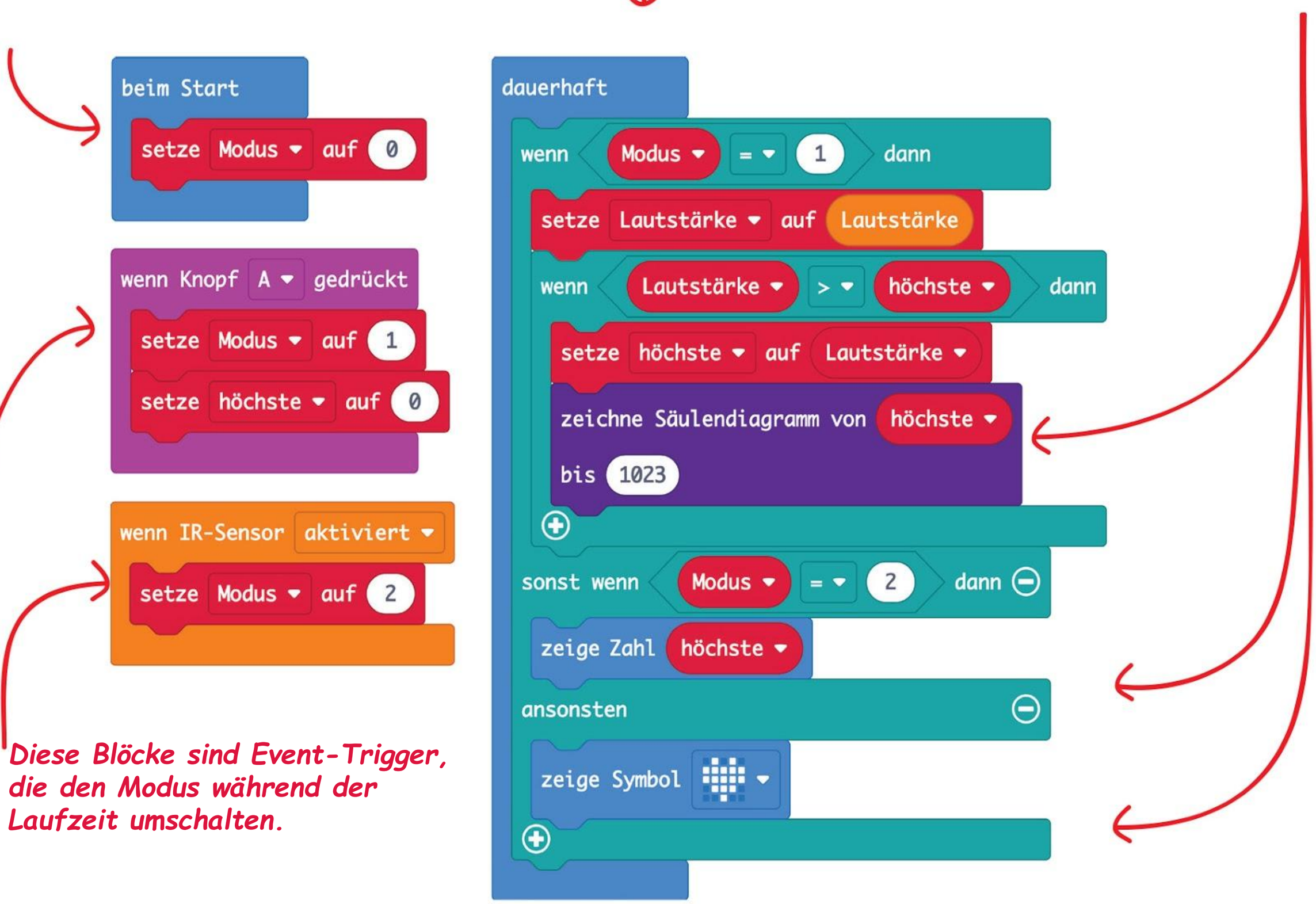
- Alle, die teilnehmen, kriegen Zeit, um eine kurze Show vorzubereiten, entweder alleine, in Paaren oder in Gruppen. Ihr könnt singen, tanzen oder Witze erzählen.
- Wenn jeder bereit ist, zeigt hintereinander eure Aufführungen. Nach jedem Auftritt klatschen die Zuschauerinnen und Zuschauer. Je besser ihnen die Vorstellung gefallen hat, desto lauter klatschen sie.
- Wenn der Applaus vorbei ist, löse IR Bit aus um den Punktestand anzuzeigen (die höchste gemessene Lautstärke).
- Drücke Knopf A, um den Punktestand wieder zurückzusetzen.
- Wer den lautesten Applaus erhalten hat, gewinnt. Viel Spaß!

KNACKE DEN CODE

Wenn ein Programm mehrere Aufgaben erfüllen soll, können wir Event-Trigger (Auslöser) verwenden um von einer Aufgabe zur anderen umzuschalten. Damit das Programm flüssig läuft, überprüfen wir dauerhaft den aktuellen Modus und führen dann den entsprechenden Code aus.

Setze beim Start den Modus auf 0 (Standby-Modus).

Überprüfe, was der aktuelle Modus ist, und führe den Code aus, der zu diesem Modus gehört.

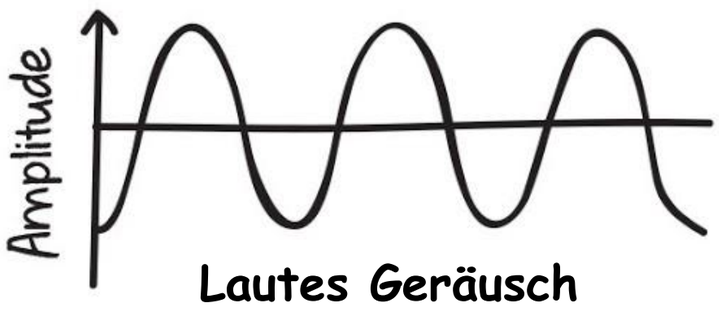


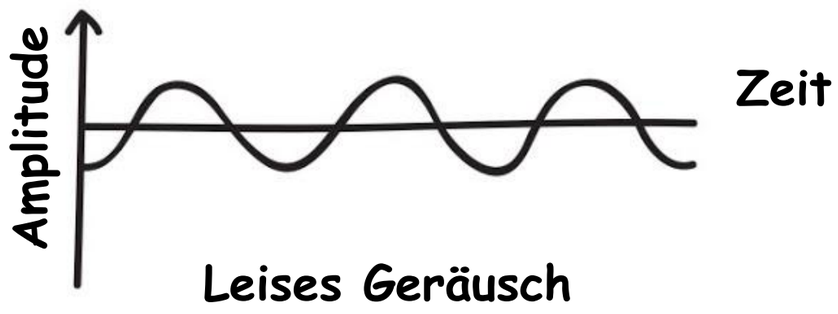
Diese Blöcke sind Event-Trigger, die den Modus während der Laufzeit umschalten.

Du kannst deinem Programm zusätzliche Modi geben, indem du Blöcke als Event-Trigger hinzufügst, z.B. ***[wenn geschüttelt]*** *oder* ***[Lautstärke > _]****, und das (+)-Symbol im Block* ***[wenn _ dann _ ansonsten]*** *drückst um mehr Bedingungen hinzuzufügen.*

GUT ZU WISSEN!

Geräusche entstehen, wenn Dinge vibrieren, wie zum Beispiel eine Trommel, wenn sie geschlagen wird. Die Vibration bringt die Luftmoleküle rundherum (Medium) zum Mitschwingen. So entstehen Schallwellen.
Ein **Schallsensor** ist ein Bauelement, dass die Stärke der Schallwellen (die Lautstärke) misst und sie in ein elektrisches Signal umwandelt.





Ein Schallsensor funktioniert ähnlich wie unsere Ohren, die die Luftschwingungen in elektrochemische Signale umwandeln, die dein Gehirn verarbeiten kann.

Anwendungsbeispiele:

- Alarmanlage gegen Einbrecher
- Intelligente Lichtsteuerung
- Babyfon

Denkst du, dass ein Schallsensor Geräusche im Weltraum erkennen kann? Warum oder warum nicht?

HERAUSFORDERUNG

Programmiere EDU:BIT als Lautstärkemessgerät für dein Klassenzimmer. Zeige die Lautstärke mit den LEDs auf Ampel Bit an.

Geräuschpegel	**Sound Bit Lautstärke**	**Ampel Bit**
Zu laut; bitte seid leiser.	von ( ) bis 1023	Rote LED
Ein wenig unruhig, achtet auf eure Lautstärke.	von ( ) bis ( )	Gelbe LED
Angenehme Lautstärke. Spitze!	von 0 bis ( )	Grüne LED

Hier sind ein paar Tipps für dich…
#1 Du musst vorher die Schwellenwerte für jeden Geräuschpegel messen.
#2 Kombiniere Lautstärkemessungen an mehreren Zeitpunkten zu einem Durchschnittswert.

Zu einfach? Ändere deinen Code so, dass die Schwellenwerte sich mit der Potentiometer-Einstellung verändern.

KAPITEL 8

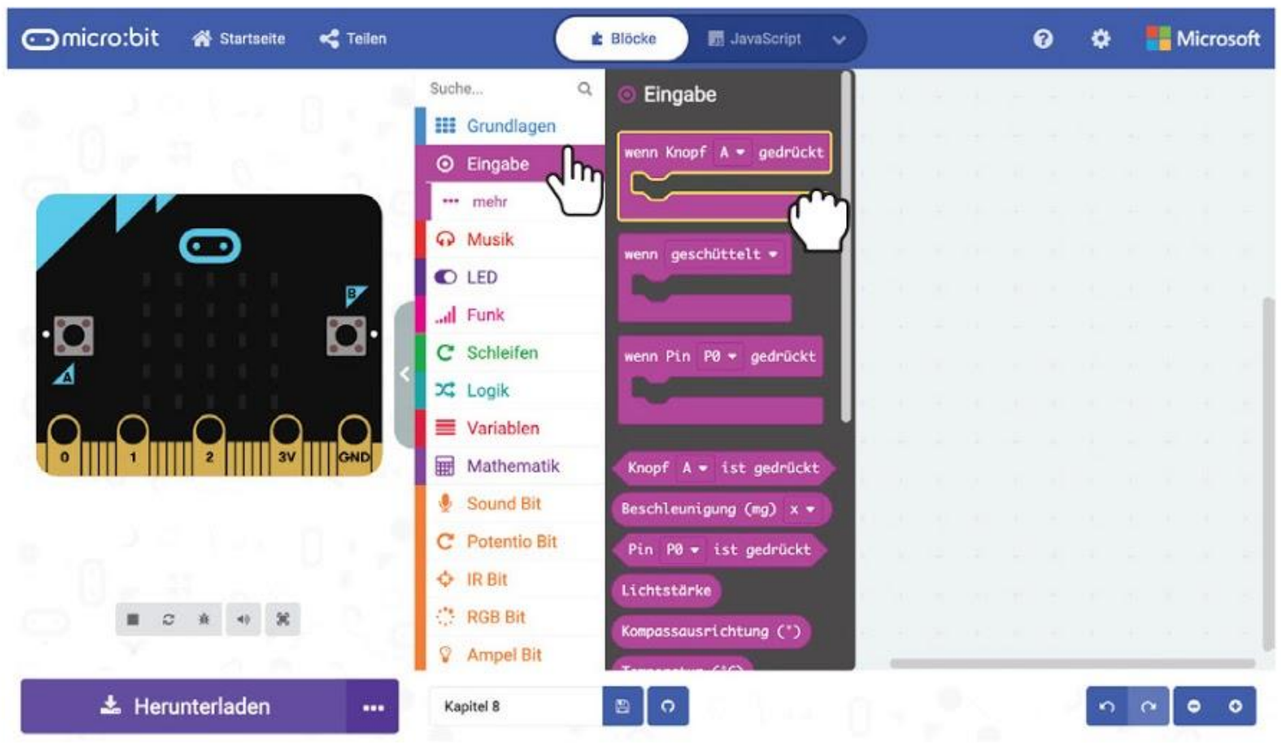
Drehen wir eine Runde!

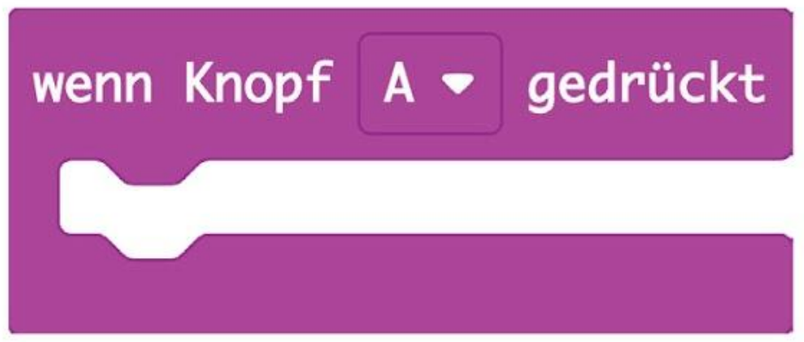
DC-Motor

link.cytron.io/edubit-chapter-8

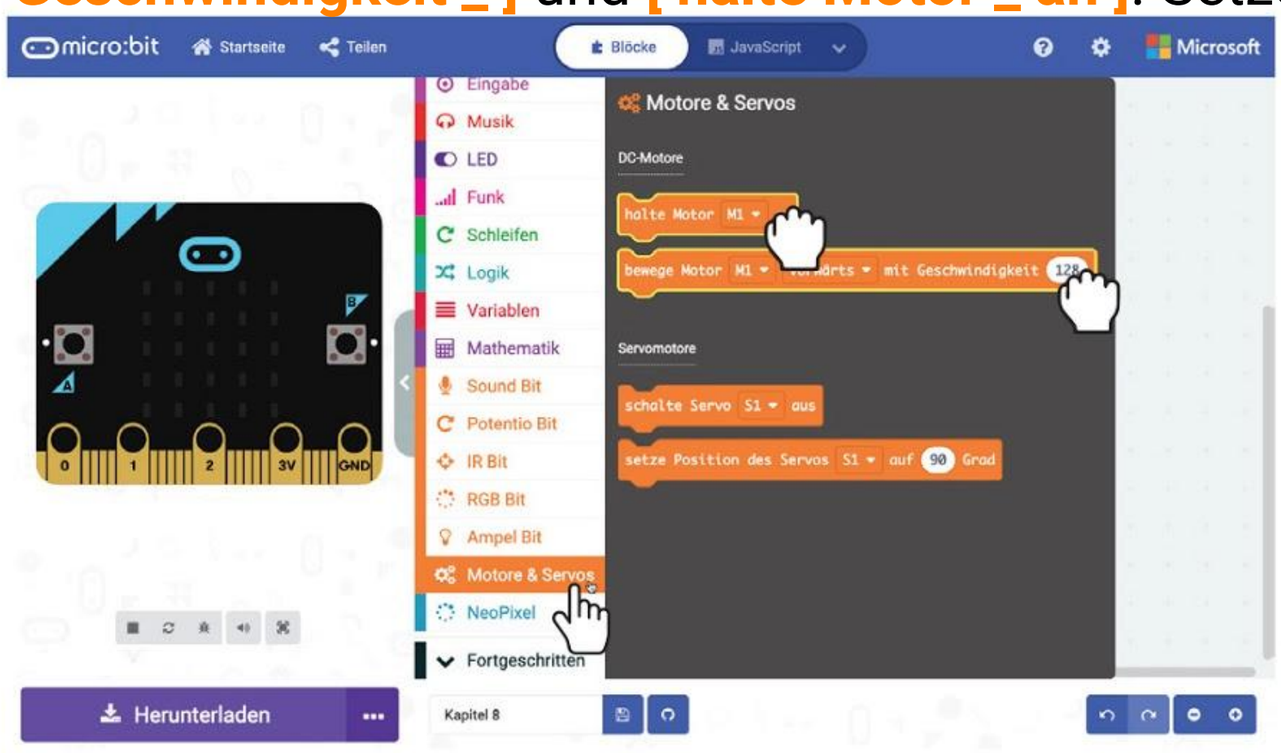
LASS UND PROGRAMMIEREN!

Schritt 1 Erstelle ein neues Projekt im MakeCode Editor und füge die EDU:BIT-Erweiterung hinzu (siehe Seite 40). Unter **[Eingabe]** klicke auf den Block **[wenn Knopf _ gedrückt]**.





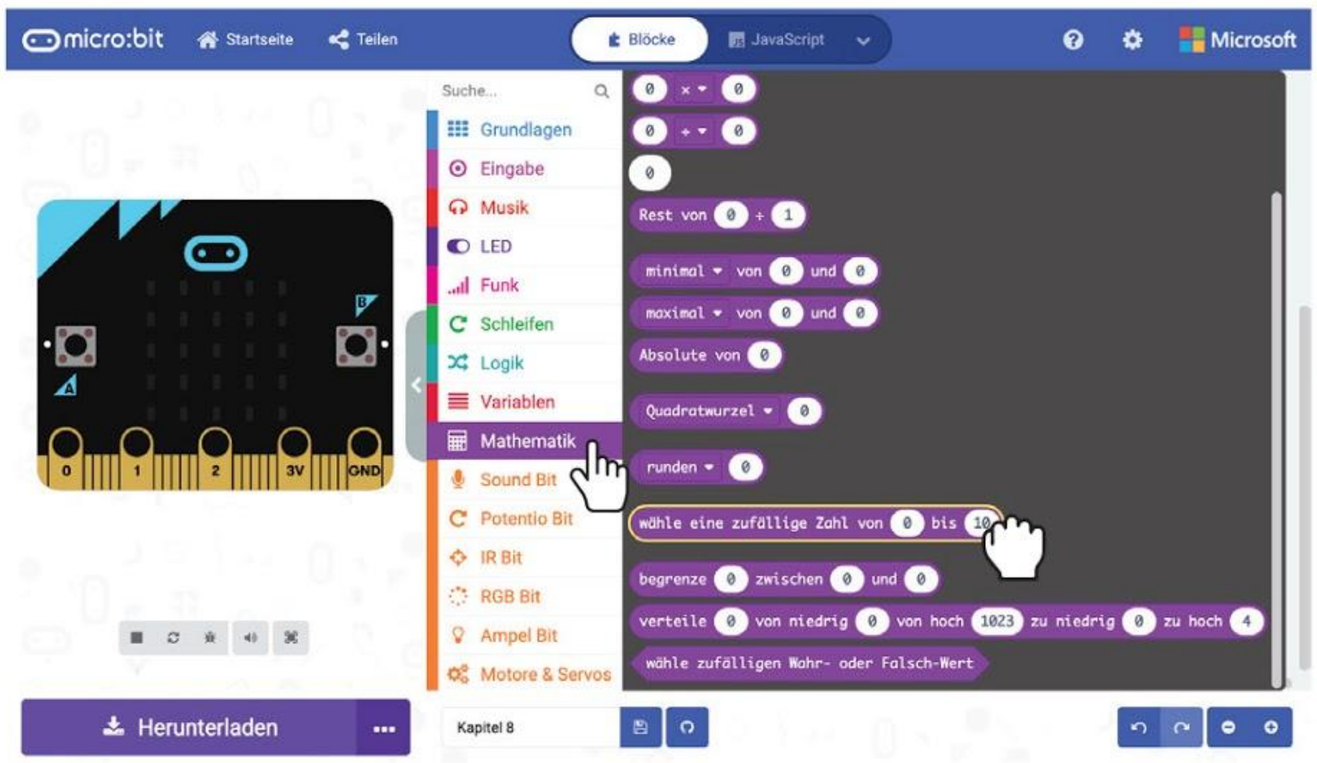
Schritt 2 Klicke unter **[Motore und Servos]** auf die Blöcke **[Bewege Motor _ _ mit Geschwindigkeit _]** und **[halte Motor _ an]**. Setze die Geschwindigkeit auf 80.



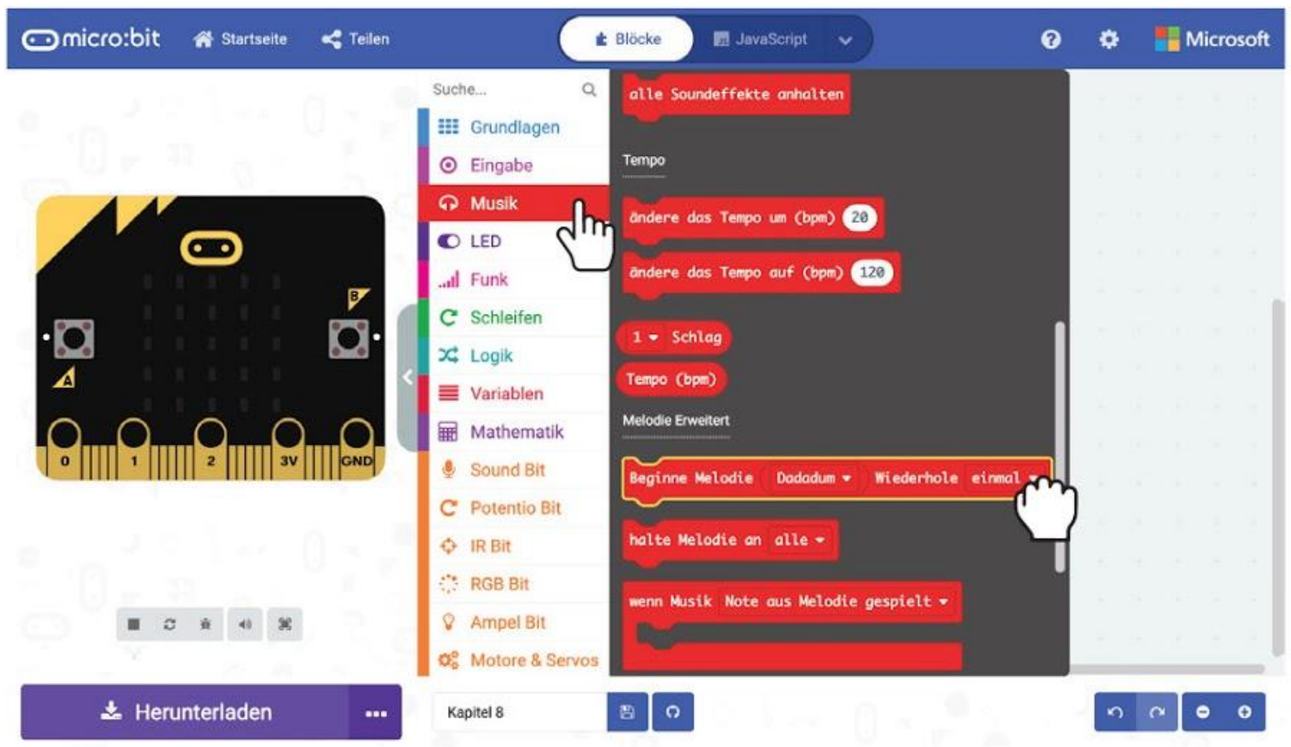
Schritt 3 Klicke in der Gruppe **[Grundlagen]** auf den Block **[pausiere (ms) _]**. Lege ihn zwischen **[Bewege Motor _ _ mit Geschwindigkeit _]** und **[halte Motor _ an]**.

KAPITEL 8 : Drehen wir eine Runde!

Schritt 4 Klicke auf **[Mathematik]** und **[wähle eine zufällige Zahl von _ bis _]**. Ziehe den Block in **[pausiere (ms) _]** und ändere die Werte auf **200** und **1000.**



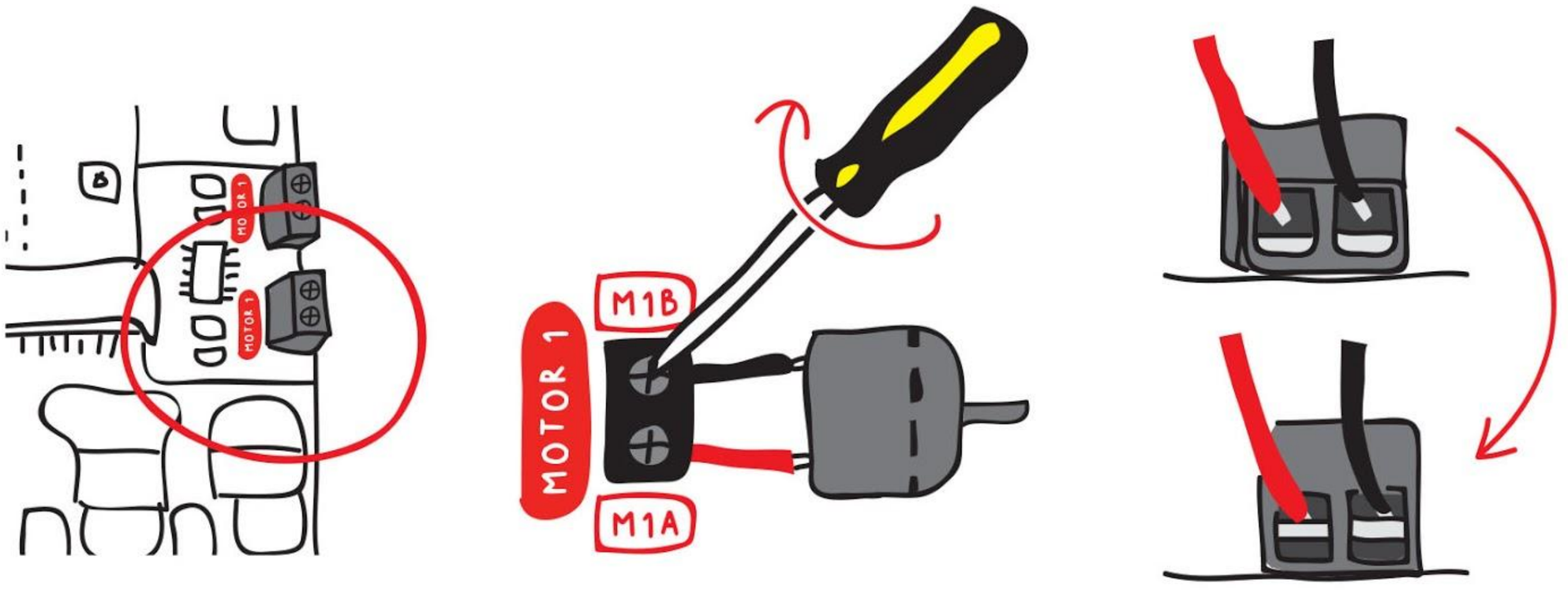
Schritt 5 Klicke unter **[Musik]** auf den Block **[Beginne Melodie _ Wiederhole _]**. Ändere die Melodie in “**Ping ping**” (oder eine andere Melodie, die dir gefällt).

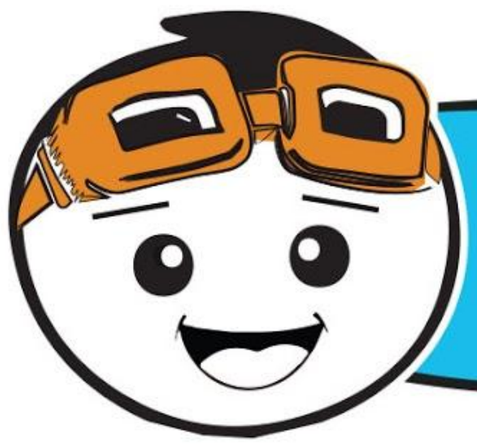


Schritt 6 Lade den Code auf deinen EDU:BIT.

Wir können diesen Code für jedes Spiel verwenden, das einen zufälligen Drehzeiger benötigt. Der Motor dreht sich, und stoppt nach einem zufälligen Zeitraum.

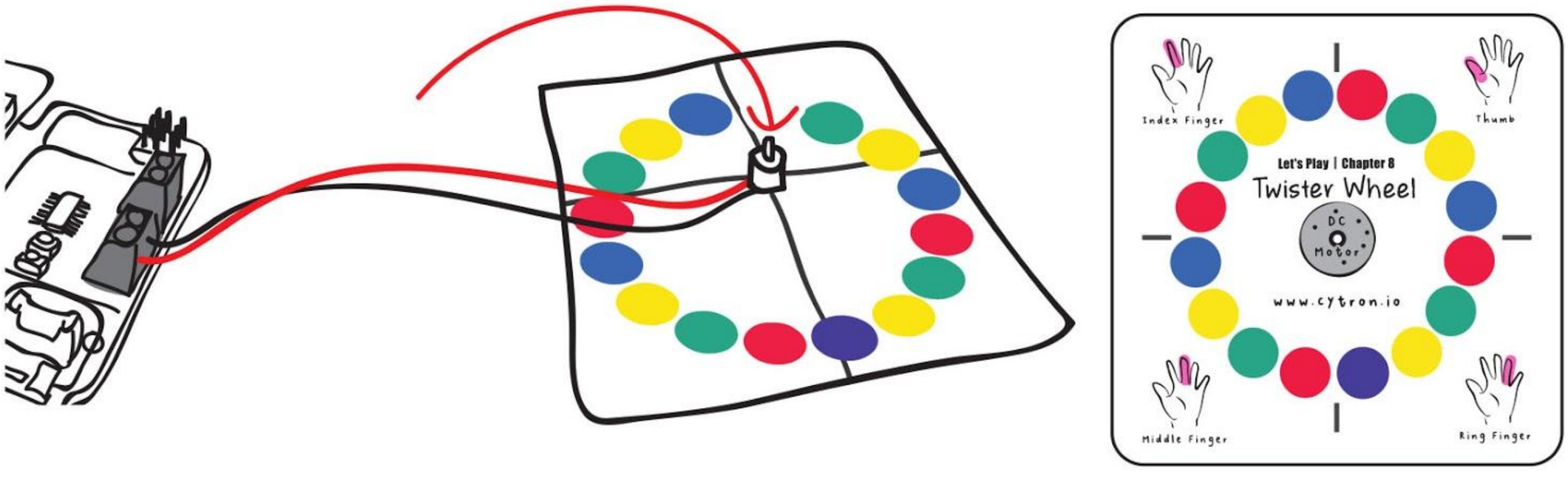
Schritt 7 Verbinde den DC-Motor mit mit dem Anschluss ‘MOTOR 1’ - (1) Stecke den freigelegten Draht hinein und (2) ziehe die Schraube mit dem beigelegten Schraubenzieher an, damit der Draht nicht mehr herausrutscht.



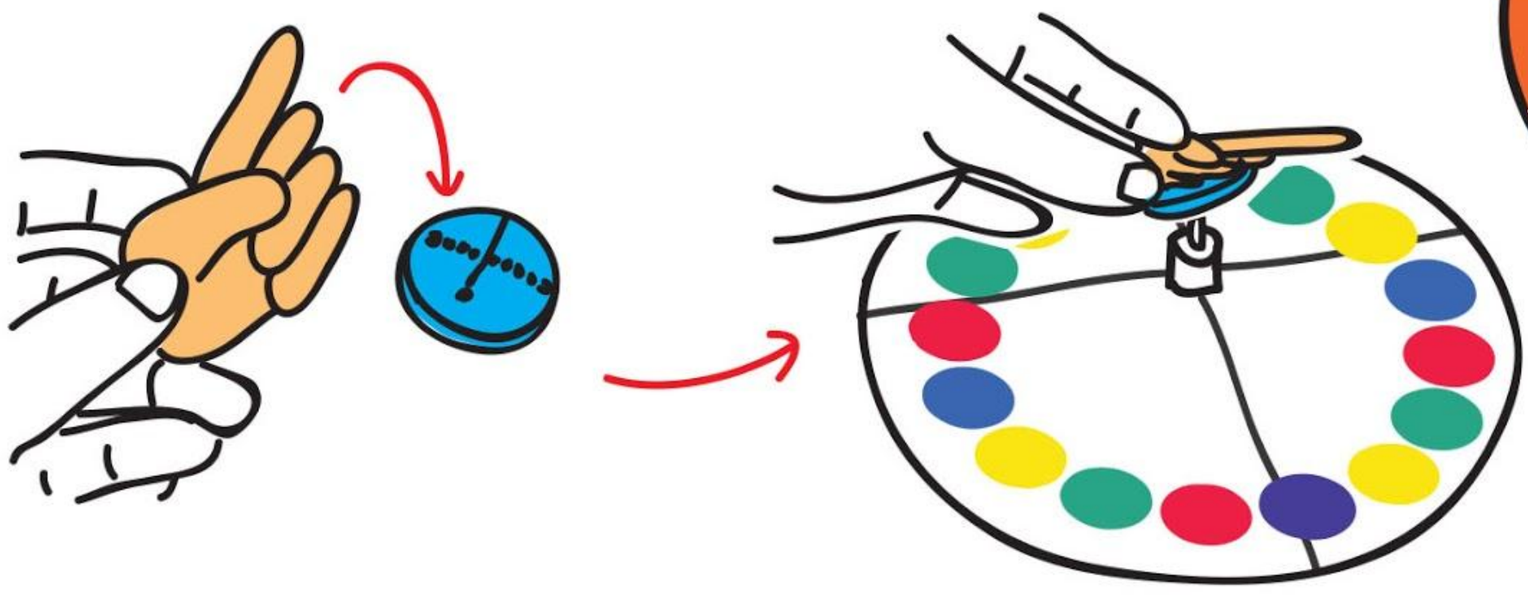


Drücke den gelben Knopf (A) zum Testen. Wenn der Motor sich nicht dreht, überprüfe ob die Verbindung am Anschluss fest sitzt und EDU:BIT eingeschaltet ist.

Schritt 8 Benutze doppelseitiges Klebeband oder eine Heißklebepistole, um den Motor in der Mitte der Twister-Drehscheibe anzukleben.



Schritt 9 Drücke den Zeiger aus dem Karton und klebe ihn an die Plastikscheibe. Befestige die Scheibe dann an der Motorwelle.



Die Twister-Drehscheibe, den Zeiger und ein Spielfeld findest du in der EDU:BIT-Box.

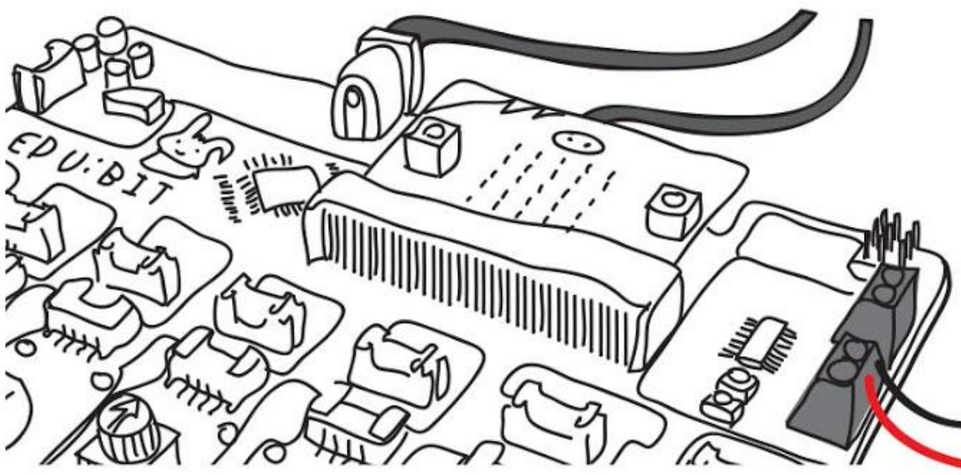
Spielen wir!

Drehen wir eine Runde!

Vorbereitung:

- Lege das Spielfeld auf den Tisch. Zwei Spielerinnen oder Spieler sitzen einander gegenüber. Wenn mehr Leute mitmachen, nehmen sie die anderen Plätze ein (max. 4 Personen).

- Die Schiedsrichterinnen und Schiedsrichter sitzen in der Nähe mit der Twister-Drehscheibe.



SPIELANLEITUNG:

In diesem Spiel müssen die Spielerinnen und Spieler abwechselnd ihre Finger auf die bunten Kreise auf dem Spielfeld legen.

Die Schiedsrichterinnen und Schiedsrichter drücken den gelben Knopf (A) um den Zeiger zu drehen und sagen dann den Finger und die Farbe an, zum Beispiel: “Zeigefinger; rot”.

Wenn du am Zug bist, musst du den richtigen Finger auf die richtige Farbe legen. Wenn der Finger schon auf dieser Farbe liegt, musst du ihn auf ein anderes Feld mit dieser Farbe bewegen.

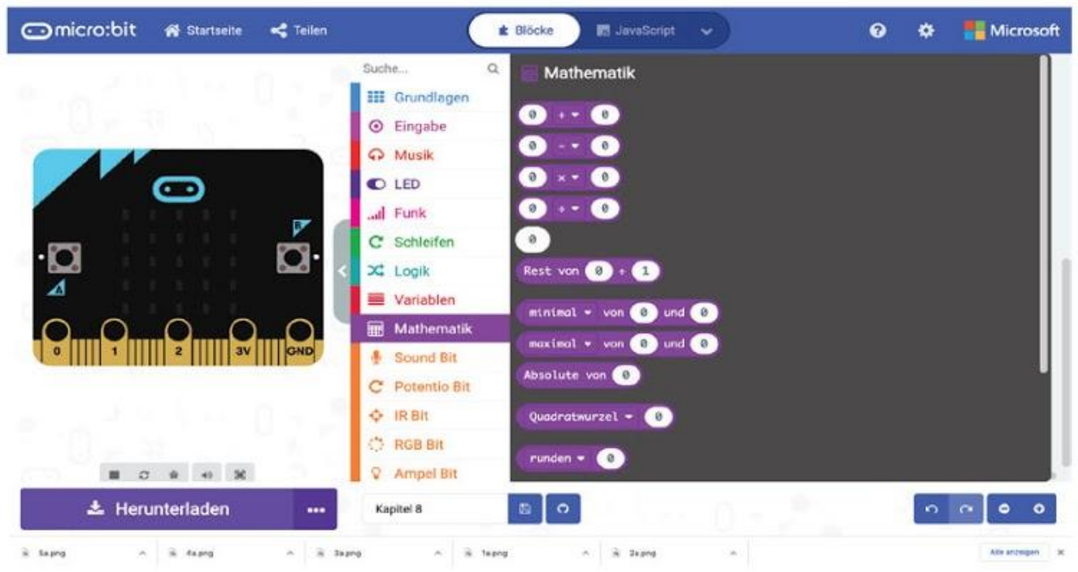
Wenn du das nicht schaffst, scheidest du aus.

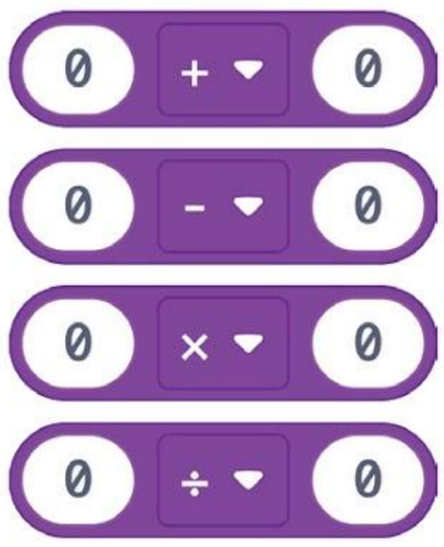
Wer bis zum Schluss übrig bleibt, GEWINNT!

ENTDECKE NOCH MEHR

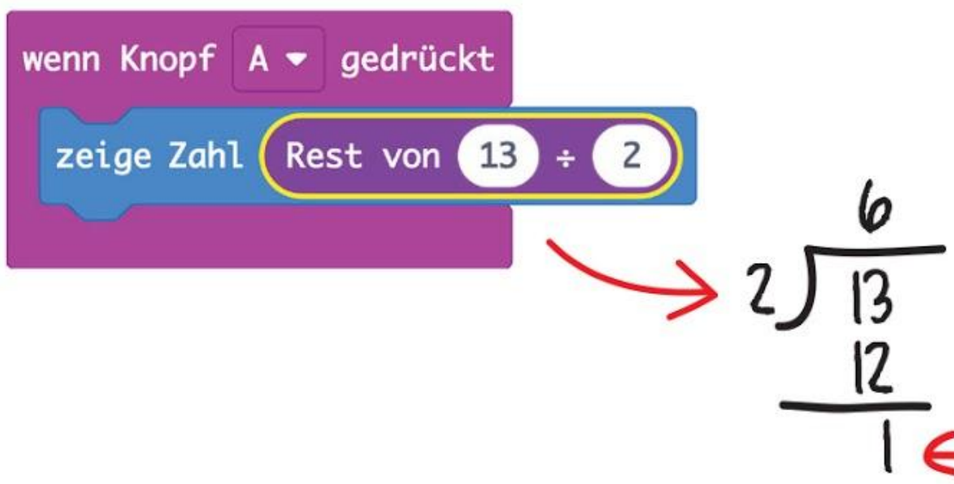
Du kannst Blöcke aus **[Mathematik]** für Rechenoperationen mit deinen Variablen verwenden.

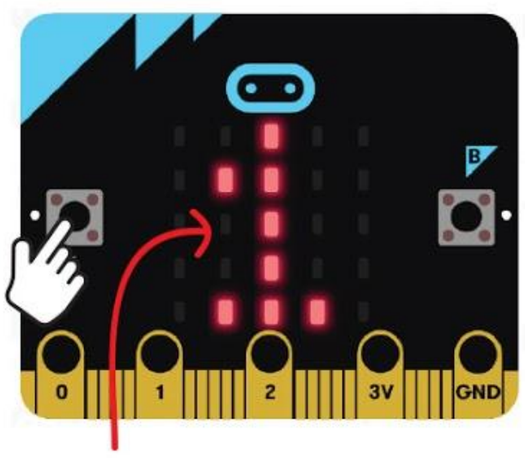
#1 Benutze diese Blöcke zum Addieren, Subtrahieren, Multiplizieren oder Dividieren.



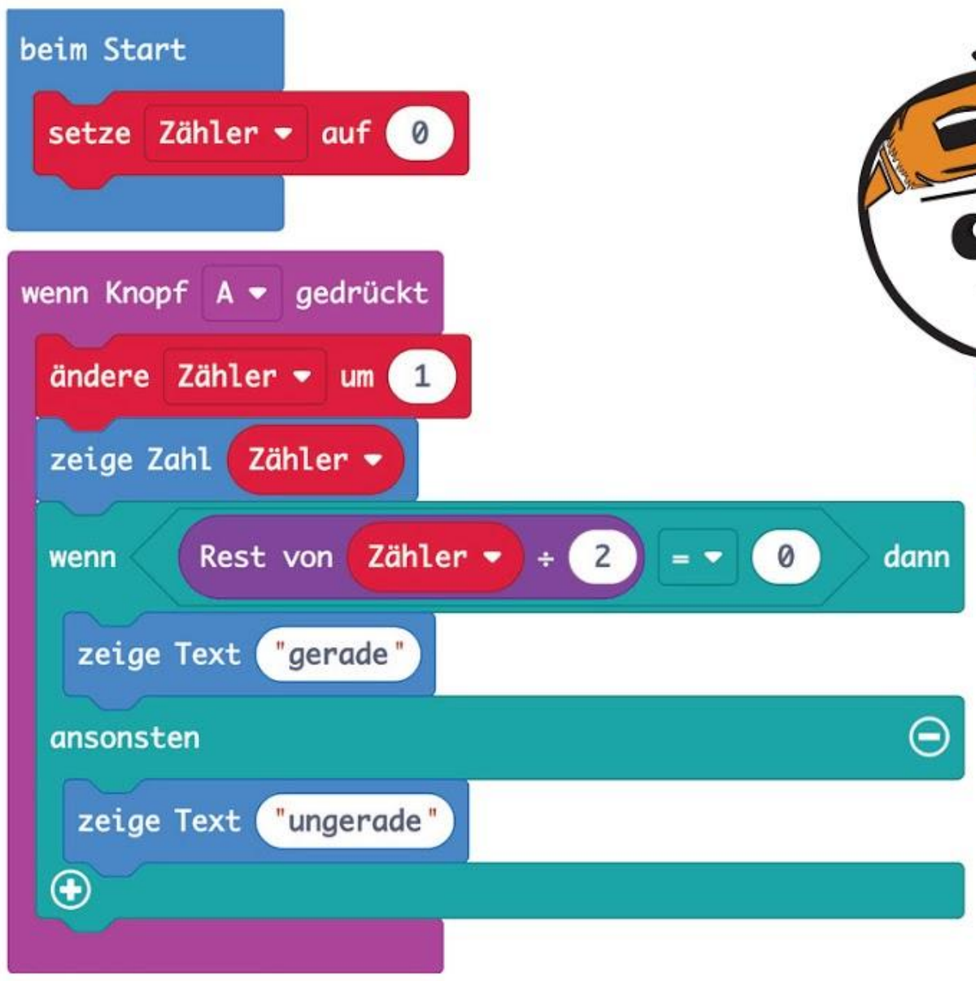


#2 Benutze **[Rest von _ ÷ _]** um zu berechnen wieviel übrig bleibt, wenn eine Zahl nicht genau durch eine andere Zahl geteilt werden kann.





#3 Benutze den Block **[Rest von _ ÷ _]** um zu berechnen, ob eine Zahl gerade oder ungerade ist. Dividiere einfach durch 2. Wenn der Rest “1” ist, ist die Zahl “ungerade”; wenn der Rest “0” ist, ist sie “gerade”. Probiere es aus!

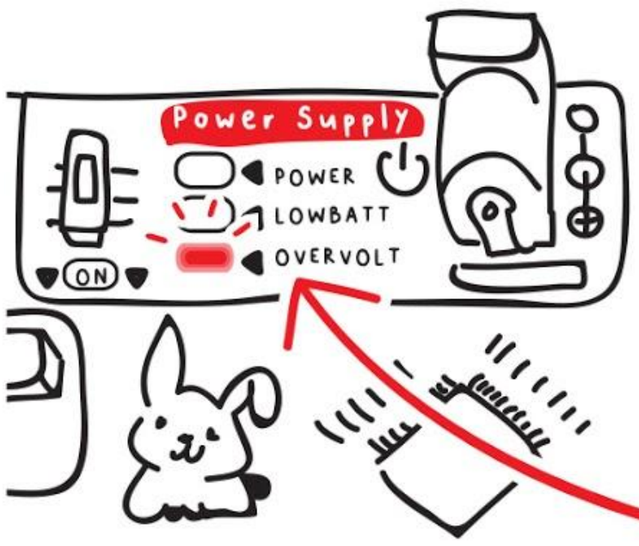


Hast du gesehen, dass EDU:BIT “ungerade” anzeigt, wenn du Knopf A das erste Mal drückst (1 = ungerade Zahl) und “gerade” beim nächsten Mal (2 = gerade Zahl)? Was wird wohl beim 99. Mal angezeigt werden?

GUT ZU WISSEN!

Ein Gleichstrommotor oder **DC-Motor (direct current motor)**, ist ein elektrisches Gerät, das elektrische in mechanische Energie umwandelt.

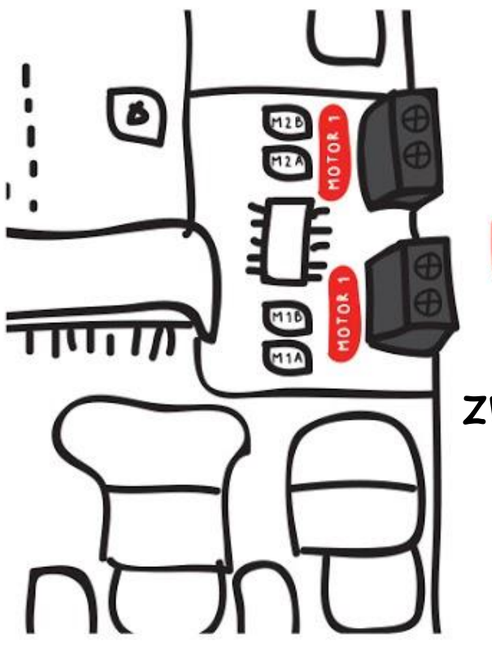
Dafür braucht der Motor eine Stromspannung. Wir können die Drehgeschwindigkeit steuern, indem wir die Spannung verändern. Je höher die Eingangsspannung, desto schneller dreht sich der Motor. Die empfohlene Spannung für den Motor in der EDU:BIT-Box ist 3,6 V bis 6 V.



Überspannungs-Anzeige

ACHTUNG!
Eine höhere Spannung (als die empfohlene Spannung) verkürzt auf lange Sicht die Lebensdauer des Motors.

Du kannst die Drehrichtung und die Geschwindigkeit des Motors ganz einfach mit den folgenden Code-Blöcken steuern.



Die EDU:BIT-Platine hat zwei DC-Motor-Anschlüsse. Wähle den richtigen Anschluss aus.

Drehrichtung

Dieser Wert geht von 0 bis 255. Je höher der Wert, desto schneller dreht sich der Motor.

Wusstest du, dass EDU:BIT einen Motortester eingebaut hat? Drücke die weißen Knöpfe (M1A, M1B, M2A und M2B) um zu überprüfen, ob die Kabelverbindung und der Motor funktionieren.

HERAUSFORDERUNG

Schreibe ein Programm für EDU:BIT, das bei Geräuschen einen Ventilator einschaltet. Die Geschwindigkeit des Ventilators soll über Potentio Bit gesteuert werden.

beim Start	Zeige ein Herz-Symbol (oder ein anderes Symbol) an. Setze die Variable Modus auf 0
wenn Lautstärke > _ (Schwellenwert)	Ändere Modus um 1
dauerhaft	Setze die Variable ‘Geschwindigkeit’ auf den Potentiometer-Wert, aber neu verteilt von niedrig 0, hoch 1023 auf niedrig 0, hoch 255. Überprüfe Modus ● WENN Modus eine gerade Zahl ist, halte M1 an. ● SONST WENN Modus eine ungerade Zahl ist, bewege Motor M1 mit der Geschwindigkeit in der Variable ‘Geschwindigkeit’.

Hier sind ein paar Tipps für dich…
#1: Finde die Lautstärke heraus (den Schwellenwert) bei der der Motor gestartet und gestoppt werden soll.
#2: Du brauchst zwei Variablen: Modus und Geschwindigkeit.
#3: Befestige das Ventilatorblatt an der Motorwelle und führe das Programm aus. Wenn du vom sich drehenden Ventilator keinen Wind spürst, musst du die Drehrichtung umkehren.

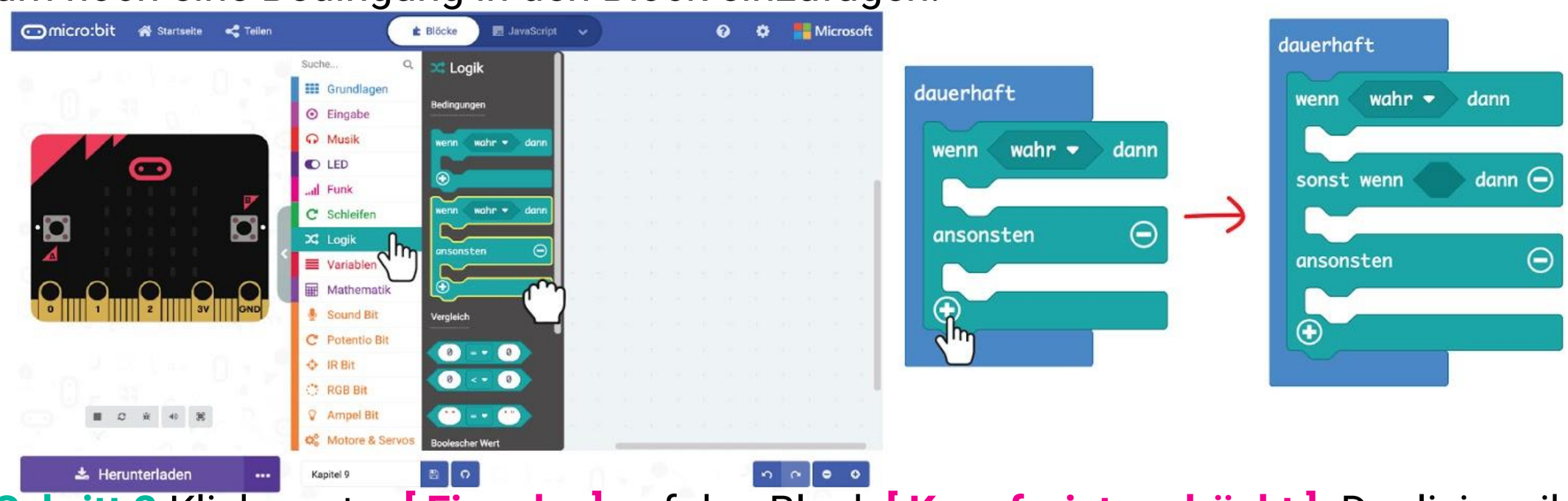
KAPITEL 9

Elfmeterschießen... Tor!!!

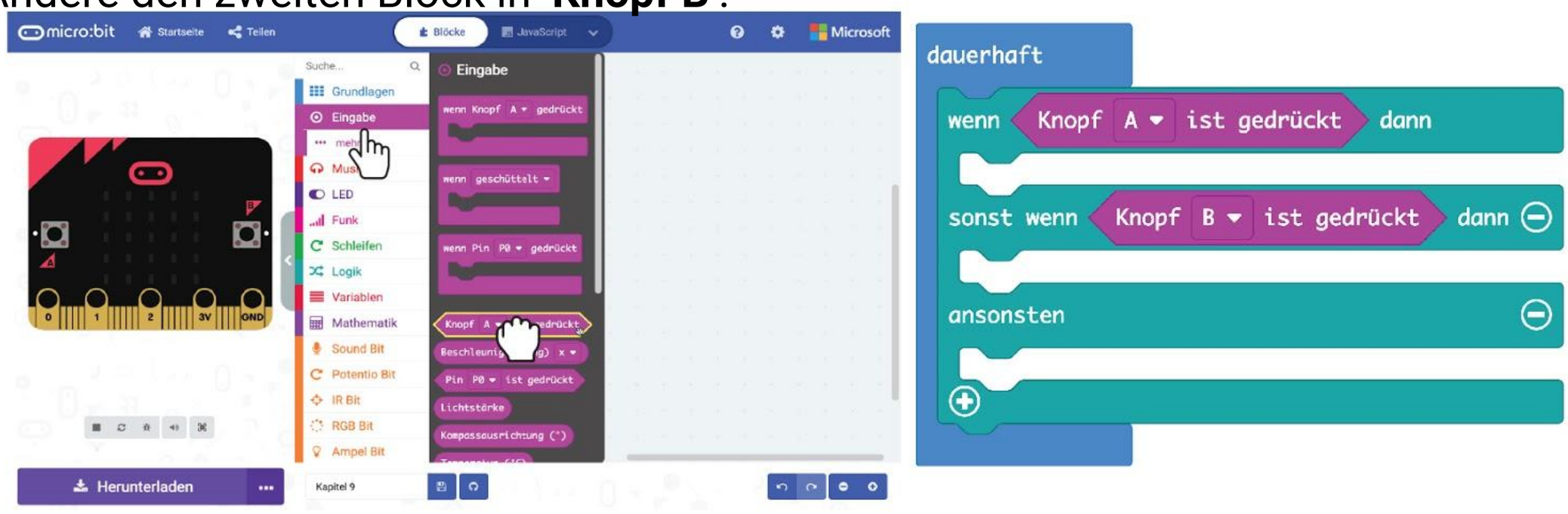
Servomotor

LASS UNS PROGRAMMIEREN!

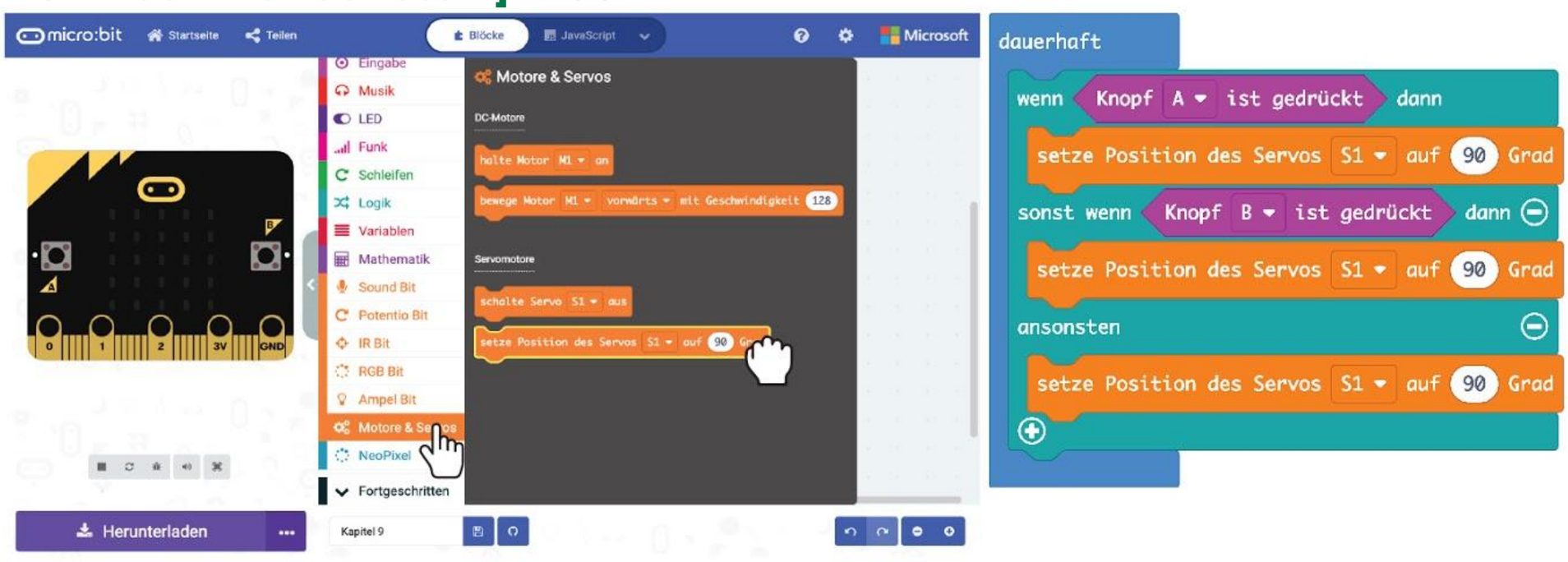
Schritt 1 Erstelle ein neues Projekt im MakeCode Editor und füge die EDU:BIT-Erweiterung hinzu (siehe Seite 40). Klicke unter **[Logik]** auf den Block **[wenn-dann-ansonsten]**. Lege den Block in **[dauerhaft]**. Klicke auf das (+)-Symbol um noch eine Bedingung in den Block einzufügen.



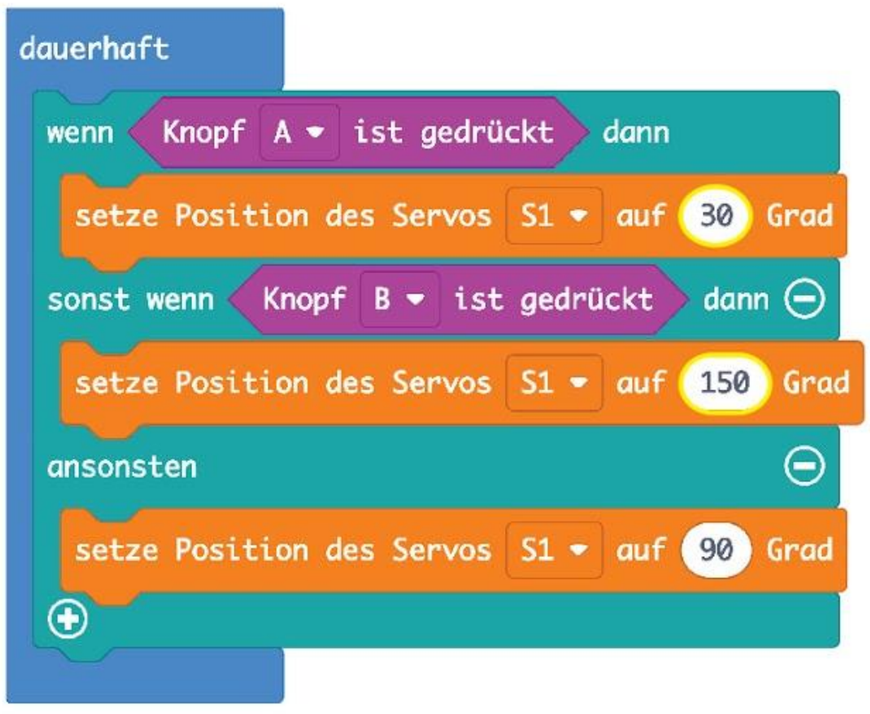
Schritt 2 Klicke unter **[Eingabe]** auf den Block **[Knopf _ ist gedrückt]**. Dupliziere ihn und lege die Blöcke in die Bedingungen in den **[wenn-dann-ansonsten]**-Blöcken. Ändere den zweiten Block in '**Knopf B**'.



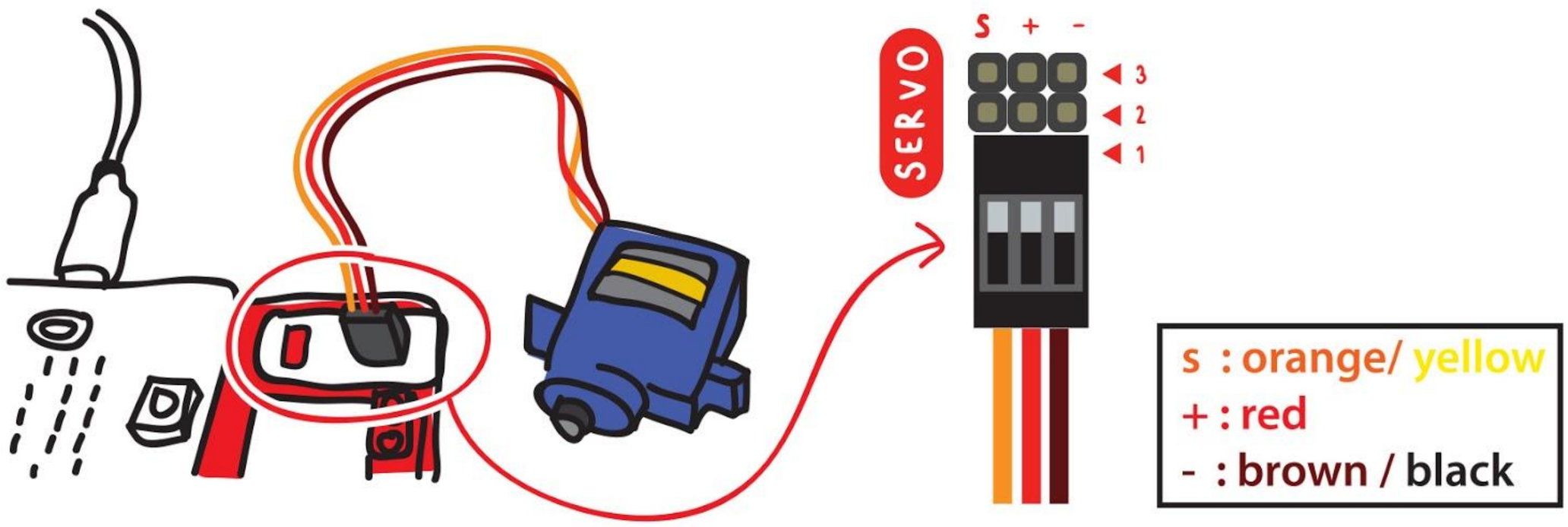
Schritt 3 Wähle unter **[Motore & Servos]** den Block **[setze Position des Servos _ auf _ Grad]**. Dupliziere den Block und lege je einen in den Code-Bereich im **[wenn-dann-ansonsten]**-Block.



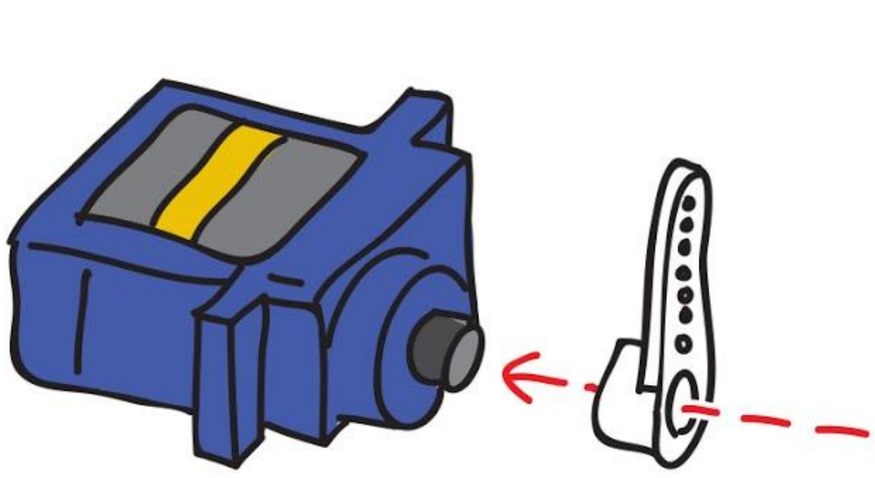
Schritt 4 Ändere die Werte im ersten und zweiten Block in **30** und **150**. Übertrage den Code auf deinen EDU:BIT.

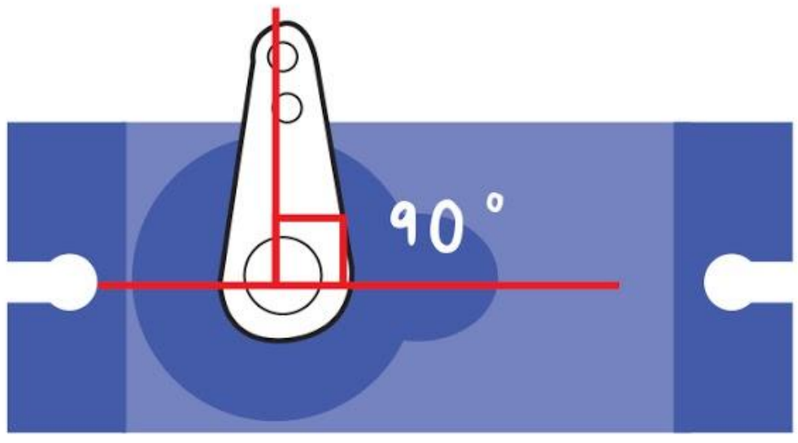


Schritt 5 Stecke das Kabel an den Servomotor-Anschluss 1 am EDU:BIT, so wie unten gezeigt.

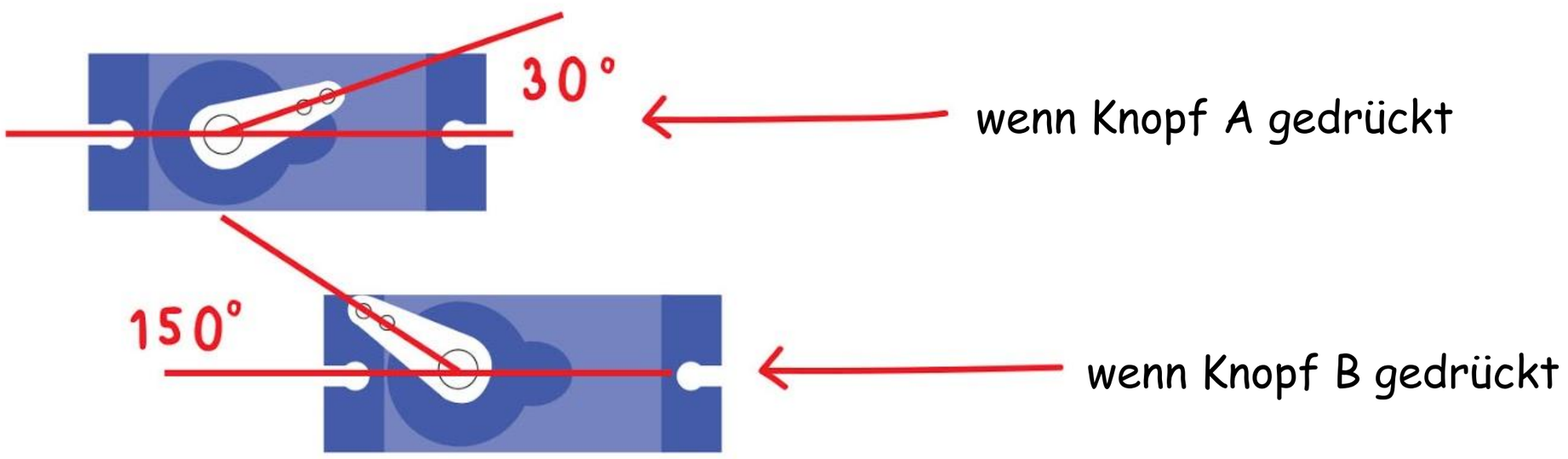


Schritt 6 Schalte EDU:BIT ein und bringe den Servohebel an der Motorwelle des Servos im Winkel von 90 Grad an, wie unten gezeigt.

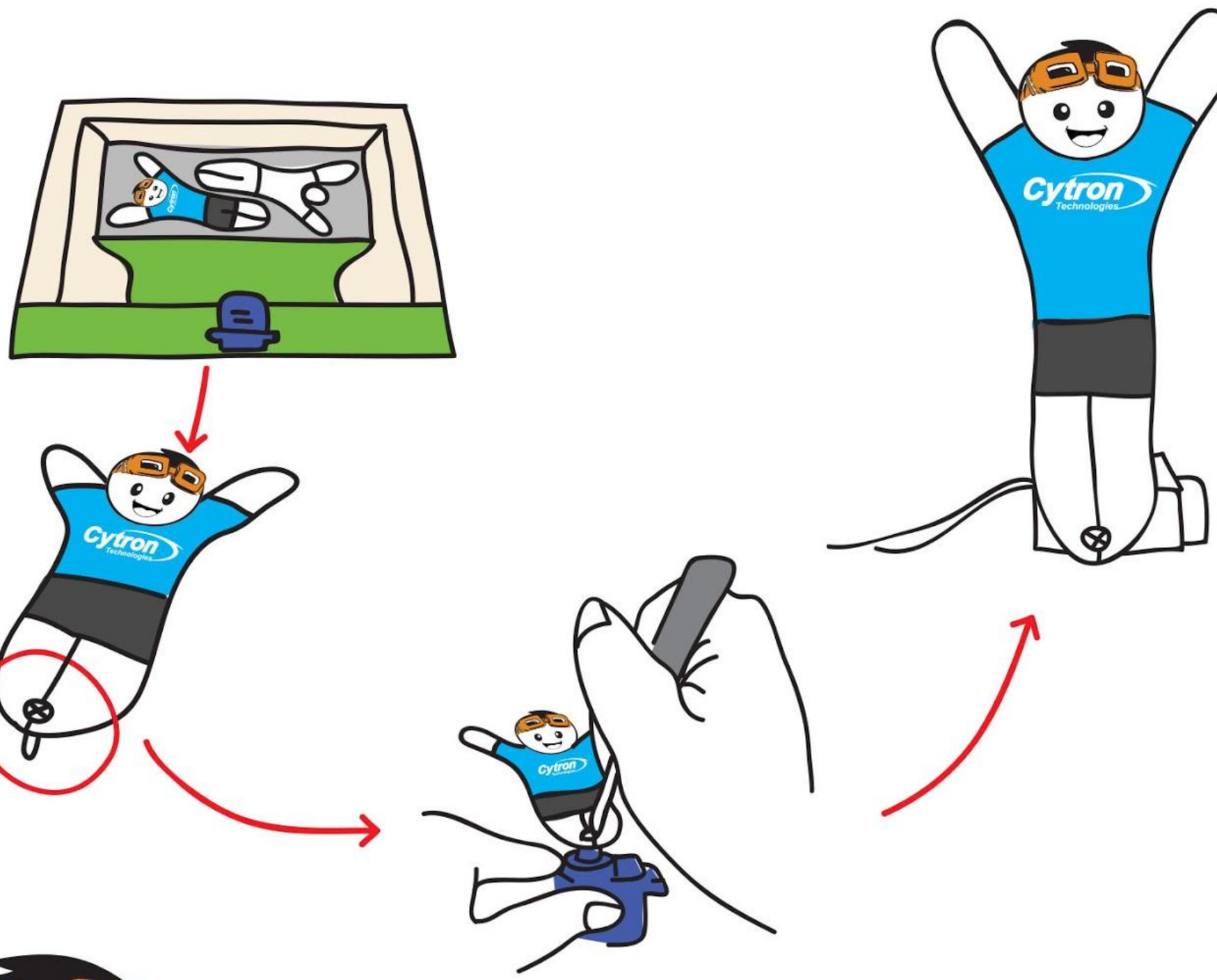




Schritt 7 Drücke zum Testen auf Knopf A und Knopf B.

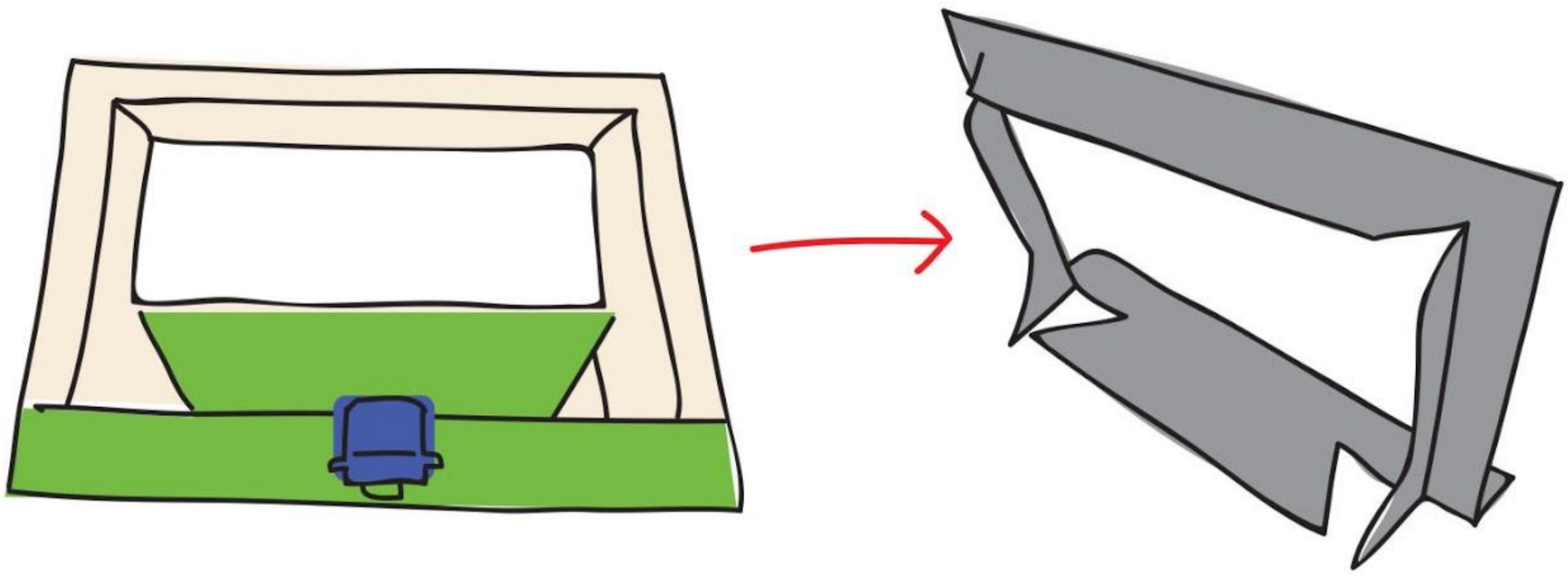


Schritt 8 Drücke die Torwache aus dem Karton. Befestige die Torwache mit dem beigelegten Schraubenzieher und der Schraube am Servohebel. Fixiere sie mit doppelseitigem Klebeband oder einer Heißklebepistole.

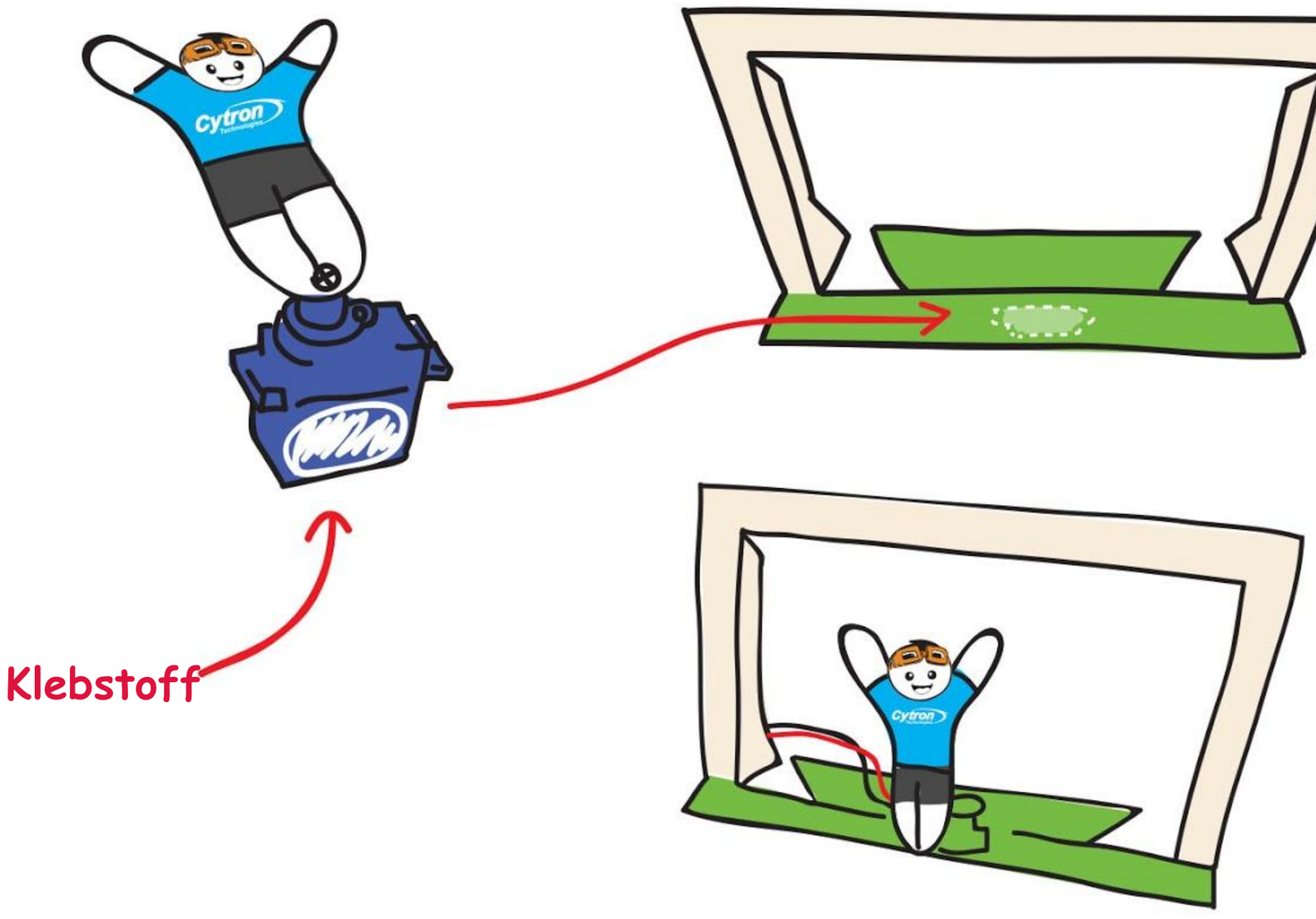


Wenn du willst, kannst du die andere Figur benutzen und ihr ein eigenes Trikot malen.

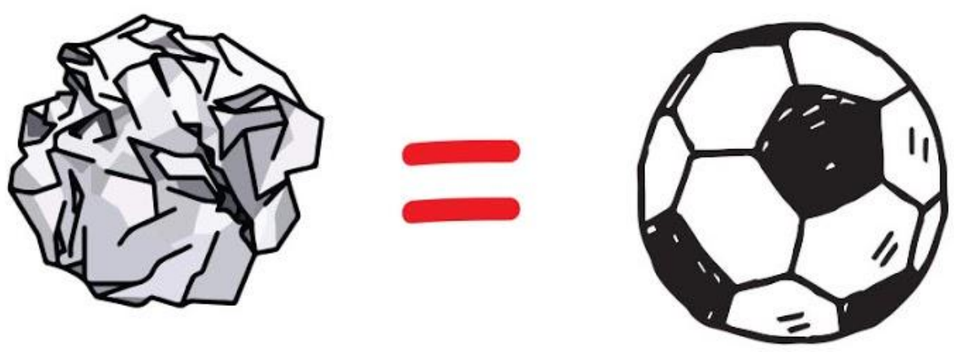
Schritt 9 Drücke das Tor aus dem Karton und stelle es wie unten gezeigt auf.



Schritt 10 Klebe den Servomotor mit dem doppelseitigem Klebeband oder mithilfe der Heißklebepistole an den markierten Platz im Tor.

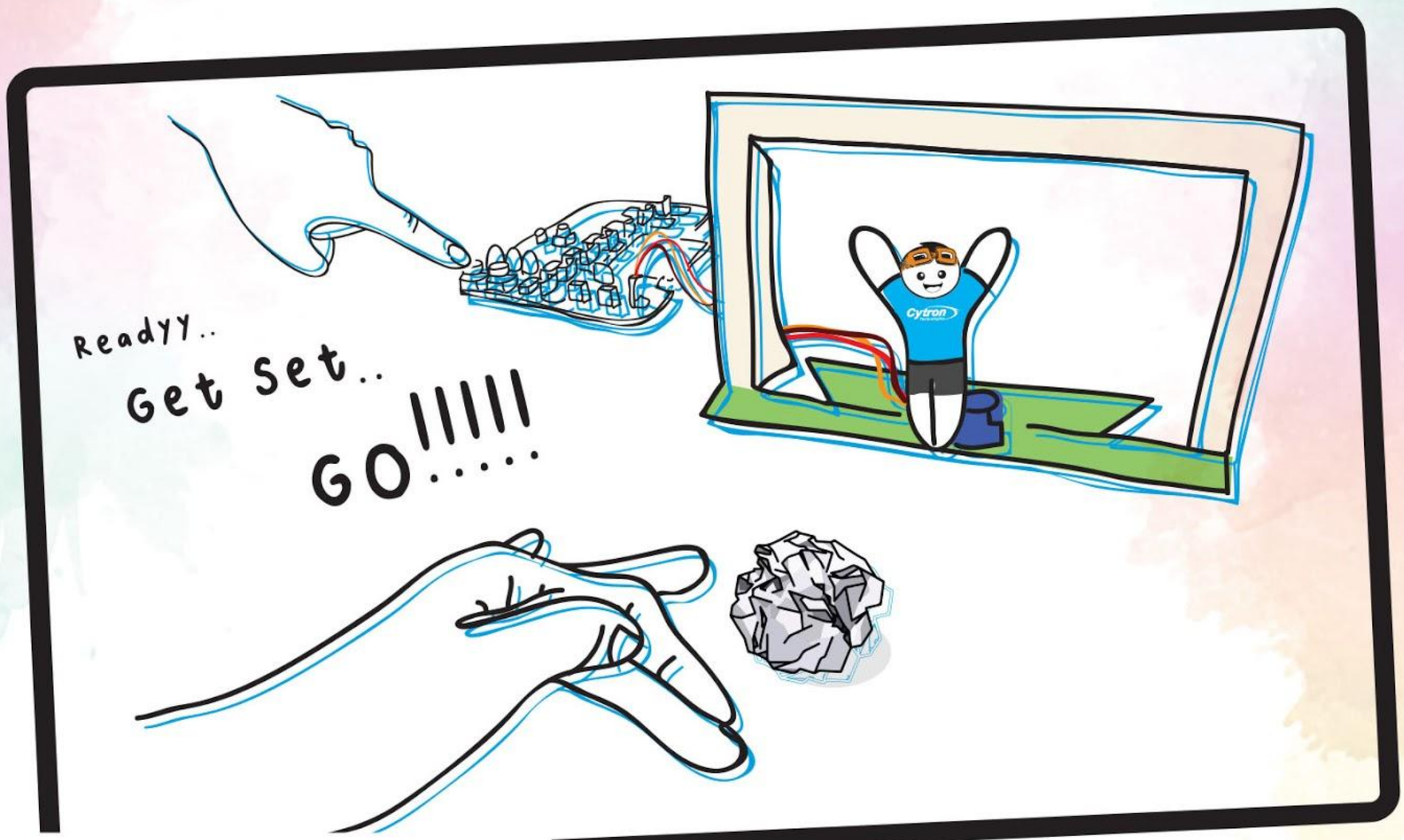


Schritt 11 Zerknülle ein kleines Stück Papier um daraus einen “Fußball” zu formen und schon bist du bereit für das Elfmeterschießen. Alles klar?



Spielen wir!

Elfmeterschießen... Tor!!!



SPIELANLEITUNG:

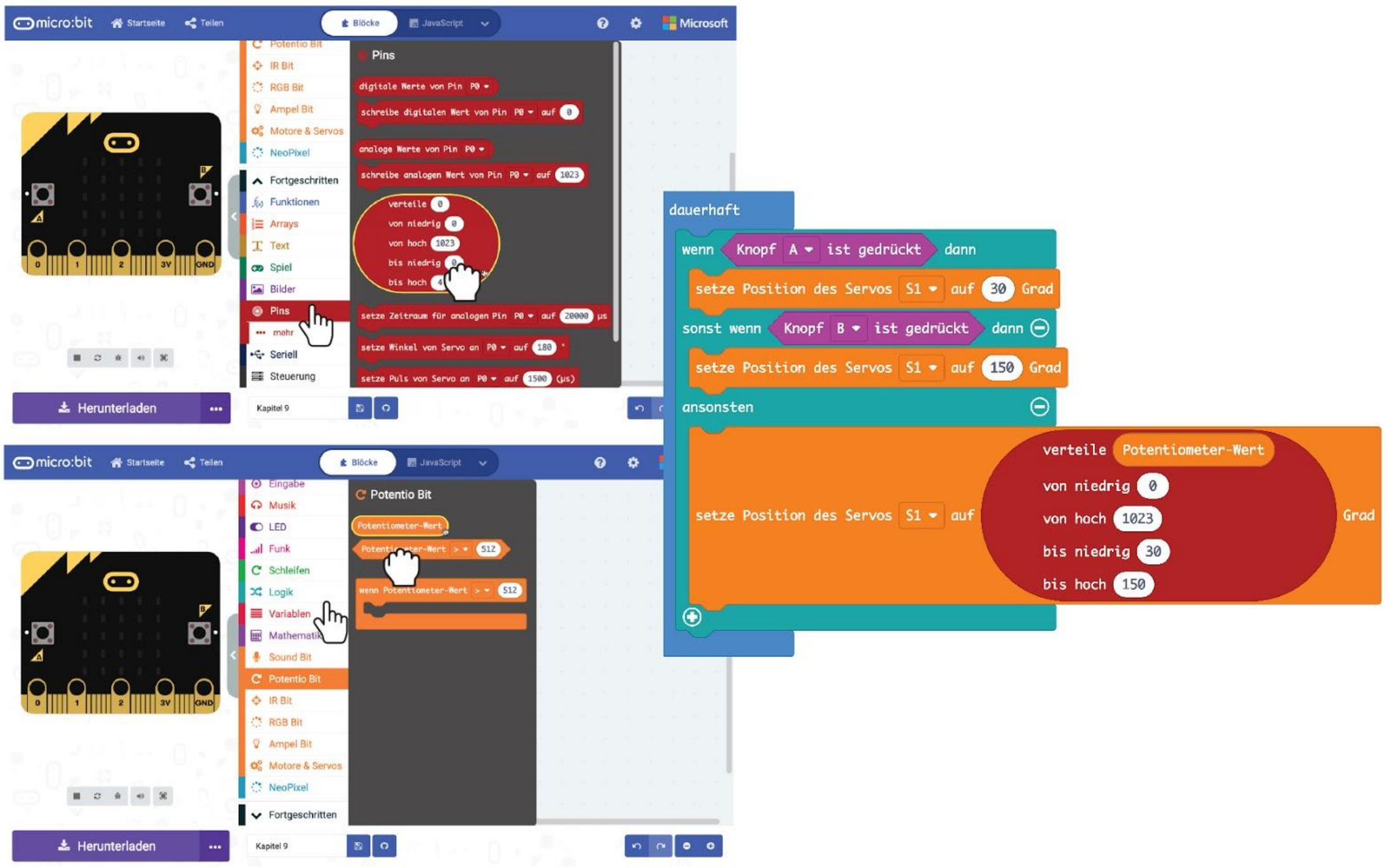
- Stelle das Tor auf und markiere, von wo du schießt (ca. 1 Meter Abstand, weniger für jüngere Spielerinnen und Spieler)
- Wechselt ab, wer schießt und wer Torwache ist.
- Die Kickerin oder der Kicker schießt den Ball auf das Tor.
- Die Torwache versucht den Ball abzuwehren, indem sie die Figur mit dem gelben Knopf (A) nach links und mit dem blauen Knopf (B) nach rechts steuert.
- Pro Runde hat jeder 5 Chancen. Wer am Ende die meisten Tore geschossen hat, gewinnt.

Wusstest du, dass Elfmeterschießen dafür verwendet wird, den Sieg zu entscheiden, wenn ein Fußballspiel nach der Nachspielzeit unentschieden endet? Beim Elfmeterschießen hat jedes Team fünf Versuche. Man schießt von der Strafstoßmarke weg auf das Tor. Abwehren darf nur die gegnerische Torwache. Das Team mit den meisten Toren gewinnt.

Im letzten Programm haben wir die Torwache mit den Knöpfen nach links und rechts gedreht. Wir können den Code verändern und die Position der Torwache mit Potentio Bit steuern.

Schritt 12 Klicke unter **[Fortgeschritten]** auf die Gruppe **[Pins]**. Füge einen **[verteile _ von niedrig _ von hoch _ bis niedrig _ bis hoch _]**-Block zu deinem Code hinzu.

Schritt 13 Klicke in der **[Potentio Bit]**-Gruppe auf den Block **[Potentiometer-Wert]**. Verbinde ihn mit dem Block **[verteile _ von niedrig _ von hoch _ bis niedrig _ bis hoch _]** und ändere die Werte zu **30** und **150**.



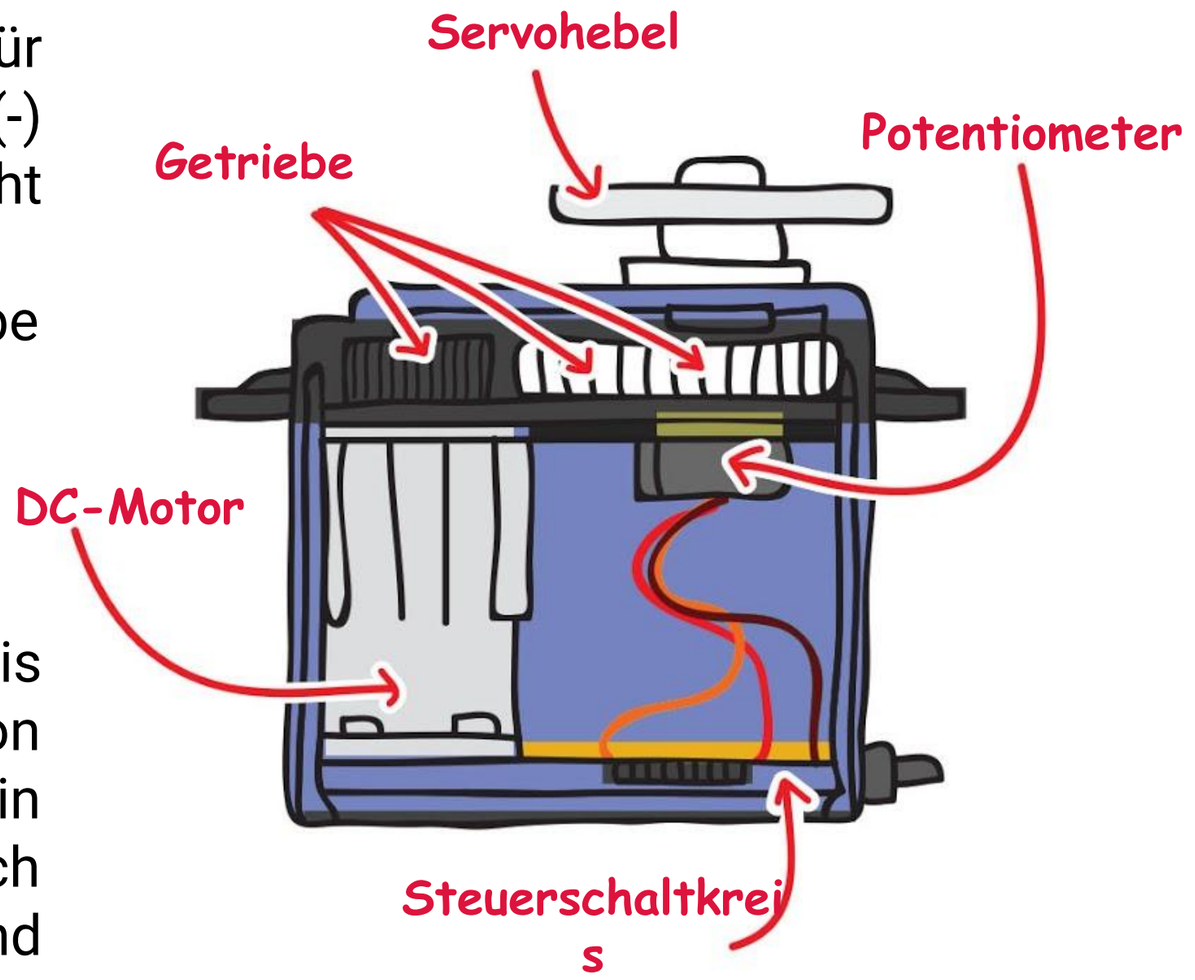
Schritt 14 Flashe den Code auf deinen EDU:BIT. Jetzt kannst du die Torwache mittels Potentiometer steuern. Viel Spaß!

Wenn du alleine Torschüsse üben willst, kannst du den Code zu einem "Übungsmodus" umschreiben, indem du die Torwache dauerhaft von links nach rechts pendeln lässt. Probiere es aus!

GUT ZU WISSEN!

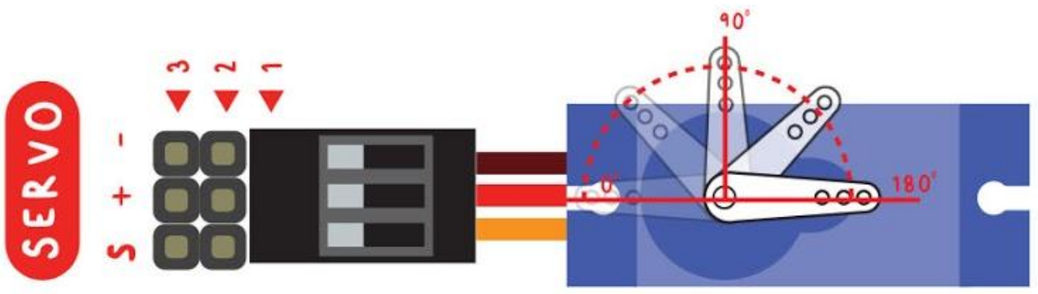
Der **Servomotor** in der EDU:BIT-Box ist auch bekannt als RC (radio control) Servo. Er wird häufig in ferngesteuerten Fahrzeugen und kleinen Robotern zur Steuerung der Bewegung benutzt.

Ein Servomotor wird mit drei Kabeln angeschlossen, eines für den Pluspol (+), die Masse (-) und das Signal (s). Er besteht meist aus einem Gleichstrom-motor, einem Getriebe (Zahnräder), einem Potentio-meter (Positionssensor) und einem Steuerschaltkreis.



Der eingebaute Steuerkreis übersetzt Befehle in Form von Stromstößen in eine Position in Grad. Der Servomotor dreht sich in die eingestellte Richtung und bleibt dort.

Im Gegensatz zu einem DC-Motor, der sich immer weiter dreht, können wir bei einem Servomotor die Drehung in einem Winkel von 0 bis 180 Grad einstellen.

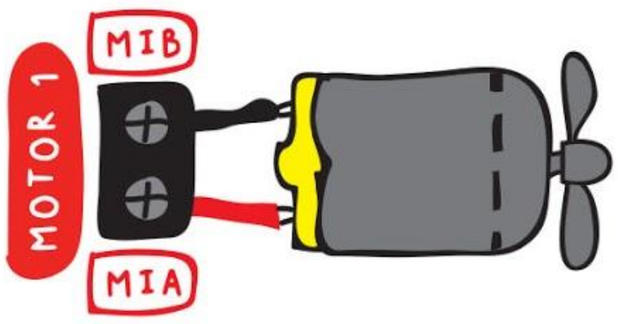


0-180° Drehung

Servomotor

VS

Gleichstrommotor



360° Drehung

youtu.be/okxooamdAP4

HERAUSFORDERUNG

Programmiere EDU:BIT so, dass er wie ein Metronom funktioniert. Nach dem Einschalten soll der Zeiger (der am Servohebel angebracht ist) immer wieder gleichmäßig nach rechts und links pendeln. Das Tempo soll mit dem Potentio Bit eingestellt werden. Wenn der gelbe Knopf (A) gedrückt wird, soll das aktuelle Tempo angezeigt werden (z.B. 120 bpm).

Ein paar Tipps für dich…
#1: Du musst zwei Variablen erstellen: Tempo und Verzögerung.
#2: Der typische Tempobereich geht von 40 bis 200 bpm.
#3: Ein Tempo von 60 bpm (beat per minute = Schläge pro Minute) bedeutet, dass der Zeiger 60 Mal pro Minute von einer Seite zur anderen schwingt, das heißt ein Mal jede Sekunde.
#4: Je schneller das Tempo, desto kleiner die Verzögerung.

"Ein Metronom ist ein Gerät, das durch akustische Impulse in gleichmäßigen Zeitintervallen ein konstantes Tempo vorgibt, das vom Benutzer eingestellt werden kann. Die Zahl, die auf dem Metronom eingestellt wird, gibt an, wie oft das Metronom pro Minute schlagen soll. Die Maßeinheit hierfür heißt „Beats per minute" (bpm). Musiker benutzen das Gerät, um das Spielen zu einem konstanten Tempo zu üben.

- wikipedia -

KAPITEL 10

Mastermind - Kannst du den Code knacken?

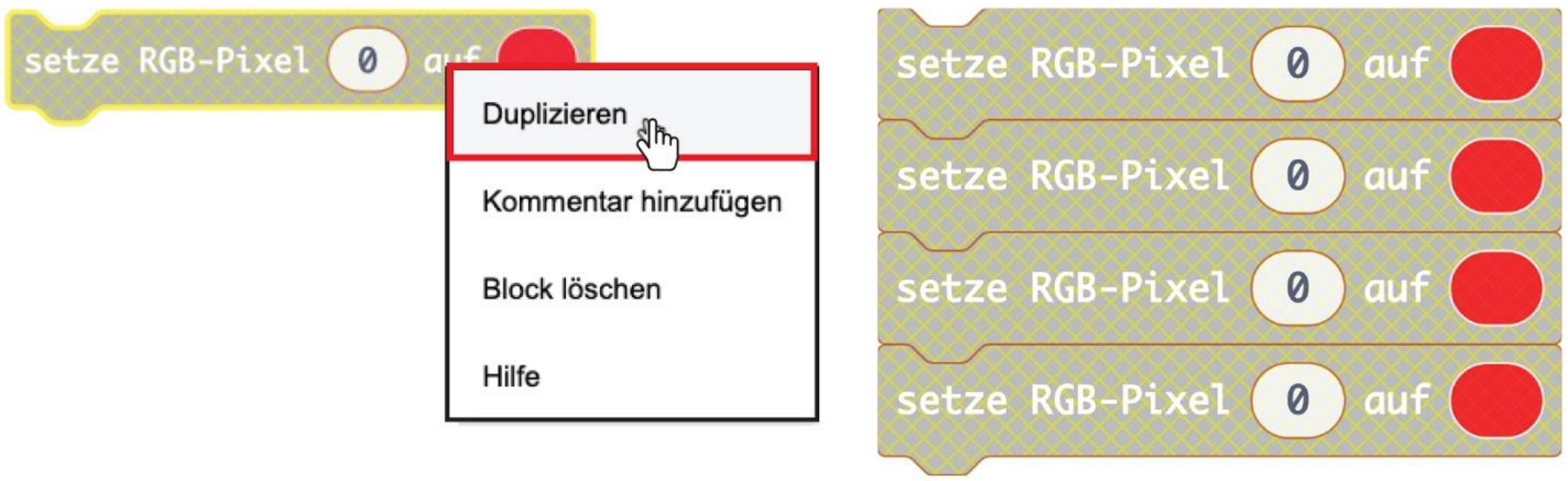
RGB Bit

link.cytron.io/edubit-chapter-10

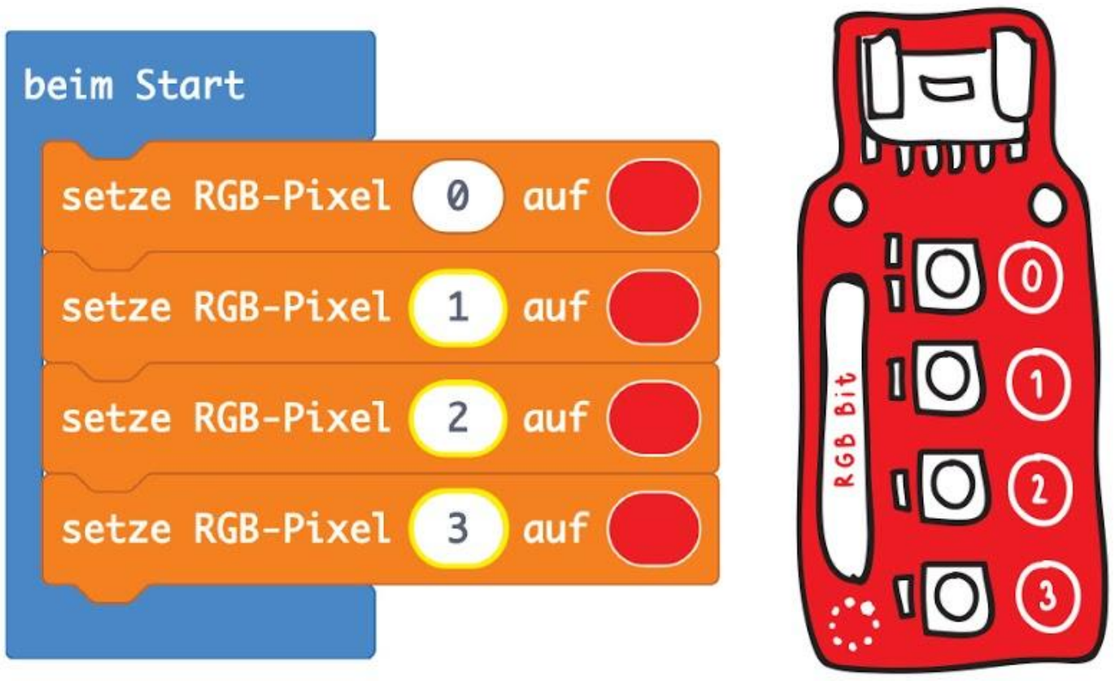
LASS UNS PROGRAMMIEREN!

Schritt 1 Erstelle ein neues Projekt im MakeCode Editor und füge die EDU:BIT-Erweiterung hinzu. Klicke auf **[RGB Bit]** und **[setze RGB-Pixel _ auf _]**.

Schritt 2 Klicke mit der rechten Maustaste auf **[setze RGB-Pixel _ auf _]** und wähle ‘Duplizieren’. Wiederhole bis du vier **[setze RGB-Pixel _ auf _]**-Blöcke hast.

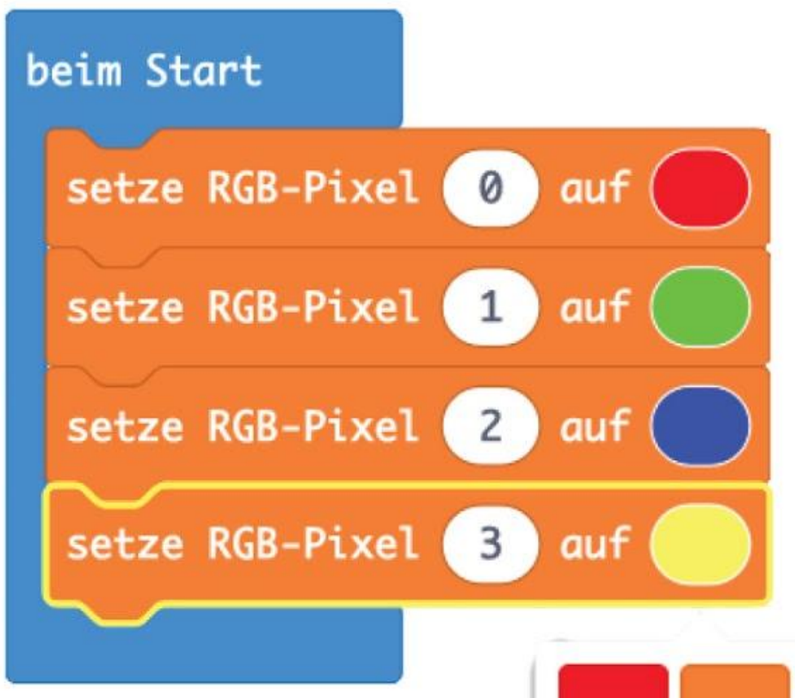


Schritt 3 Lege die Blöcke in den **[beim Start]**-Block. Ändere die RGB-Pixel-Nummer im zweiten, dritten und vierten Block auf **1**, **2** und **3**.



Es gibt 4 RGB-LEDs auf RGB Bit und jedes hat eine eigene Nummer (0 bis 3). Mit dieser Nummer kannst du jede LED einzeln ansteuern.

Schritt 4 Lade den Code auf deinen EDU:BIT.

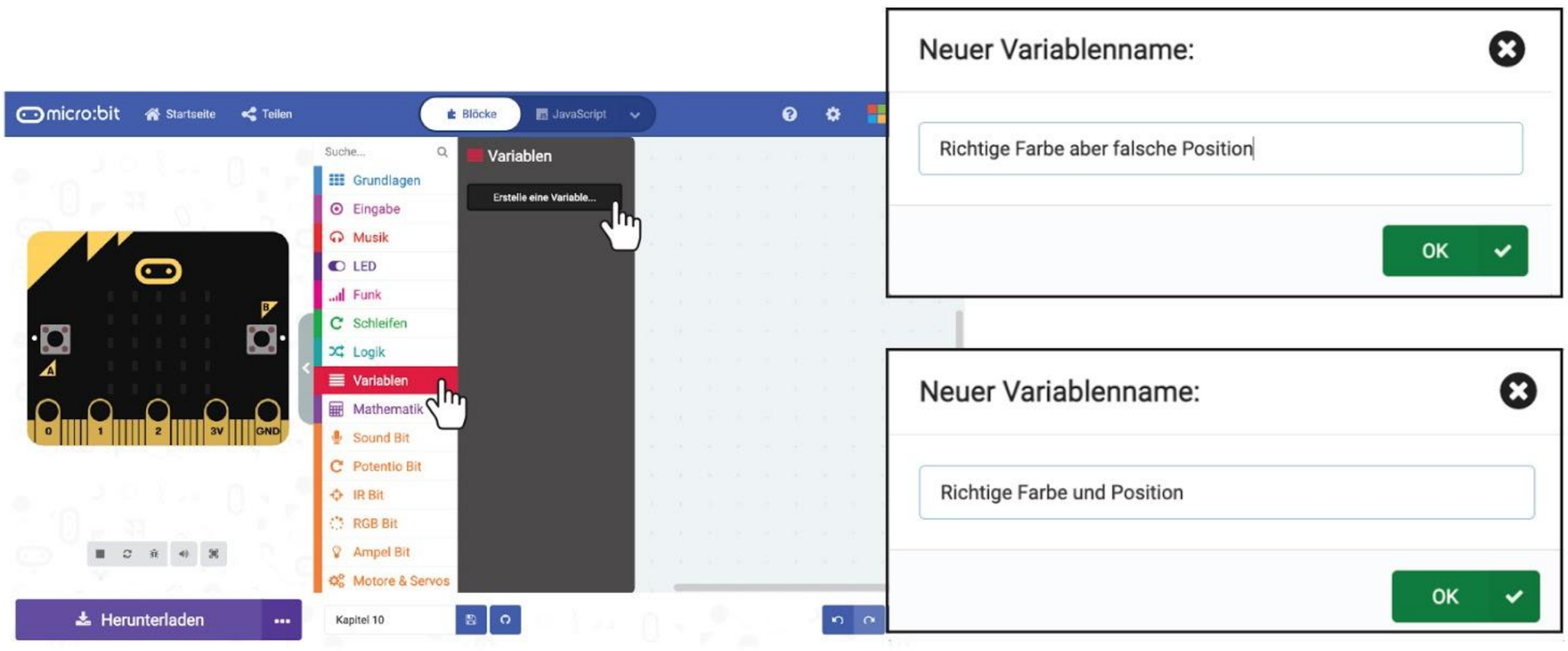


Wenn du die Platine einschaltest, leuchten die vier LEDs auf RGB Bit mit den eingestellten Farben.

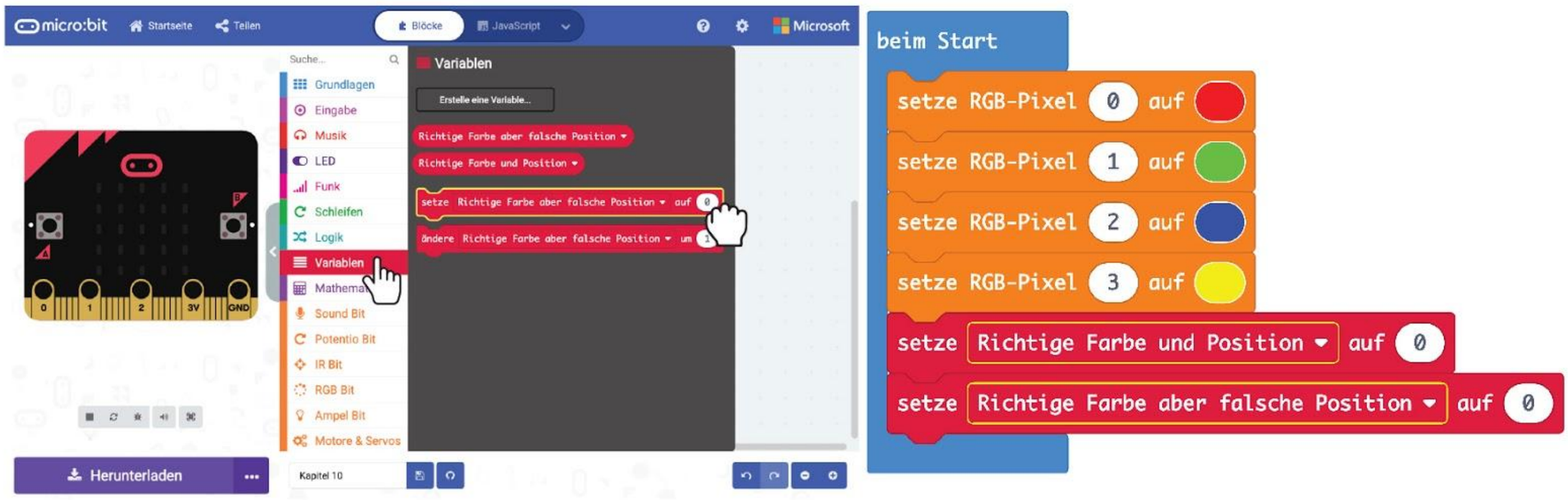
Du kannst eine Farbe einfach ändern, indem du den Block anklickst und eine andere Farbe auswählst.

Probiere es aus!

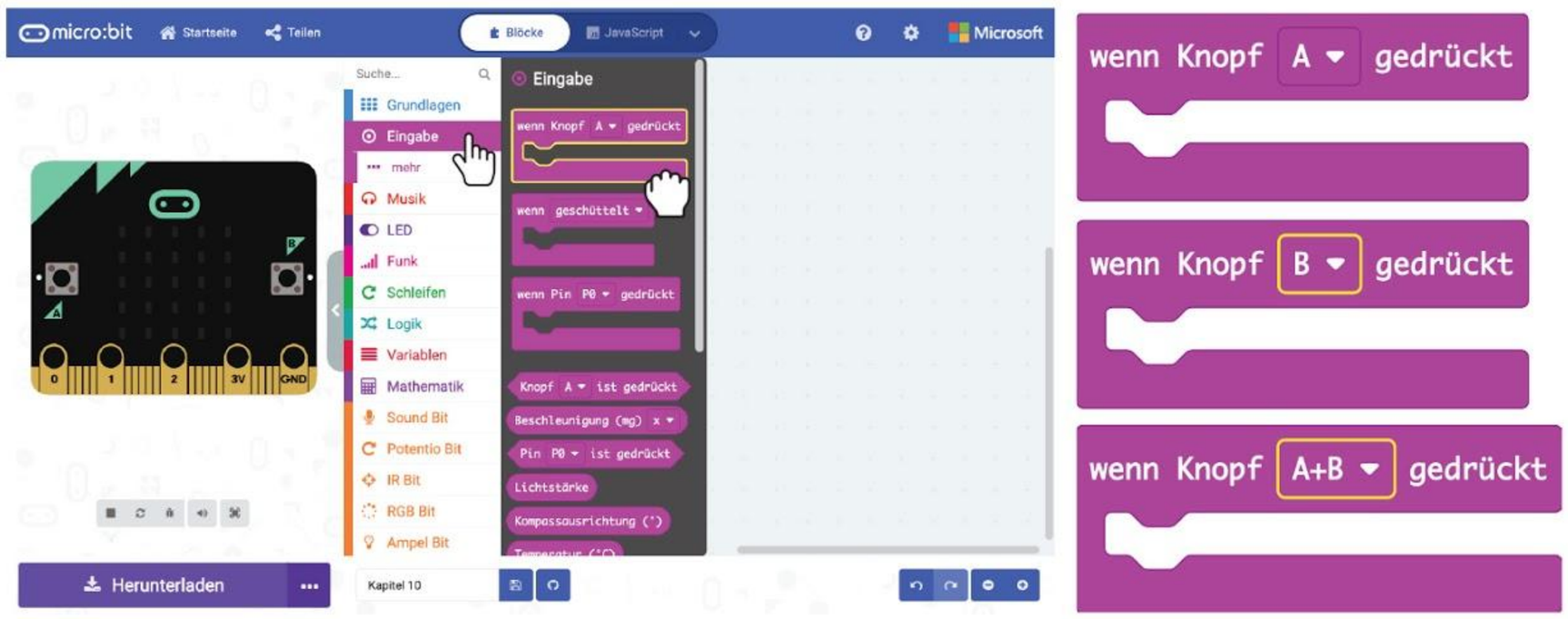
Schritt 5 Füge zwei Variablen hinzu und nenne sie - **“Richtige Farbe und Position”** und **“Richtige Farbe aber falsche Position”**.



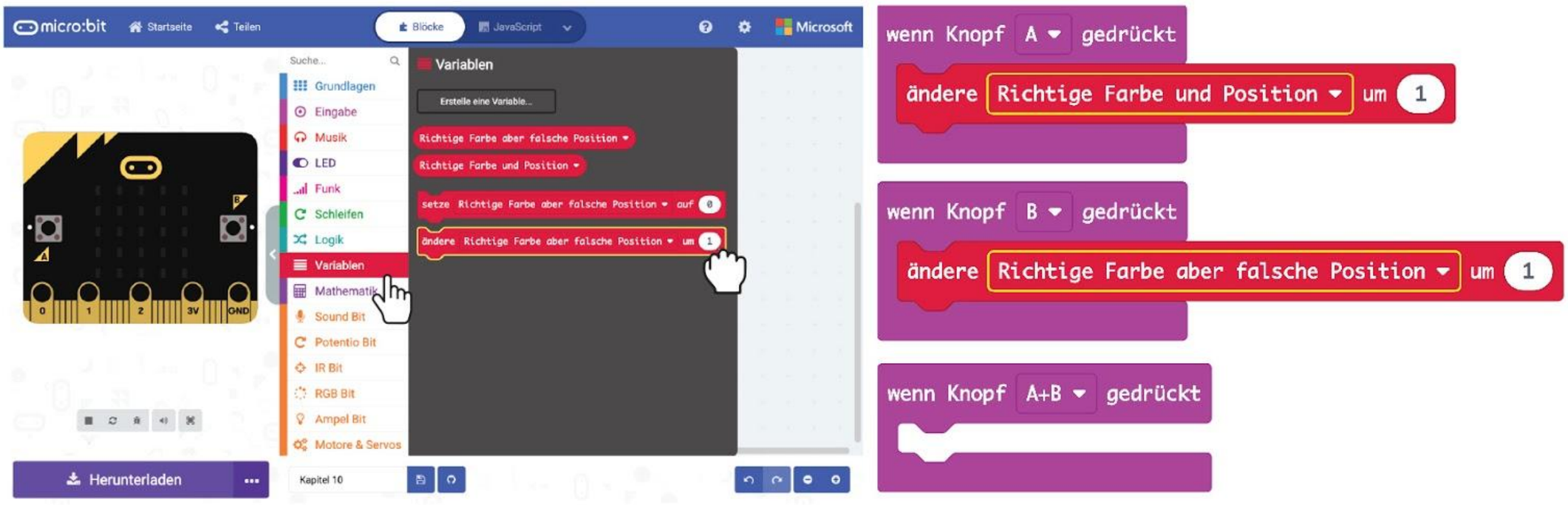
Schritt 6 Klicke unter **[Variablen]** auf den Block **[setze _ auf _]**. Dupliziere den Block und lege beide in den **[beim Start]**-Block. Setze eine Variable auf **“Richtige Farbe und Position”** und die andere auf **“Richtige Farbe aber falsche Position”**.



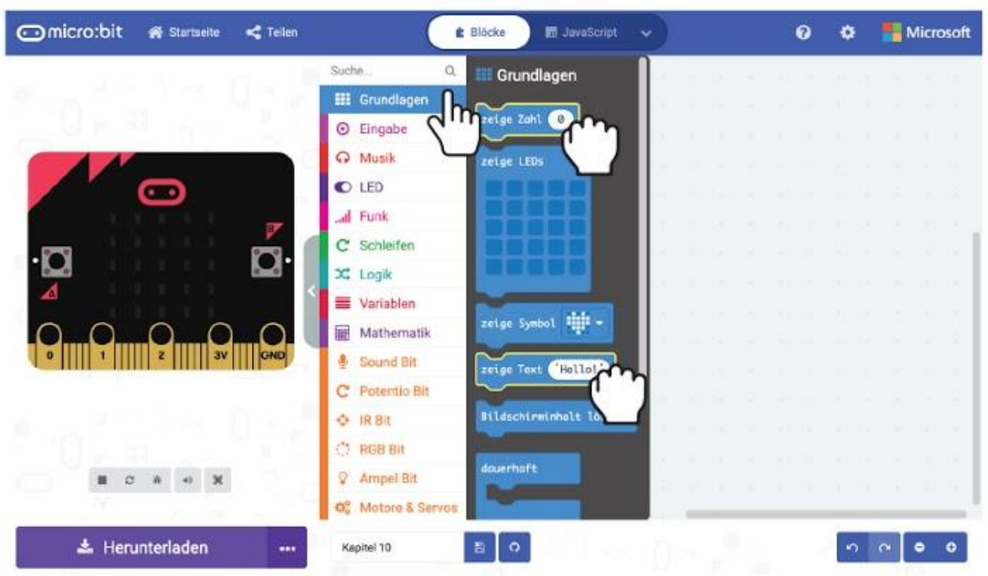
Schritt 7 Klicke auf **[Eingabe]** und **[wenn Knopf _ gedrückt]**. Dupliziere den Block. Ändere den zweiten Block zu **B** und den dritten Block zu **A+B**.

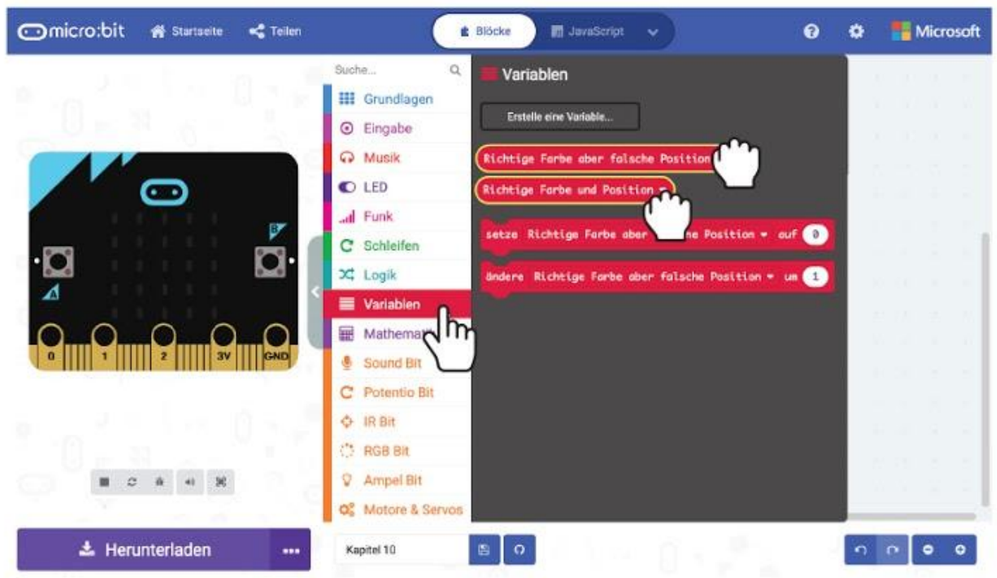


Schritt 8 Klicke auf die Gruppe **[Variablen]** und wähle den Block **[ändere _ um _]**. Dupliziere und lege die Blöcke in **[wenn Knopf A gedrückt]** und **[wenn Knopf B gedrückt]**. Setze eine Variable auf **“Richtige Farbe und Position”** und die andere auf **“Richtige Farbe aber falsche Position”**.

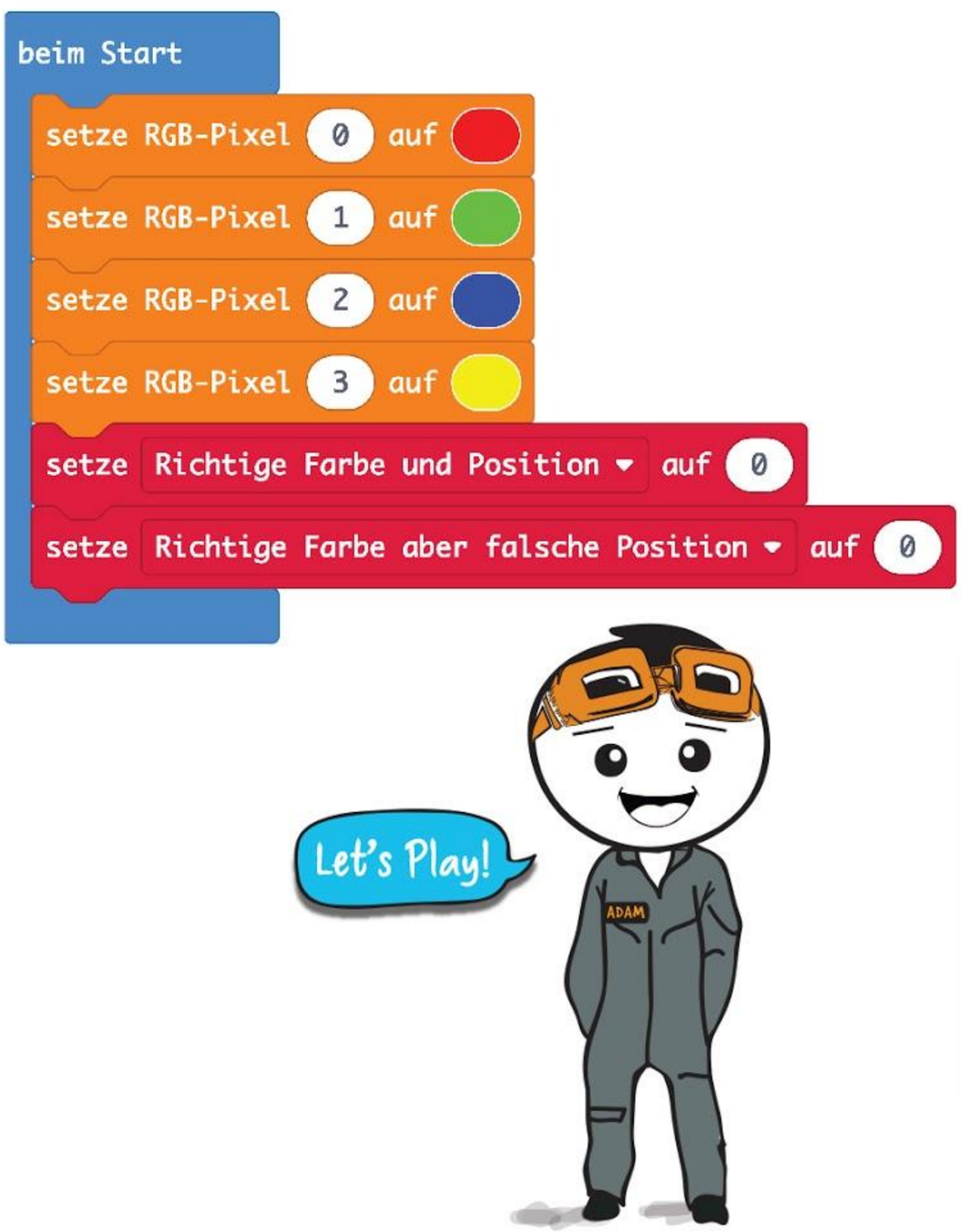


Schritt 9 Füge Blöcke **[zeige Zahl]** und **[zeige Text]** aus der Gruppe **[Fortgeschritten]** und **[Richtige Farbe und Position]** sowie **[Richtige Farbe aber falsche Position]** aus der Gruppe **[Variablen]** hinzu.





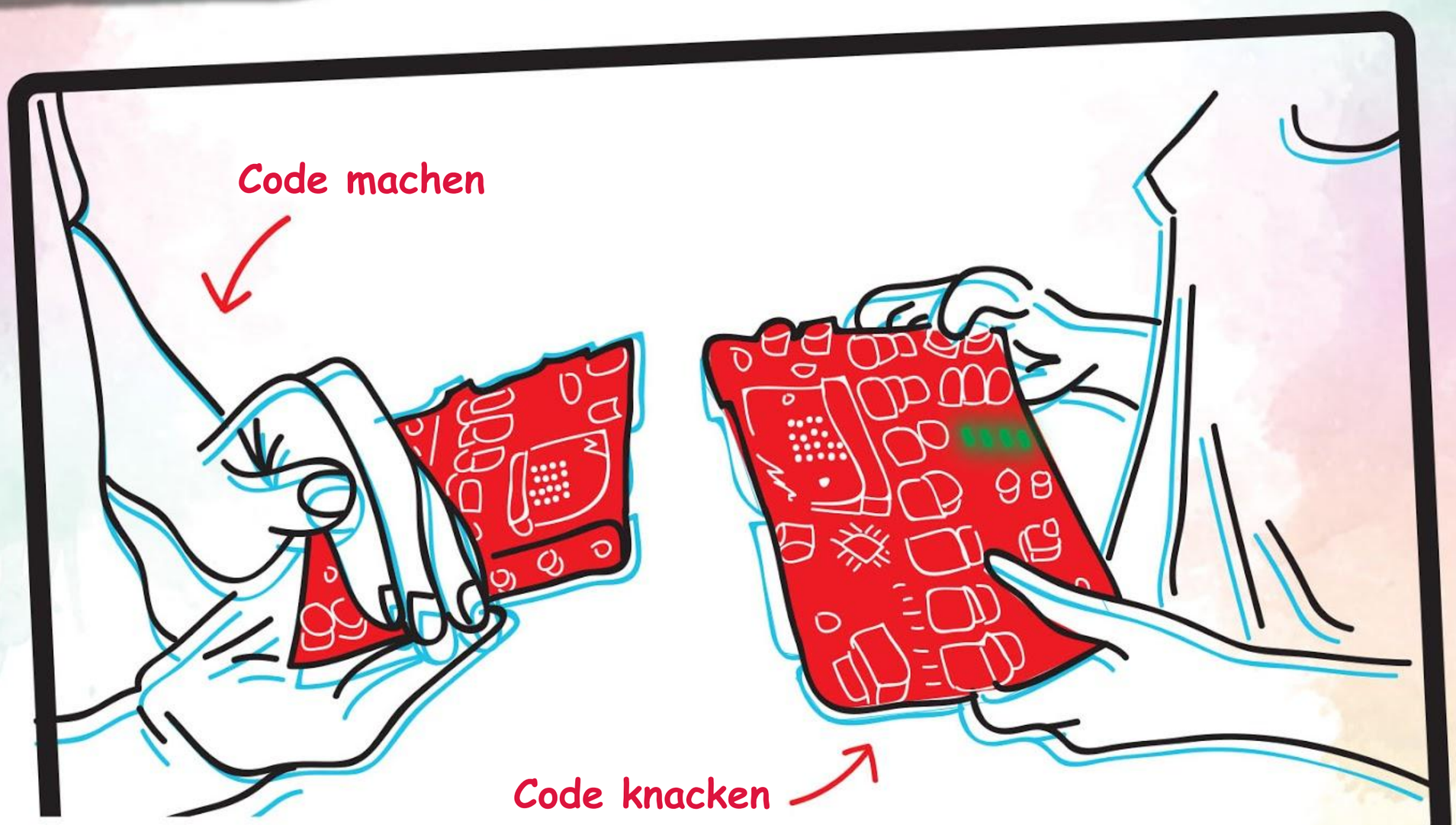
Schritt 10 Ändere “Hello!” in den Blöcken **[zeige Text]** zu “**A =**” und “**B =**”. Hier ist der fertige Code.



Schritt 11 Übertrage den Code auf deinen EDU:BIT. Du kannst jetzt Mastermind spielen.

Spielen wir!

Mastermind - kannst du den Code knacken?



SPIELANLEITUNG:

- Person 1 erfindet einen Code, indem sie die vier LEDs in einer zufälligen Reihenfolge leuchten lässt. Begrenze die Farben am Anfang auf rot und grün. Verdecke deinen Code, damit die andere Spielerin bzw. der andere Spieler nicht sieht, was du eingestellt hast.
- Person 2, versucht den Geheimcode zu erraten. Sie stellt die RGB-LEDs auf deinem EDU:BIT entsprechend ein und zeigt sie dir.
- Du prüfst den Code und drückst wiederholt den gelben Knopf (A) und/oder den blauen Knopf (B) am EDU:BIT der Person, die den Code knackt, um ihr einen Hinweis zu geben, wie viele LEDs mit der “richtigen Farbe und Position” und wie viele mit der “richtigen Farbe aber falschen Position” leuchten.
- Sie kann dann beide Knöpfe zugleich drücken um den Hinweis zu lesen.
- Wiederholt die Schritte 2 und 3 bis die Person, die den Code knackt, die Farbreihenfolge richtig erraten hat (max. 10 Versuche pro Runde).
- Tauscht die Rollen und spielt noch eine Runde.

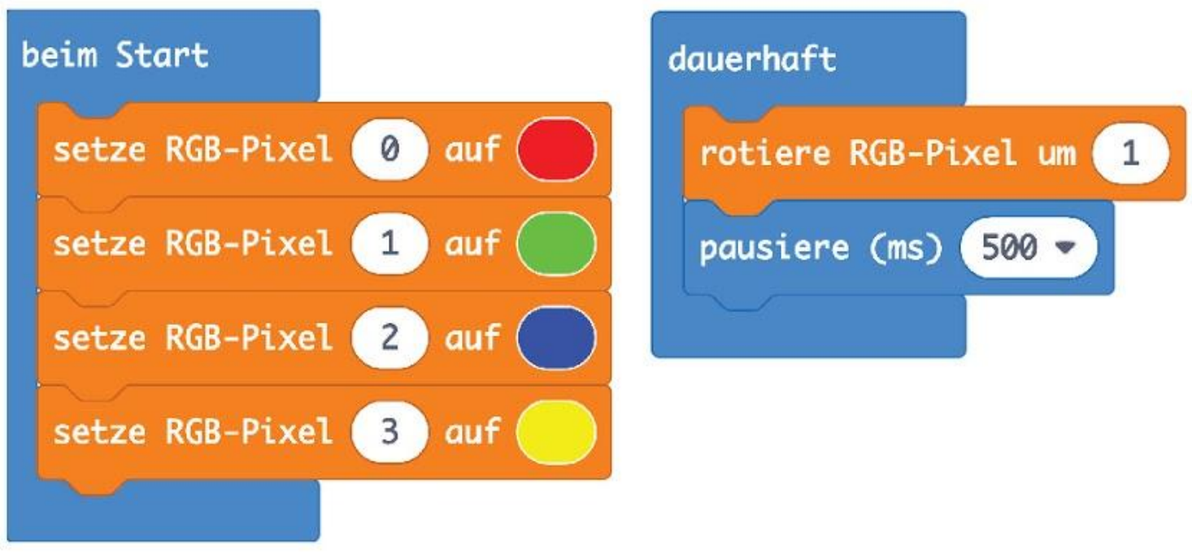
Das Ziel:

Du gewinnst, wenn du erfolgreich den Code knacken kannst (die richtige Farbreihenfolge errätst). Schaffst du es nicht, geht der Sieg an die Person, die den Code erstellt hat.

ENTDECKE NOCH MEHR

#1 Die LEDs auf RGB Bit können ein Regenbogenmuster anzeigen, wenn du den Block **[setze RGB-Pixel _ auf _]** durch **[zeige Regenbogen auf RGB-Pixeln]** ersetzt.

#2 Du kannst einen fortlaufenden Lichteffekt erstellen, indem du einen **[rotiere RGB-Pixel um _]**- Block in einen **[dauerhaft]**-Block legst. Füge einen **[pausiere (ms)]**-Block hinzu, damit das Programm langsamer läuft und du den Effekt sehen kannst. Hier ist der Beispiel-Code:

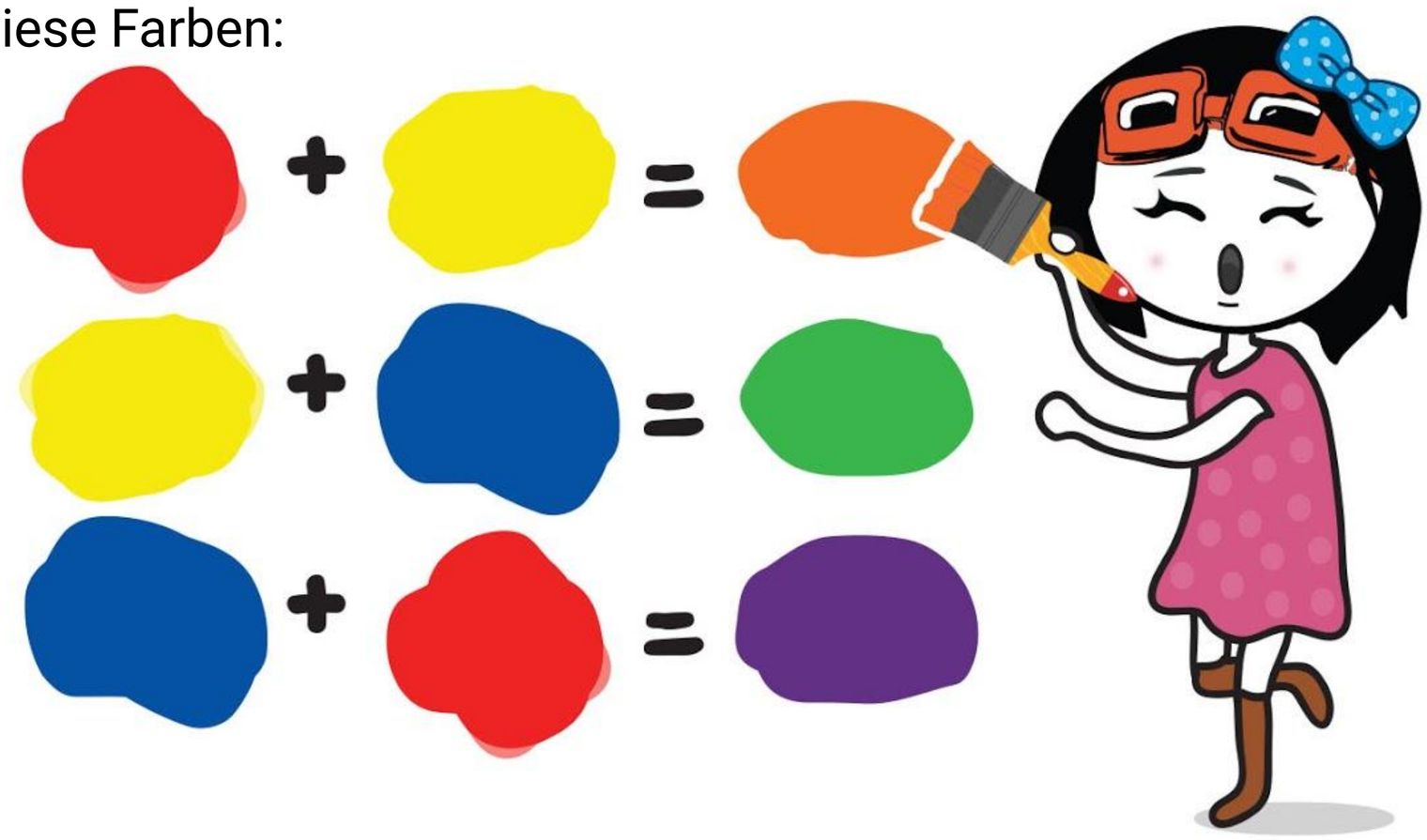


#3 Du kannst die Pixel auch verschieben. Nutze dafür einen **[verschiebe RGB-Pixel um _]**- Block in einem **[dauerhaft]**-Block. Du musst das Programm mit einem **[pausiere (ms)]**-Block verlangsamen, damit du den Effekt siehst. Die Pixel werden einer nach dem anderen abgeschalten. Hier ist der Beispiel-Code:

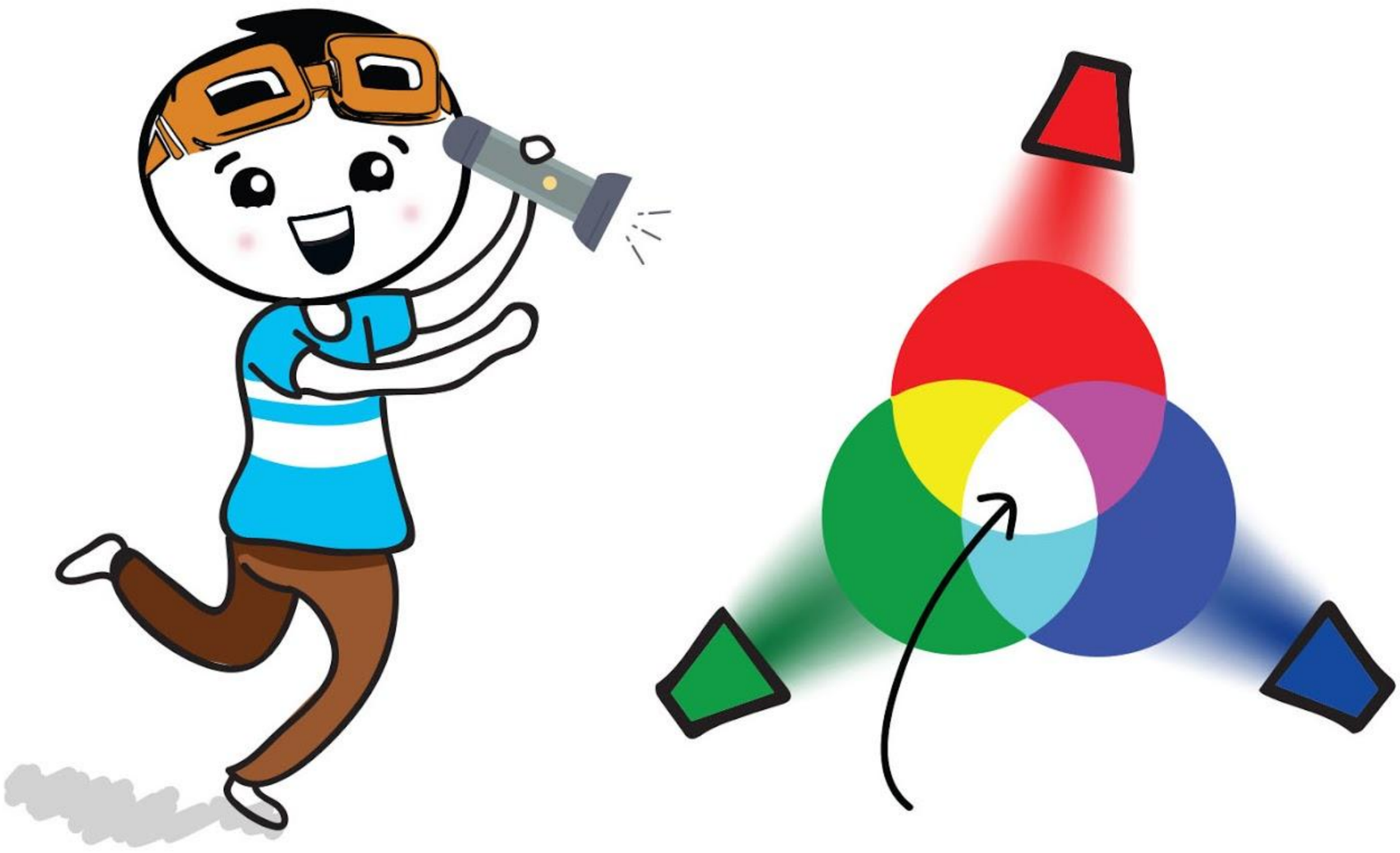
Du kannst die Richtung der beiden Effekte in #2 und #3 ändern, indem du statt einer positiven eine negative Zahl verwendest.

GUT ZU WISSEN!

Du hast im Zeichenunterricht wahrscheinlich gelernt, dass die 3 Hauptfarben Rot, Gelb und Blau sind. Wenn du sie mischst, erhältst du diese Farben:



Für Geräte, die Farben mit Licht darstellen, z.B. deinen Fernseher und Computerbildschirm, wird hingegen das RGB-Farbmodell verwendet.



In diesem Modell sind die Hauptfarben Rot (R), Grün (G) und Blau (B). Wenn sie kombiniert werden, entsteht weißes Licht!

HERAUSFORDERUNG

Programmiere EDU:BIT, dass er wie eine Memory-Trainings-App funktioniert.

Wie funktioniert das?

- Um zu starten, neige EDU:BIT nach links. Auf den LEDs des RGB Bit leuchtet für ein paar Sekunden ein zufälliges Farbmuster auf.
- Du musst es dir merken und das Farbmuster laut aussprechen nachdem die LEDs ausgegangen sind.
- Um deine Antwort zu prüfen, drücke den blauen Knopf (B), damit dasselbe Farbmuster auf den RGB Bit wieder aufleuchtet.
- Richtig geraten? Drücke den gelben Knopf (A) um deinen Punktestand zu erhöhen und anzuzeigen. Falsch? Game Over.
- Du kannst den Schwierigkeitsgrad mit dem Poti auf Potentio Bit verändern. Erhöhe/vermindere, wie lange die LEDs aufleuchten.
- Der Spieler mit den meisten Punkten gewinnt!

Hier sind ein paar Tipps:
#1: Du brauchst zwei Variablen - Punkte und Muster.
#2: Du musst die Farbmuster voreinstellen (mit der Farbe für jedes RGB-LED). Verwende mehr Farben und voreingestellt Muster um das Spiel schwieriger zu machen. Für jüngere Spielerinnen oder Spieler kannst du das Spiel einfacher machen, indem du die Farben/Muster begrenzt.

BONUSKAPITEL

“Simon sagt” mit LEDs

Funkübertragung

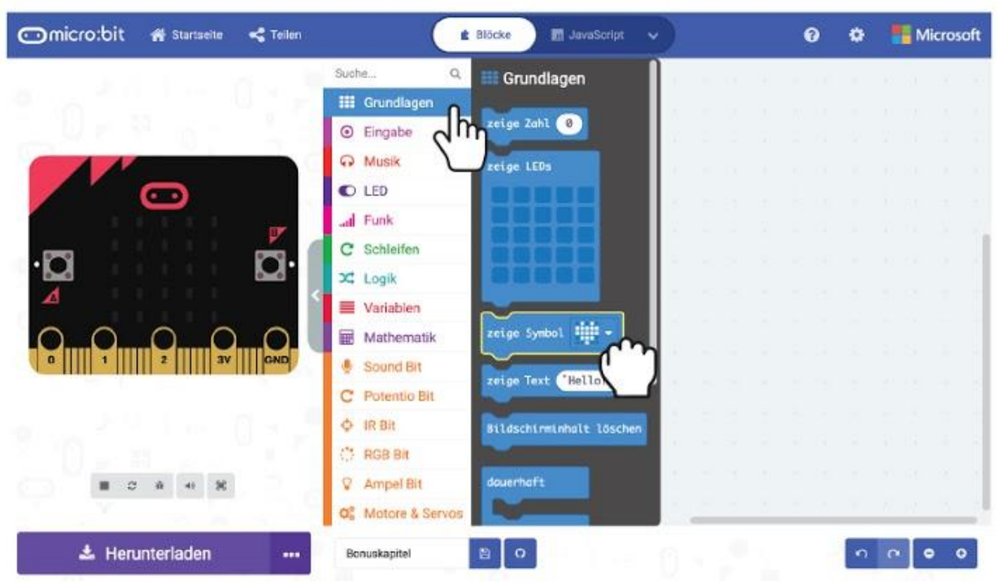
link.cytron.io/edubit-bonus-chapter

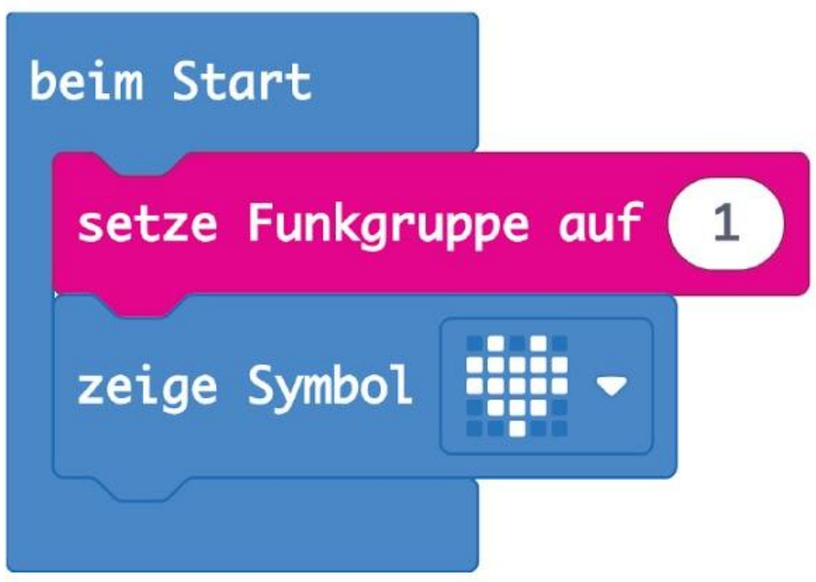
Bei jeder Form der Kommunikation brauchen wir zumindest zwei Beteiligte - einen Sender und einen Empfänger. Bei diesem Spiel brauchen wir zwei EDU:BITs, die miteinander kommunizieren, indem sie Funksignale senden und empfangen.

LASS UNS PROGRAMMIEREN!

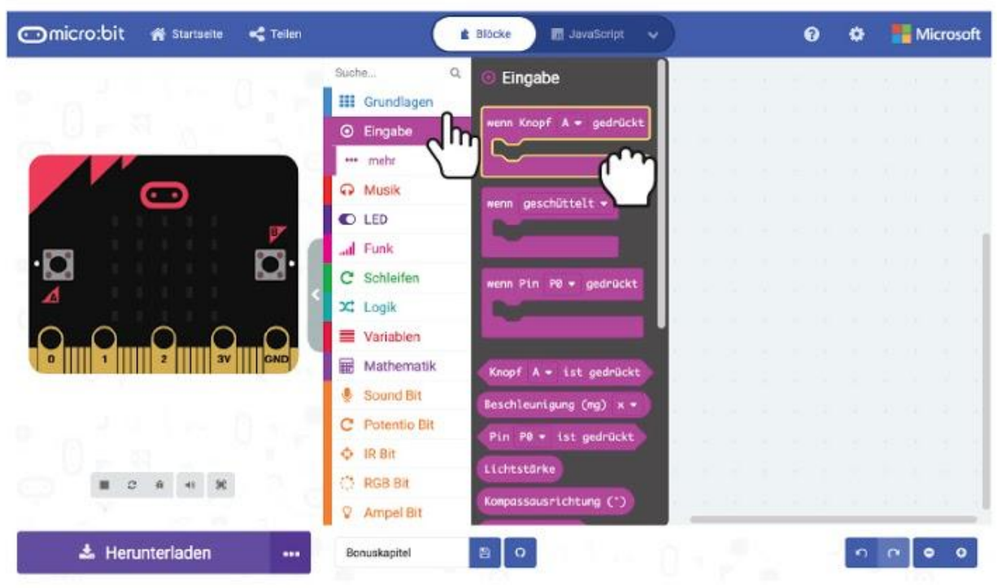
Schritt 1 Erstelle ein neues Projekt im MakeCode Editor und füge die EDU:BIT-Erweiterung hinzu. Klicke auf die Kategorie **[Funk]** und den Block **[setze Funkgruppe auf _]**. Ziehe den Block in den Block **[beim Start]**.

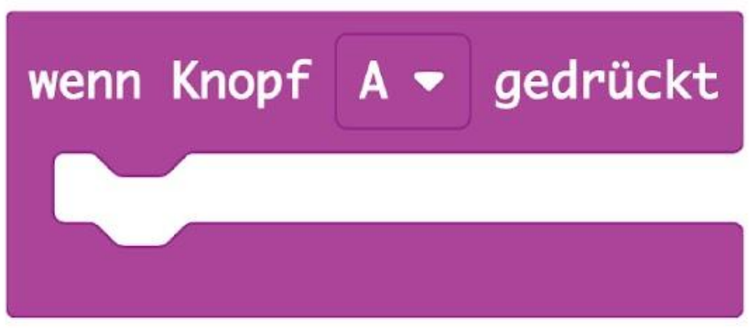
Schritt 2 Klicke auf **[Grundlagen]** und füge den Block **[zeige Symbol]** zu deinem Programm hinzu.



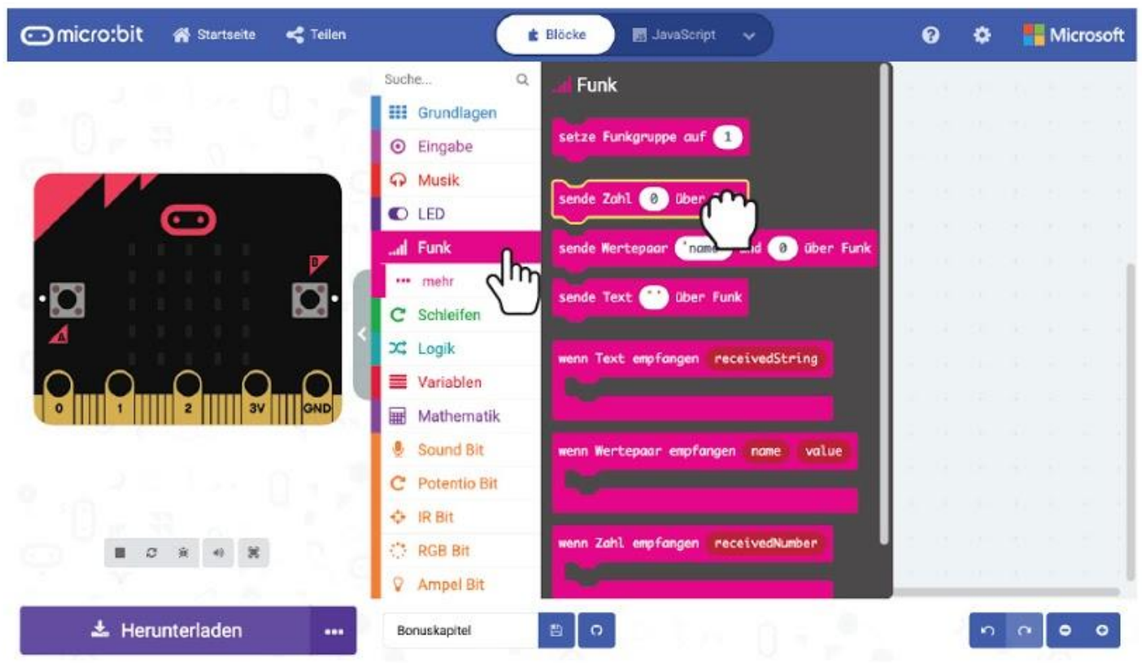


Schritt 3 Klicke unter **[Eingabe]** auf den Block **[wenn Knopf _ gedrückt]**.

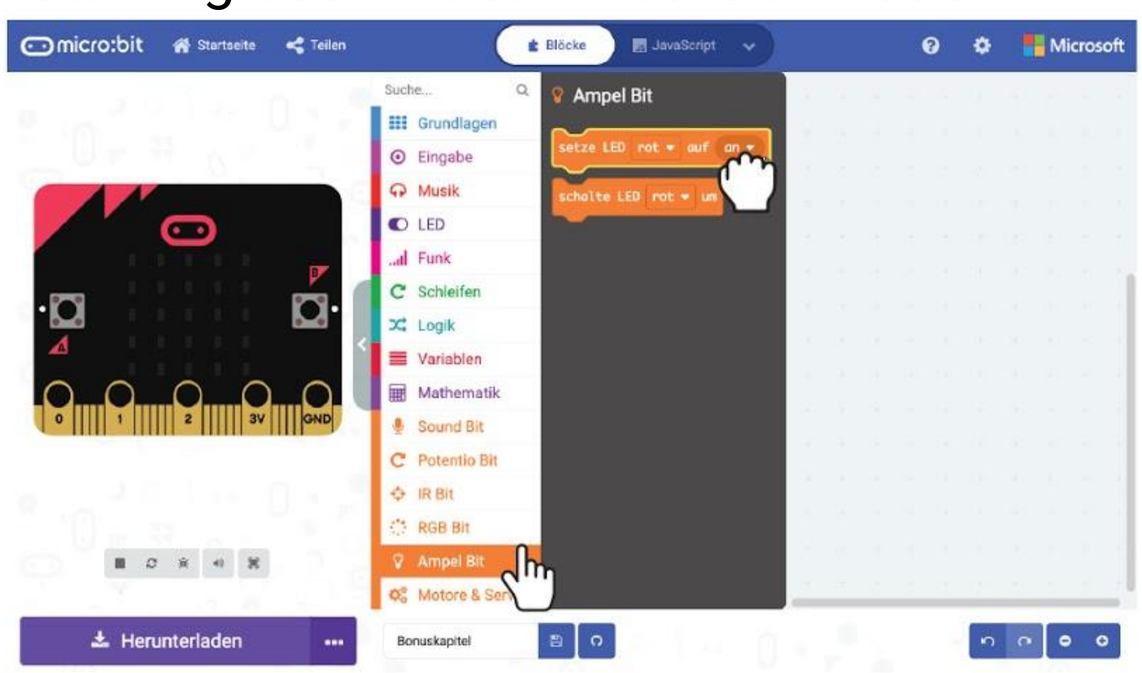




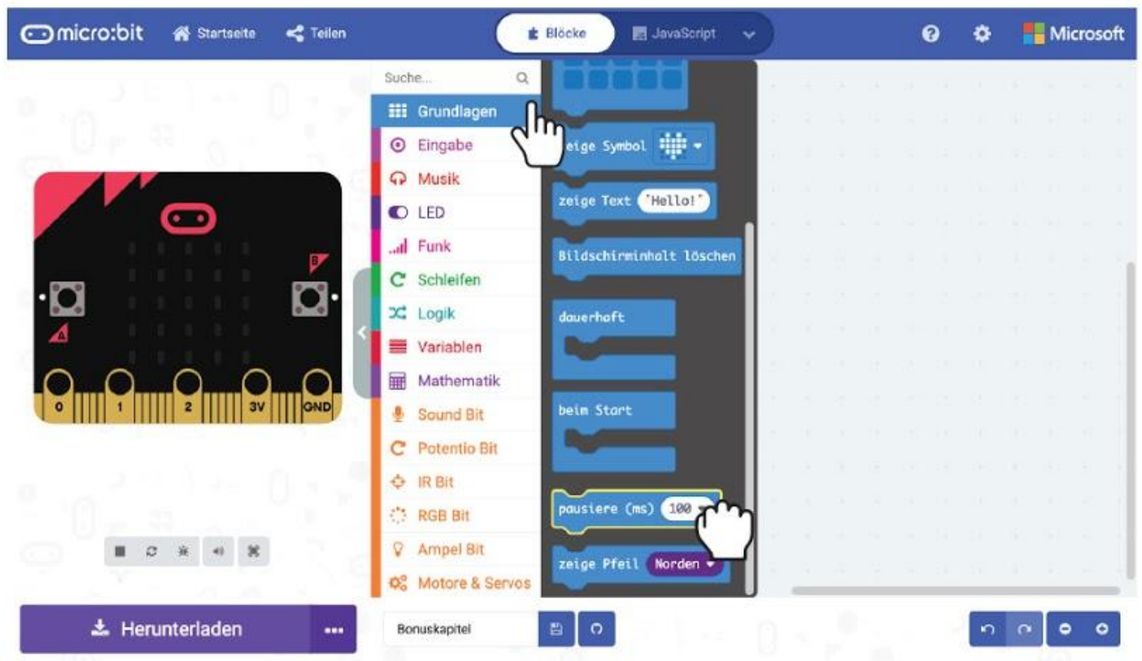
Schritt 4 Klicke unter **[Funk]** auf den Block **[sende Zahl _ über Funk]**. Ändere den Wert auf **1**.



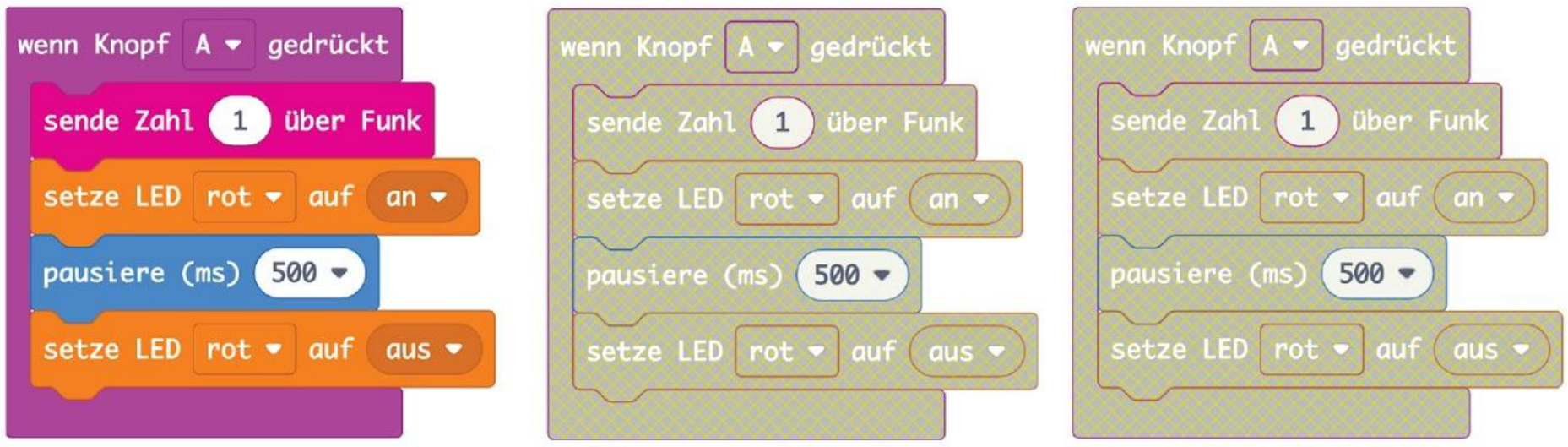
Schritt 5 Klicke in der Gruppe **[Ampel Bit]** auf den Block **[setze LED _ auf _]**. Dupliziere ihn und lege die Blöcke in den Block **[wenn Knopf A gedrückt]**. Ändere die Einstellung des zweiten Blocks auf **‘aus’**.



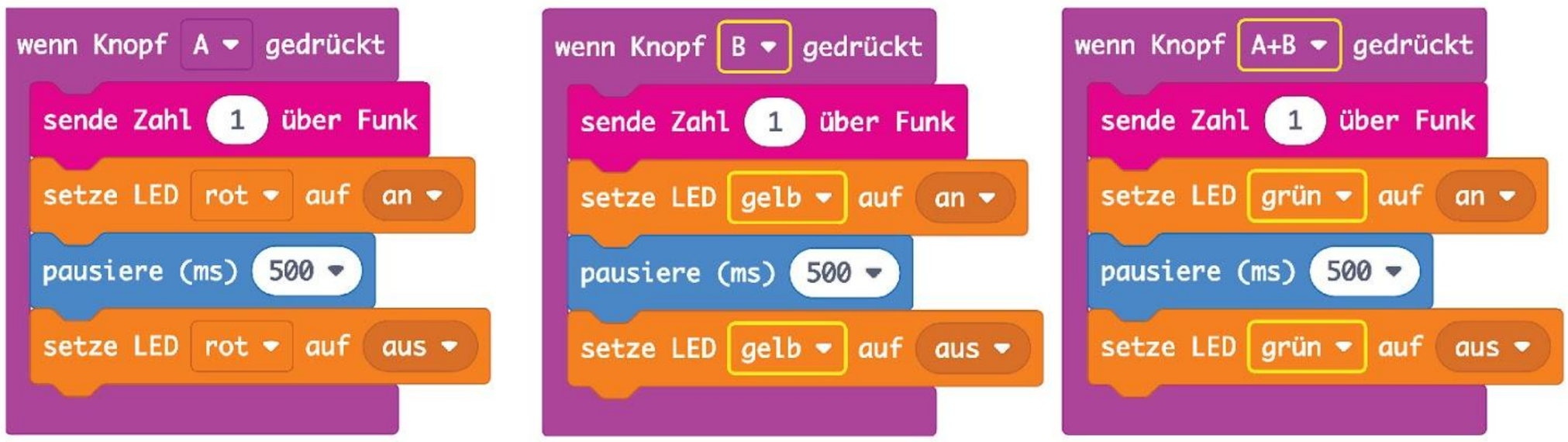
Schritt 6 Klicke auf die Kategorie **[Grundlagen]** und wähle **[pausiere (ms) _]**. Lege den Block zwischen die beiden **[setze LED _ auf _]**-Blöcke und gib **500** ein.



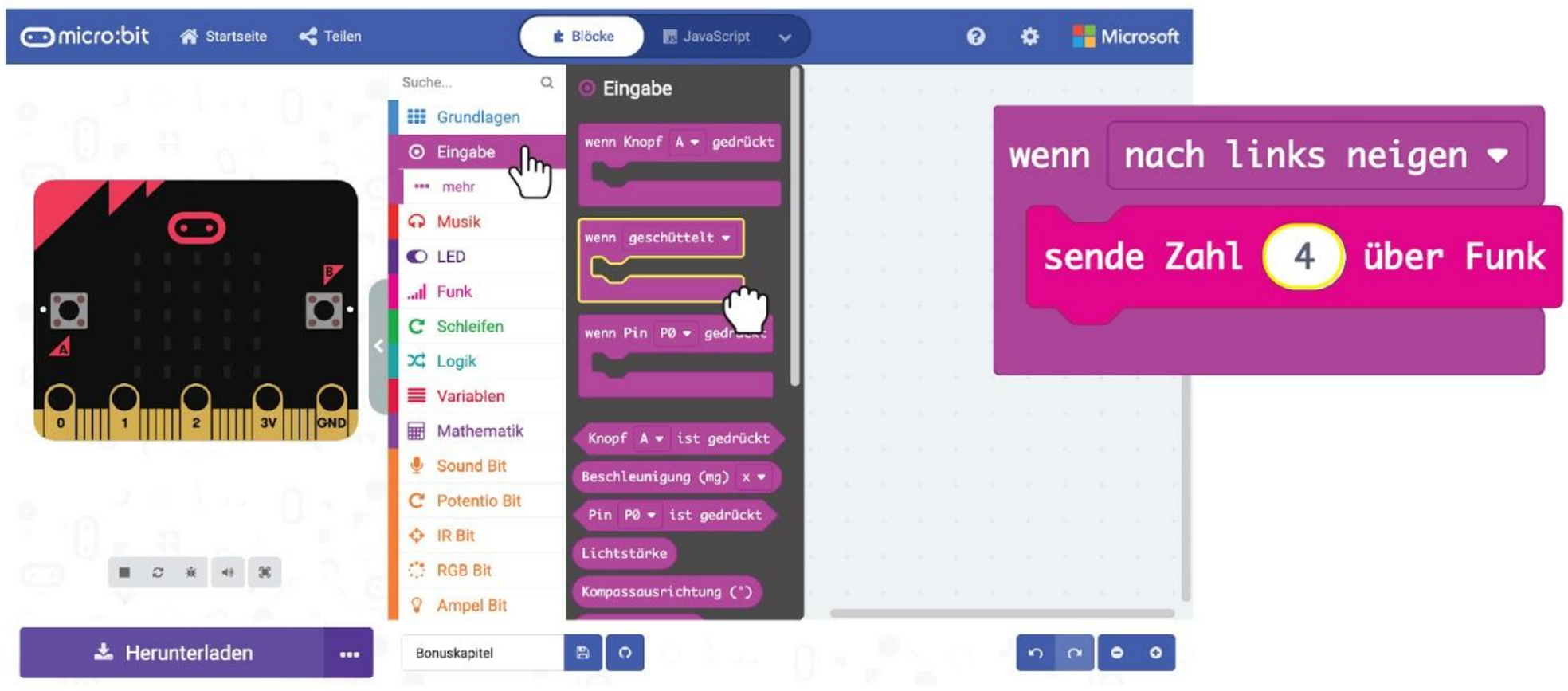
Schritt 7 Klicke mit der rechten Maustaste auf den Block **[wenn Knopf A gedrückt]** und dupliziere, bis du drei gleiche Blöcke hast.



Schritt 8 Ändere die Einstellungen für Knopf, Zahl und LED-Farbe folgendermaßen:

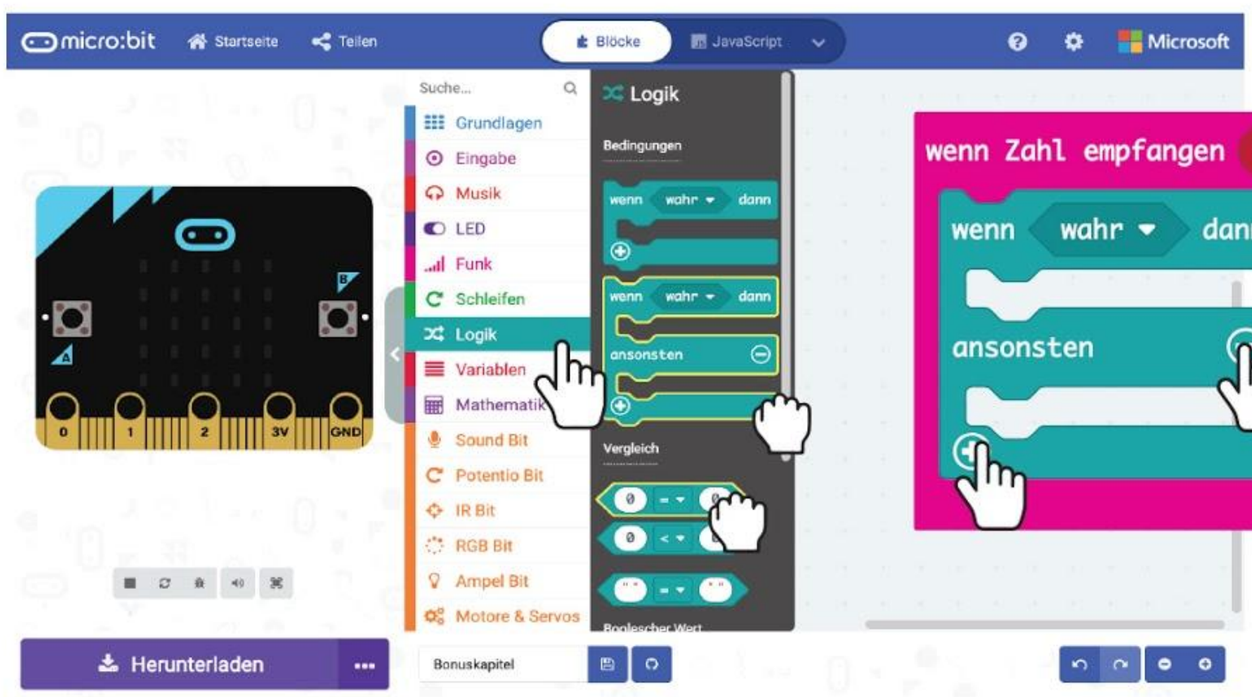


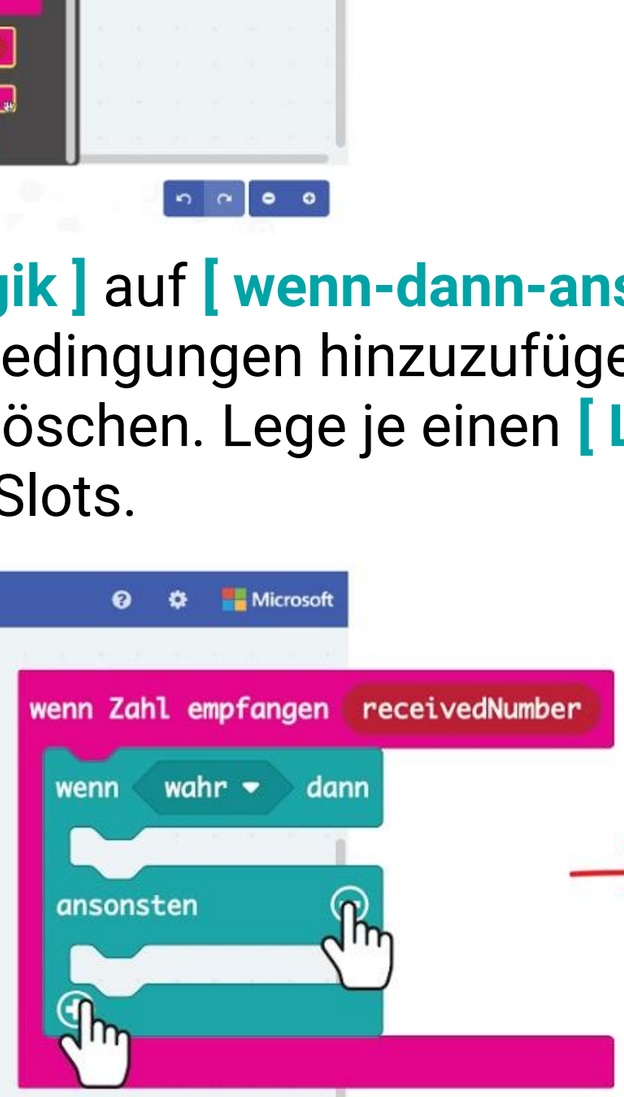
Schritt 9 Klicke unter **[Eingabe]** auf den Block **[wenn geschüttelt]**. Ändere ihn auf ‘**nach links neigen**’. Klicke in der Gruppe **[Funk]** auf **[sende Zahl _ über Funk]**. Ändere die Zahl auf **4**.

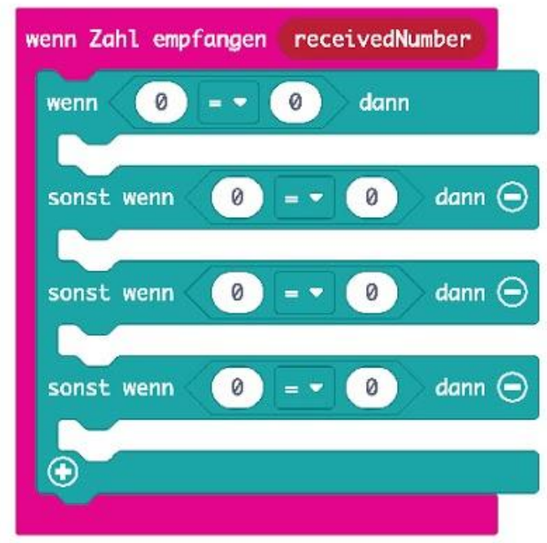


Schritt 10 Klicke unter **[Radio]** auf den Block **[wenn Zahl empfangen _]**.

Schritt 11 Klicke in der Gruppe **[Logik]** auf **[wenn-dann-ansonsten]**. Klicke auf das (+)-Symbol um drei **[sonst wenn]**-Bedingungen hinzuzufügen und auf das (-)-Symbol um die **[ansonsten]**-Bedingung zu löschen. Lege je einen **[Logik] : [_ = _]**-Block in die freien “wenn”- und “wenn dann”-Slots.

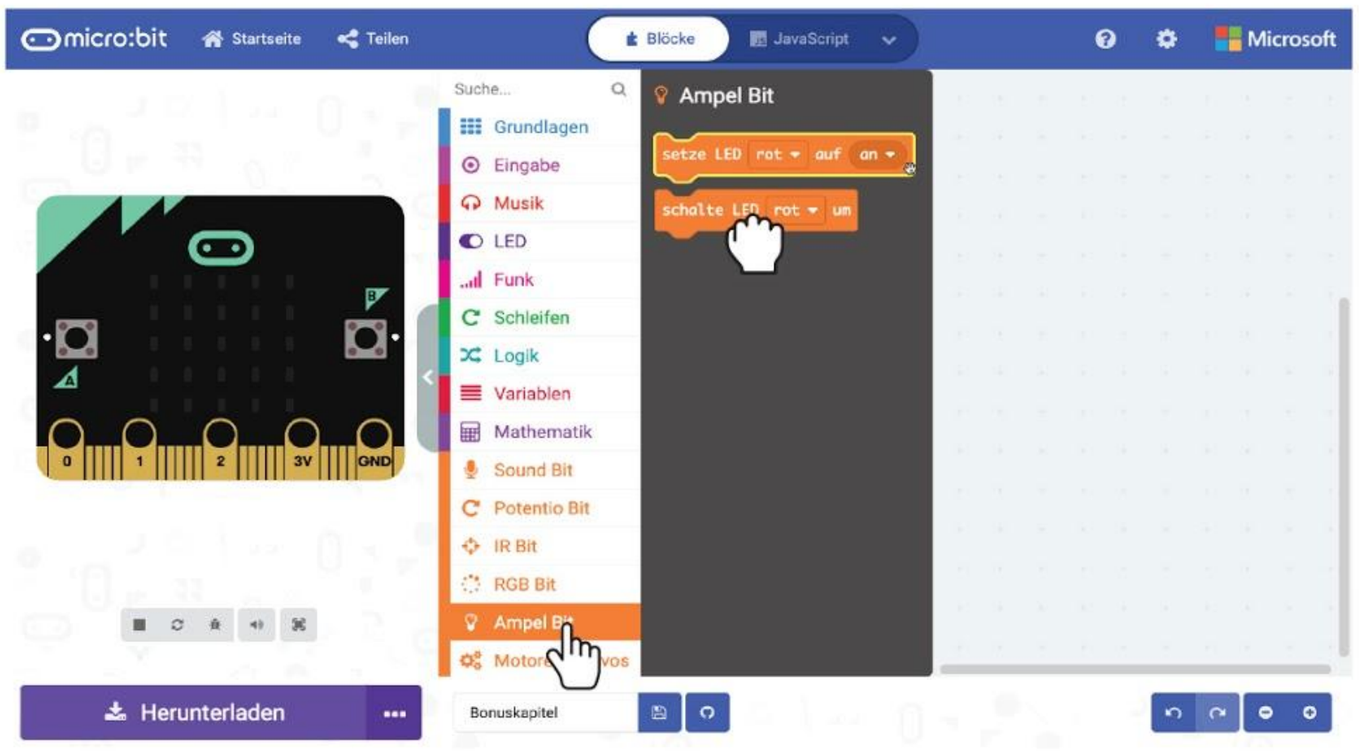




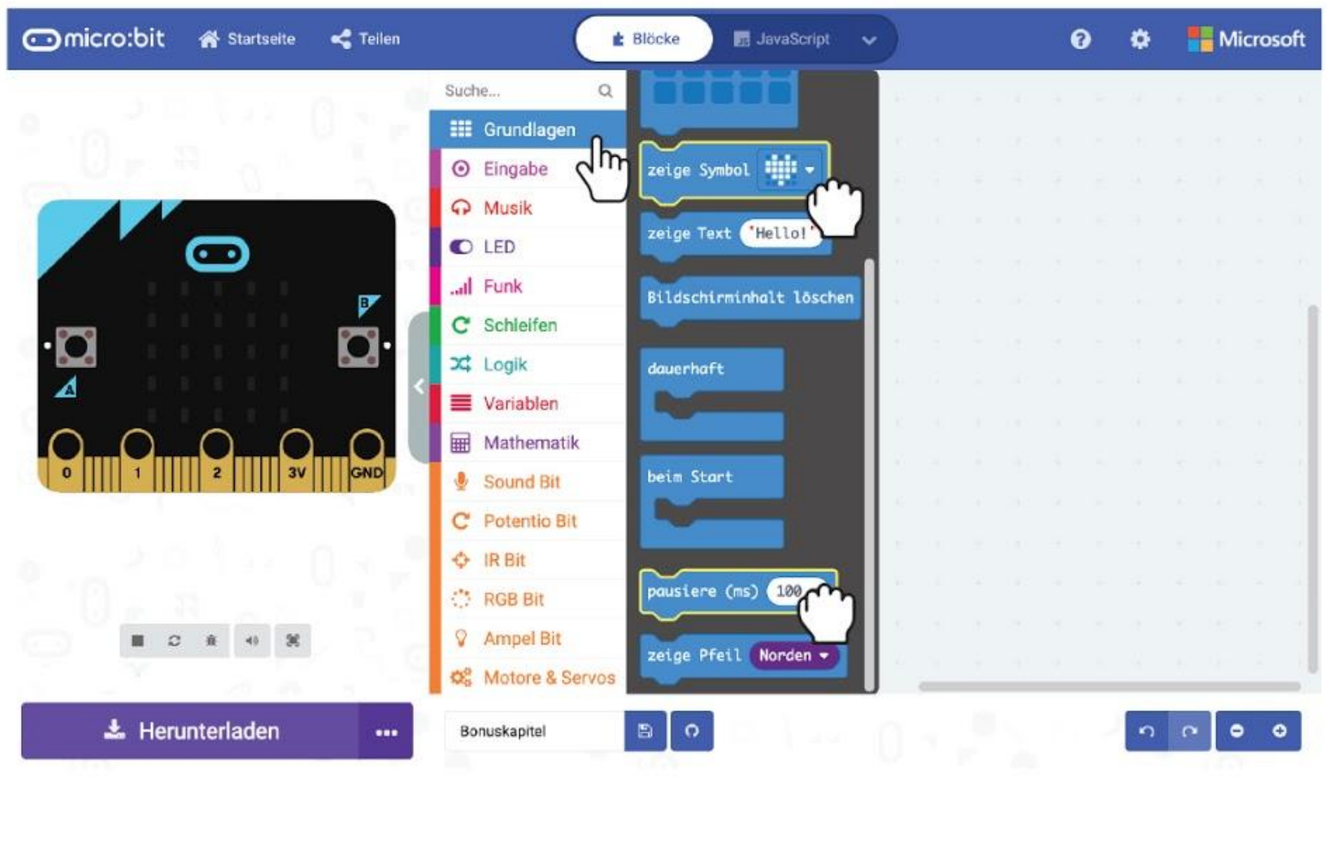


Schritt 12 Ziehe die Variable **‘receivedNumber’** in die Vergleichs-Blöcke und ändere die Werte auf **1, 2, 3** und **4**.

Schritt 13 Klicke auf **[Ampel Bit]** und wähle den Block **[setze LED _ auf _]**. Dupliziere ihn und lege die Blöcke in die freien Bereiche. Ändere die Einstellungen wie hier gezeigt.

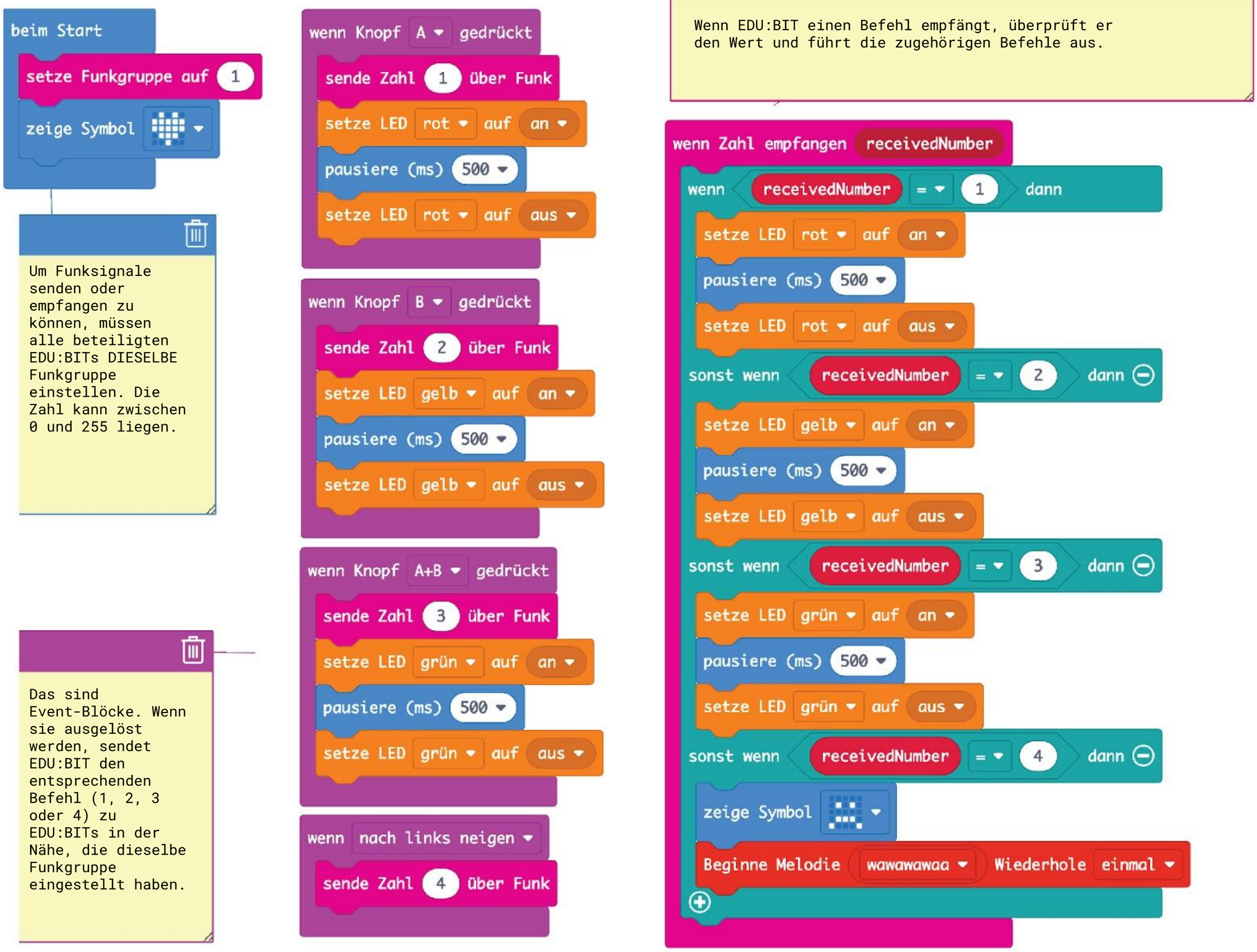


Schritt 14 Klicke auf **[Grundlagen]** und **[pausiere (ms) _]**. Dupliziere und lege die Blöcke zwischen die **[setze LED _ auf _]**-Blöcke. Ändere die Werte auf **500**. Lege einen **[Grundlagen] : [zeige Symbol]**-Block in den letzten “sonst wenn”-Block und ändere das Symbol zu “traurig”.



Schritt 15 Füge zum Schluss aus der Gruppe **[Musik]** einen Block **[Beginne Melodie _ Wiederhole _]** hinzu. Ändere die Melodie zu “**wawawawaa**” (oder eine andere Melodie deiner Wahl).

Das ist der vollständige Code:

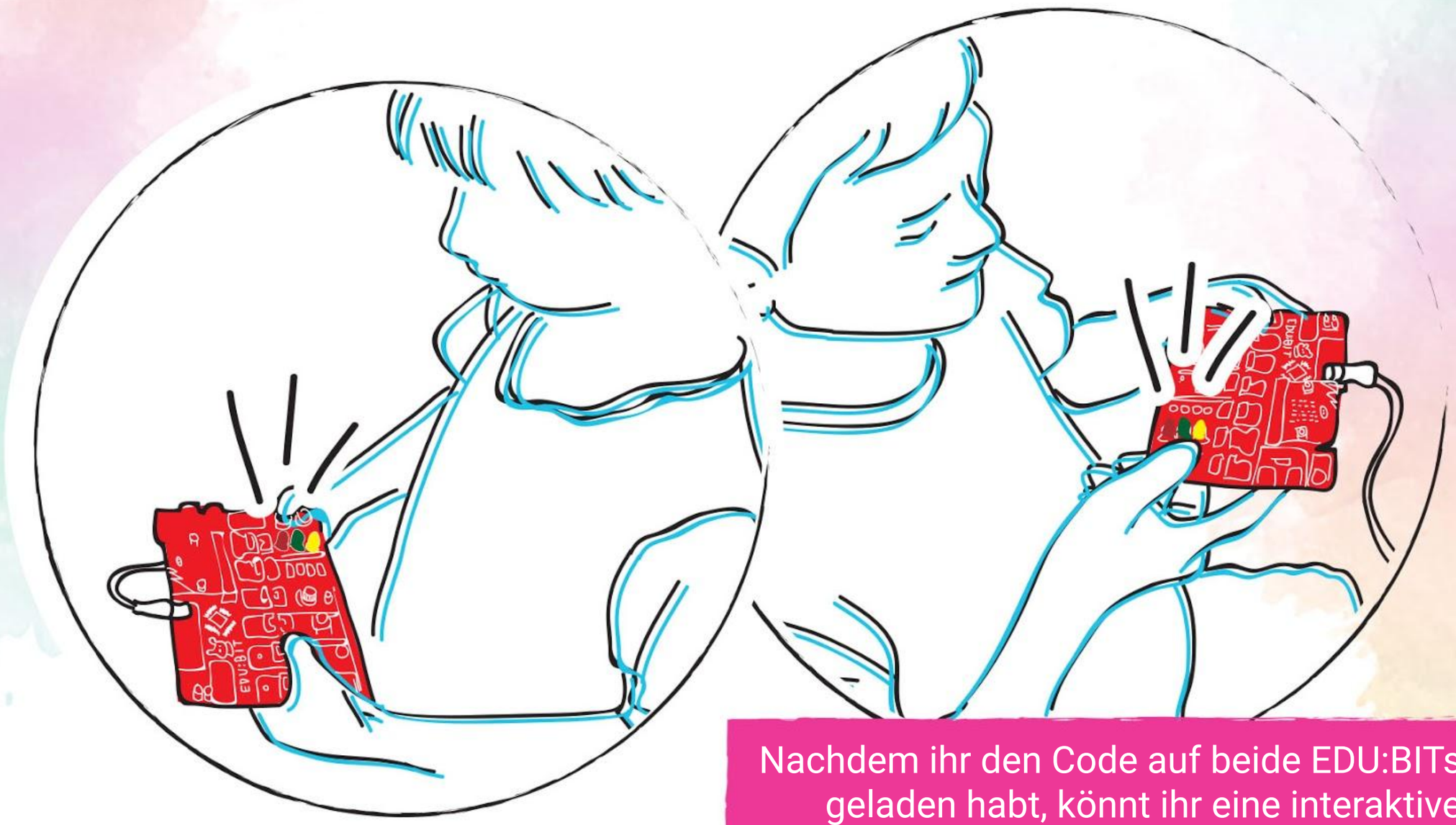


Schritt 16 Lade den fertigen Code auf deinen EDU:BIT und auf den einer Freundin oder eines Freundes.

Wenn beide EDU:BITs eingeschaltet sind, kannst du mittels Funkübertragung die LEDs auf dem EDU:BIT einer anderen Person blinken lassen. Umgekehrt kann die andere Person die LEDs auf deinem EDU:BIT aktivieren, indem sie auf ihrem Gerät die Knöpfe drückt.

Spielen wir!

"Simon sagt" mit LEDs



Nachdem ihr den Code auf beide EDU:BITs geladen habt, könnt ihr eine interaktive Version von "Simon sagt" spielen.

SPIELANLEITUNG:

Beide sind abwechselnd "Simon". Wenn du dran bist, drücke die Knöpfe um die roten, gelben und grünen LEDs aufleuchten zu lassen.

- Am Anfang lässt Person 1 EINE LED leuchten.
- Person 2 beobachtet und lässt dieselbe LED leuchten, gefolgt von einer anderen.
- So geht es weiter. Jede Person wiederholt die letzte Reihenfolge und lässt eine LED zusätzlich aufleuchten.
- Wenn jemand die falsche LED leuchten lässt (oder in der falschen Reihenfolge) neigt die oder der andere den EDU:BIT um das Spiel zu beenden.
- Für ein neues Spiel muss die Person, die verloren hat, ihren EDU:BIT neu starten.

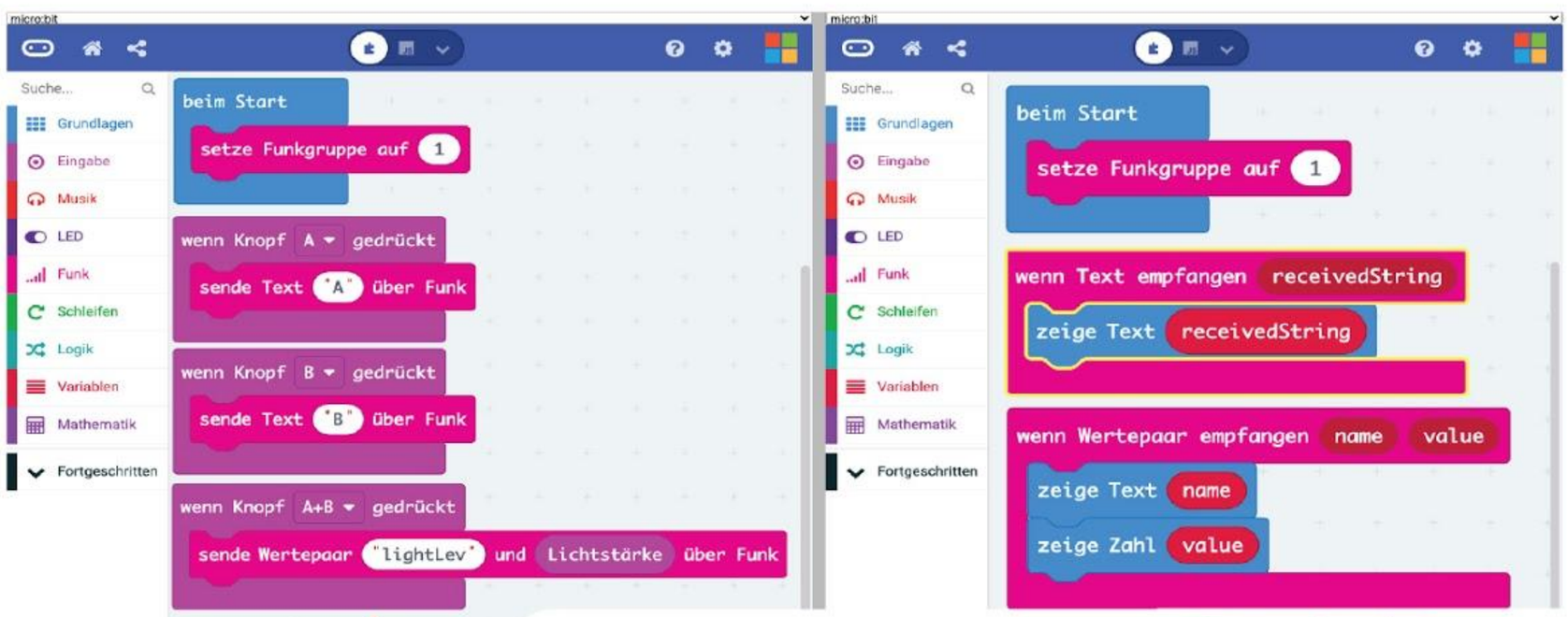
Dieses Spiel ist am Anfang einfach, aber es wird schnell schwieriger. Um zu gewinnen, musst du aufmerksam beobachten. Es ist ein tolles Spiel um dein Gedächtnis zu trainieren.

ENTDECKE NOCH MEHR

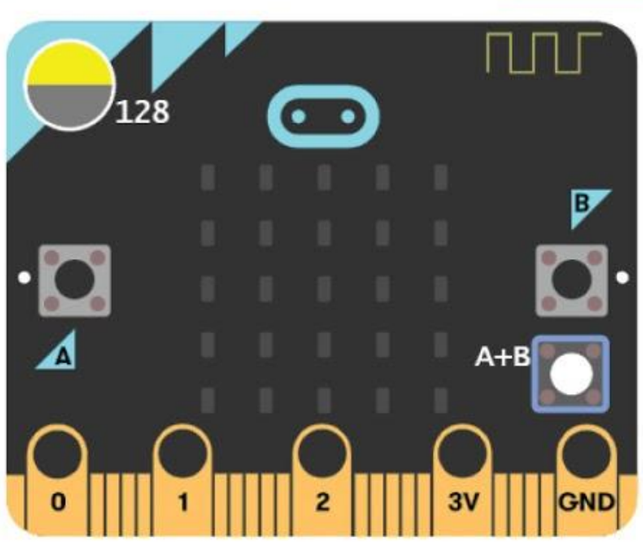
#1 Neben Zahlen kannst du mit dem **[sende Text _ über Funk]**-Block auch Textnachrichten schicken. Mit dem Block **[wenn Text empfangen receivedString]** kannst du Textnachrichten empfangen. Die maximale Textlänge ist 19 Buchstaben.

#2 Benutze die Blöcke **[sende Wertepaar _ und _ über Funk]** und **[wenn Wertepaar empfangen name value]**, wenn du einen Text und eine Zahl gemeinsam übertragen willst. Die maximale Textlänge ist hier 8 Buchstaben.

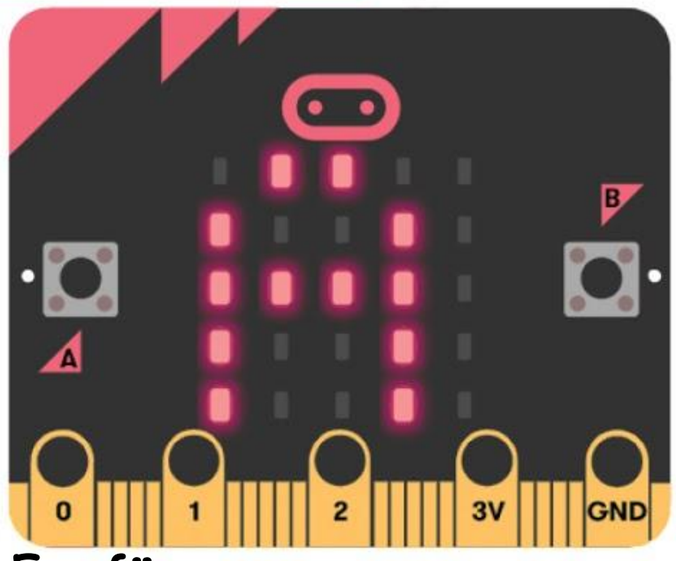
Wenn du nur einen EDU:BIT hast, kannst du die Funkübertragung trotzdem testen. Rufe in deinem Browser makecode.com/multi auf und schreibe deinen Code für den "Sender" und "Empfänger". Du siehst die Ergebnisse in den Simulator-Fenstern.



Klicke auf den micro:bit um ihn in Vollbild zu öffnen.



Sender



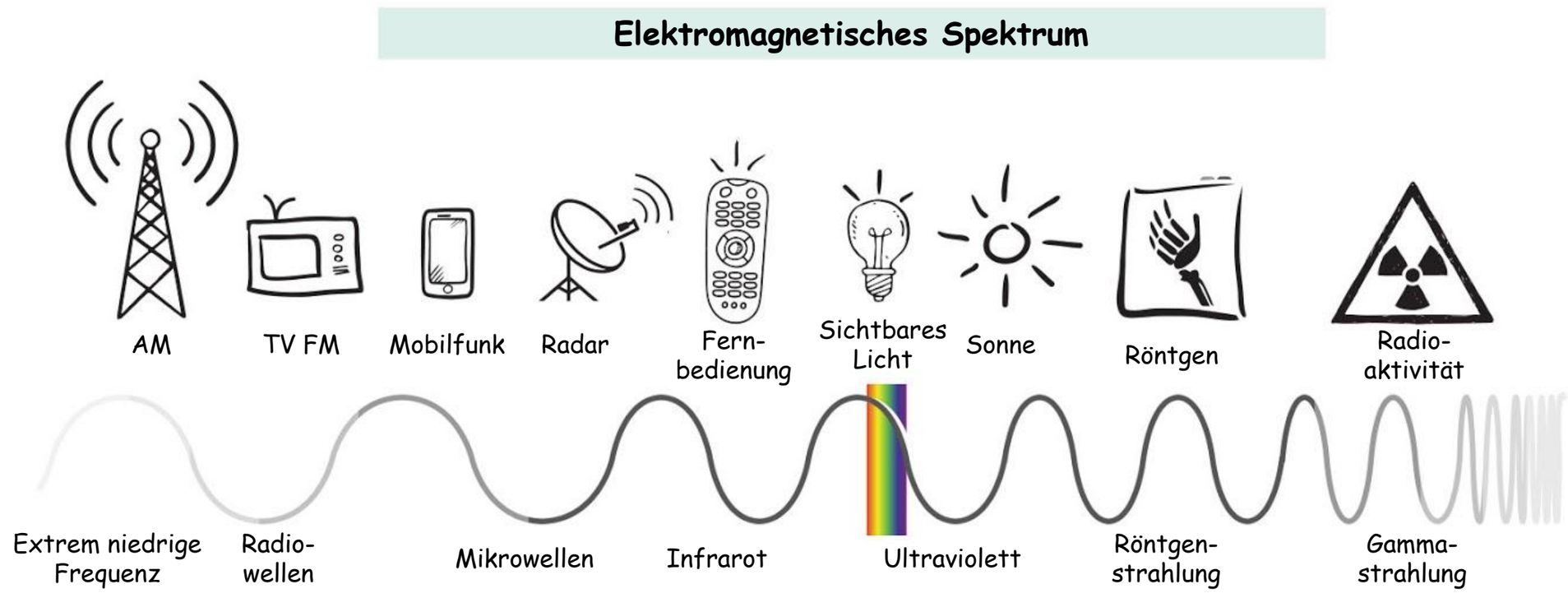
Empfänger

Wenn du Knopf A am "Sender" drückst, wird der "Empfänger" das Funksignal empfangen und den Text anzeigen, z.B. A. Was passiert, wenn du die Knöpfe A+B gleichzeitig drückst?

GUT ZU WISSEN!

Wenn du deinen EDU:BIT umdrehst, kannst du die eingebaute Funk- und Bluetooth-Antenne sehen.

Die Antenne überträgt Signale in Form von elektromagnetischen Wellen, so wie sie auch für Fernseh- und Radioübertragungen, sowie für Satelliten verwendet werden.



MERKE!

Damit dein EDU:BIT Signale von anderen EDU:BITs empfangen kann, müssen alle auf die gleiche Funkgruppe eingestellt werden.

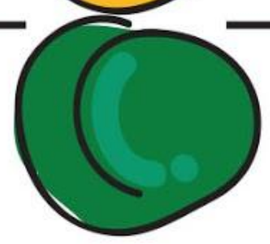
HERAUSFORDERUNG

Programmiere ein Rückmeldungs-Funknetz für deine Klasse.

Wie funktioniert es?

Setze alle EDU:BITs in der Klasse auf dieselbe Funkgruppe.

Der EDU:BIT der Lehrerin oder des Lehrers scrollt Text, wenn er ein Text-Signal empfängt und aktiviert die LEDs auf Ampel Bit, wenn er ein Zahl-Signal empfängt. Die Zahlen sind so zugewiesen:

Zahl	LED	Was heißt es?
1	ROT 	A / Nein / Falsch
2	GELB 	B / Vielleicht / Nicht sicher
3	GRÜN 	C / Ja / Stimmt

Die EDU:BITs der Schülerinnen und Schüler senden einen Text mit ihrem jeweiligen Namen und dann eine Zahl (1, 2 oder 3) um die LEDs auf dem EDU:BIT des Lehrers leuchten zu lassen.

Drücke Knopf A um die Zahl 1 zu senden.
Drücke Knopf B um die Zahl 2 zu senden.
Drücke Knopf A+B um die Zahl 3 zu senden.

Hier ist ein Tipp. Verwendet kurze Kosenamen (mit nur 2 oder 3 Buchstaben), damit ihr nicht so viel Text zu scrollen habt.

Ich habe gelernt...

- [ ] mit EDU:BIT Text und Animationen auf der LED-Matrix anzuzeigen.
- [ ] MakeCode .hex-Dateien herunterzuladen, zu speichern und zu veröffentlichen.
- [ ] Eingabe-Blöcke für eventbasiertes Programmieren zu verwenden.
- [ ] Variablen zu erstellen und zu verwenden.
- [ ] mit dem Piezo-Summer auf dem Music Bit Melodien zu spielen.
- [ ] Funktionen zu erstellen und zu verwenden.
- [ ] Musiknoten zu lesen.
- [ ] mit EDU:BIT die LEDs auf dem Ampel Bit zu steuern - an, aus & umschalten.
- [ ] Erweiterungen zum MakeCode Editor hinzuzufügen.
- [ ] EDU:BIT zu programmieren, mit IR Bit Hindernisse zu erkennen.
- [ ] Während-Schleifen zu verwenden.
- [ ] Arrays / Felder zu verwenden.
- [ ] mit EDU:BIT analoge Eingangssignale zu lesen.
- [ ] analoge Eingangssignal auf einen neuen Wertebereich zu verteilen.
- [ ] mit Logik-Blöcken bedingte Anweisungen zu programmieren.
- [ ] EDU:BIT so zu programmieren, dass er mit Sound Bit Geräusche messen kann.
- [ ] mit Hilfe von Event-Triggern zwischen verschiedenen Modi umzuschalten.
- [ ] mit EDU:BIT die Geschwindigkeit und Drehrichtung eines Gleichstrommotors zu steuern.
- [ ] mit Mathematik-Blöcken Rechenoperationen auszuführen.
- [ ] mit EDU:BIT die Drehposition eines Servomotors zu steuern.
- [ ] EDU:BIT so zu programmieren, dass er die RGB-LEDs auf RGB Bit in verschiedenen Farben und Mustern leuchten lässt.
- [ ] mit EDU:BIT Funksignale zu senden und zu empfangen.

*Hake ein Kästchen ab, wenn du es erlernt hast; sonst sieh dir das zugehörige Kapitel noch einmal an.

NACHRICHT VON EDUTEAM @ CYTRON

GRATULIERE!!!

Du hast alle Kapitel geschafft und gelernt, mit dem MakeCode-Editor Programme zu schreiben. Wir hoffen, du hattest viel Spaß mit den Spielen, die wir für dich ausgesucht haben. Sei stolz auf die Herausforderungen, die du gemeistert hast.

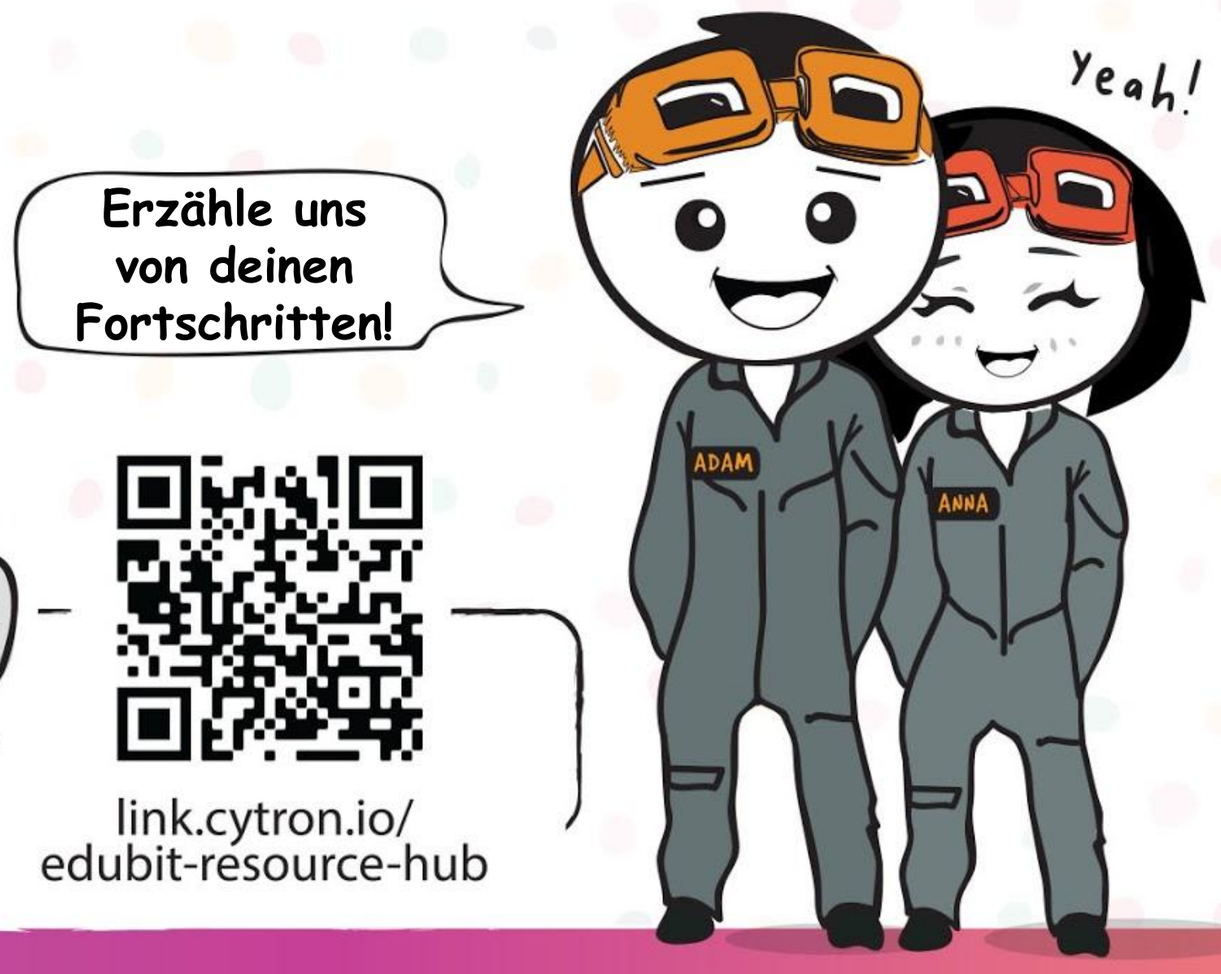
Mittlerweile solltest du ein gutes Verständnis dafür haben, was du alles mit micro:bit und den zusätzlichen Geräten auf deinem EDU:BIT machen kannst. Pssst… wusstest du, dass du jedes Bit herunterbrechen kannst?

Wenn du willst, brich die Bits ruhig herunter. Sobald sie von der Hauptplatine abgebrochen sind, musst du sie mit den mitgelieferten Kabeln anschließen. Mit den einzelnen Bits kannst du dann neue Projekte machen.

Jetzt kannst du kreativ sein und dir viele spannende Spiele und Programme einfallen lassen. Wir können es kaum erwarten zu sehen, welche genialen Projekte du entwickeln wirst.

Wir freuen uns, wenn du uns eine Nachricht mit deinem Projekt schickst. Schreibe uns eine E-Mail oder hinterlasse eine Nachricht auf unserer Facebook-Seite. Es wäre toll, von dir zu hören!

Mach's gut!
Adam & Anna



link.cytron.io/
edubit-resource-hub


Bit..
Bit...