CALLIOPE mini

ARBEITSHEFT 2

Projekte und Übungen für den Unterricht

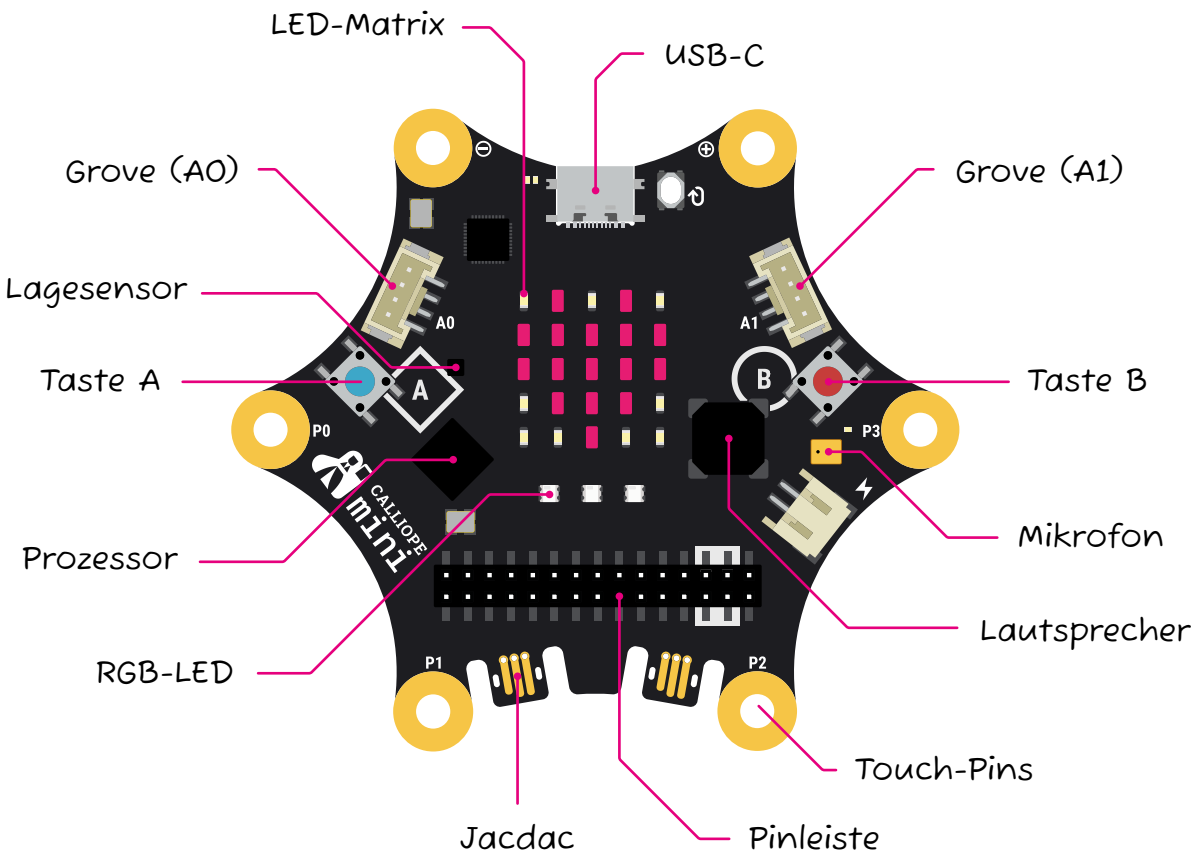
Autorin und Autoren:
Franka Futterlieb
Amando Pascotto
Jørn Alraun

calliope.cc

INHALTSVERZEICHNIS

DER CALLIOPE MINI

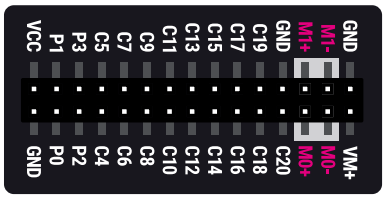
Betrachte deinen Calliope mini. Wenn du genau hinsiehst, kannst du verschiedene Symbole erkennen, die dir Hinweise zu Bauteilen geben.



Der Calliope mini lässt sich ganz einfach erweitern:

Über die **Grove Konnektoren** A0 und A1 lässt sich zum Beispiel ein Ultraschallsensor, ein Feuchtigkeitssensor aber auch das MotionKit anstecken.

Über die **Jacdac Konnektoren** lassen sich durch einfaches Verbinden diverse Module beliebig kombiniert verwenden.



Über die **Pinleiste** können zusätzliche Motoren angeschlossen werden. Auf der Rückseite des Calliope mini findest du die Bezeichnungen der einzelnen Pins.

PROGRAMME PLANEN

Wenn du eine Idee hast, mache einen Plan. Überlege, in welcher Reihenfolge einzelne Schritte ablaufen sollen. Ein Plan hilft dir bei der Umsetzung deiner Idee und ist nützlich, damit du den Überblick bewahrst. Am besten funktioniert es, wenn du bekannte Symbole und Bezeichnungen verwendest. So können auch andere deinen Plan lesen und mitdenken.

Pseudocode

Pseudocode ist eine Möglichkeit, den Ablauf eines Programms in verständlicher Sprache zu beschreiben, ohne dabei eine bestimmte Programmiersprache zu verwenden. Pseudocode ist eine Mischung aus normalem Text und Programmieranweisungen. Unter Verwendung von Schlüsselwörtern wie Start, Ende, wiederhole solange bis, während, wenn/dann und ansonsten, lassen sich die Schritte eines Programms eindeutig und logisch aufschreiben, sodass der Ablauf auch ohne Programmierkenntnisse nachvollziehbar ist. Pseudocode hilft dabei, eine Idee zu strukturieren und den Plan für ein Programm zu erstellen, bevor es in eine Programmiersprache übersetzt wird.

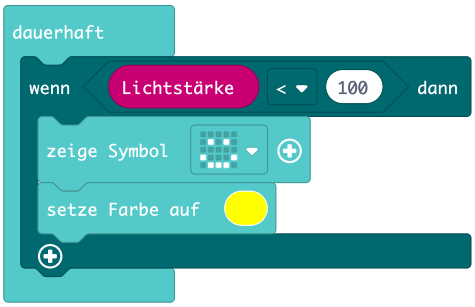
1. Schaue dir das Beispiel an und formuliere einen Pseudocode für das untere Programm.

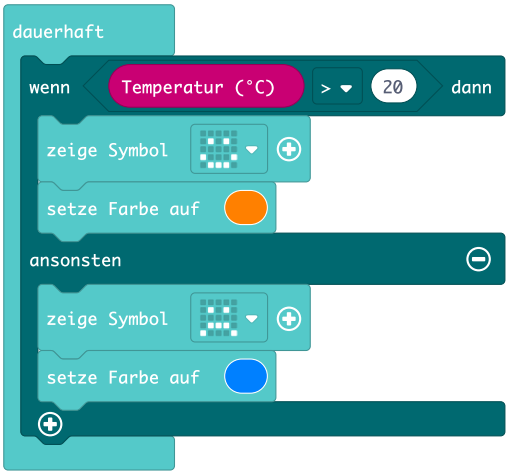
Wenn die Lichtstärke unter 100 liegt,

dann erscheint ein lachender Smiley

auf der LED-Matrix und die

RGB-LED leuchtet gelb.

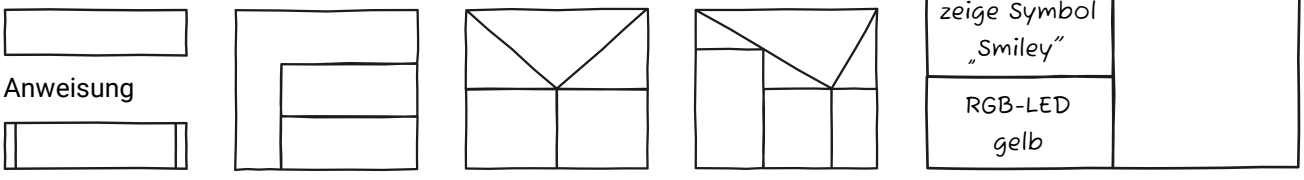




STRUKTOGRAMME

Ein Struktogramm, auch als Nassi-Shneiderman-Diagramm bekannt, ist eine visuelle Methode, um den Ablauf eines Programms darzustellen. Es hilft dabei, die Struktur und Logik eines Programms klar und übersichtlich darzustellen. In einem Struktogramm werden die verschiedenen Schritte und Entscheidungen in rechteckigen Feldern dargestellt, die miteinander verbunden sind. Diese Felder zeigen Anweisungen, Schleifen, Bedingungen und Funktionen. Struktogramme werden von oben nach unten gelesen und kommen daher ohne Pfeile aus. MakeCode verfügt über abgeschlossene Blöcke wie die Startfunktion, Dauerschleifen, Events. Diese Blöcke werden unabhängig und parallel ausgeführt. Im Arbeitsheft werden diese Blöcke sowie Funktionen als schwebende Balken dargestellt.

2 Schaue dir deinen Pseudocode aus Aufgabe 1 an und erstelle ein Struktogramm für diesen Code.

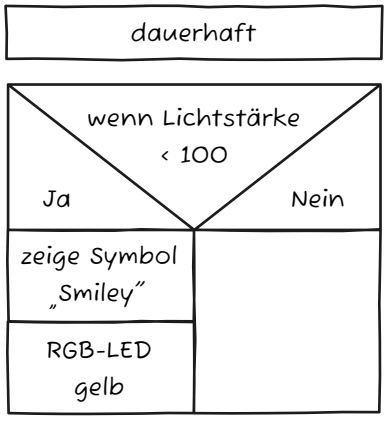


Funktionsaufruf

Schleife

Abfrage (ein-/zweifach)

Abfrage (mehrfach)

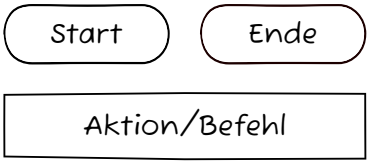


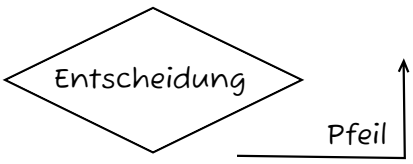
Beispiel

PROGRAMMABLAUFPLÄNE

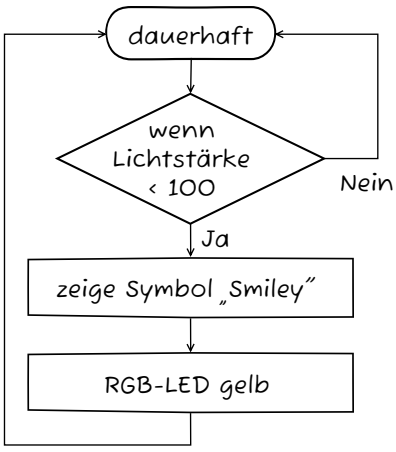
Ein Programmablaufplan (PAP), auch als Flussdiagramm bekannt, ist eine visuelle Methode, um den Ablauf eines Programms darzustellen. Ein PAP besteht aus verschiedenen Symbolen wie Rechtecken, Rauten und Pfeilen, die die Schritte, Entscheidungen und Verbindungen in einem Programm zeigen. Programmablaufpläne helfen dabei, die Logik und Struktur eines Programms leicht verständlich zu machen. Rechtecke stehen für Anweisungen oder Aktionen, Rauten für Entscheidungen, und Pfeile zeigen die Reihenfolge der Abläufe an.

3 Schaue dir deinen Pseudocode aus Aufgabe 1 an und erstelle einen Programmablaufplan für diesen Code.





Elemente



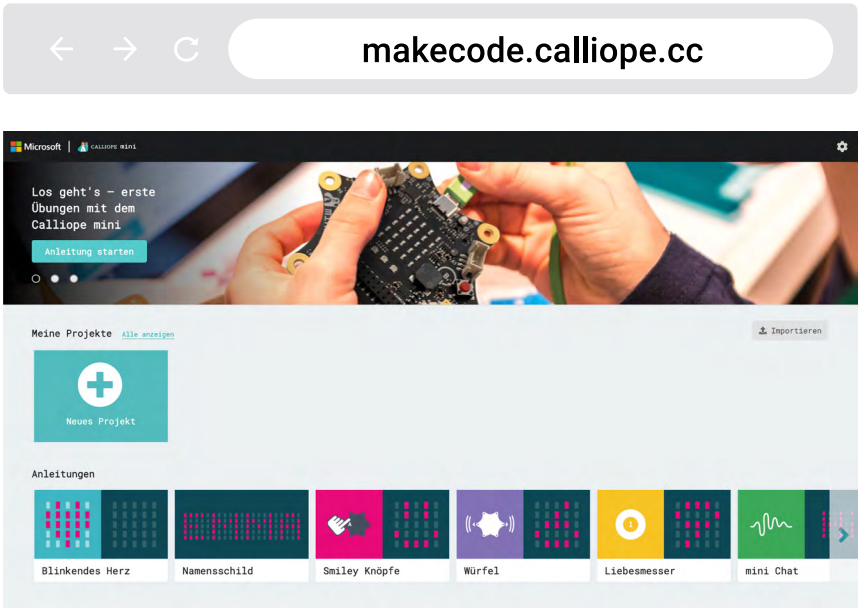
Beispiel

PROGRAMMIERUMGEBUNG

Damit der Calliope mini deine Anweisungen versteht, müssen diese in der passenden Sprache geschrieben werden. Dazu brauchst du eine Programmierumgebung - auch Editor genannt.

Eine passende Programmierumgebung ist der Editor MakeCode.
Du findest ihn unter folgender Internetadresse:

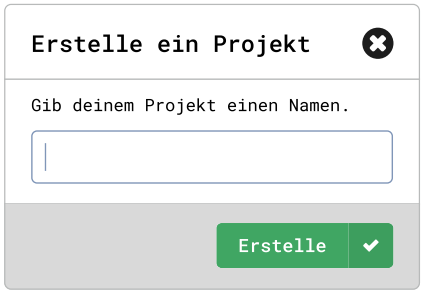
1. Gib die URL in die Browserleiste ein - **makecode.calliope.cc**



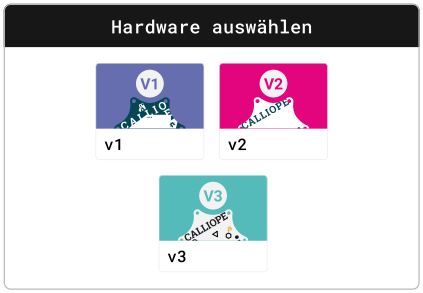
Oder scanne den QR-Code.

2. Starte mit einem neuen Projekt und schau dir die Oberfläche genau an.

Ein neues Projekt starten
Klicke auf die türkise Schaltfläche mit dem großen „+“ Zeichen.



Projekt benennen
Gib einen passenden Namen ein und klicke auf „Erstelle“.



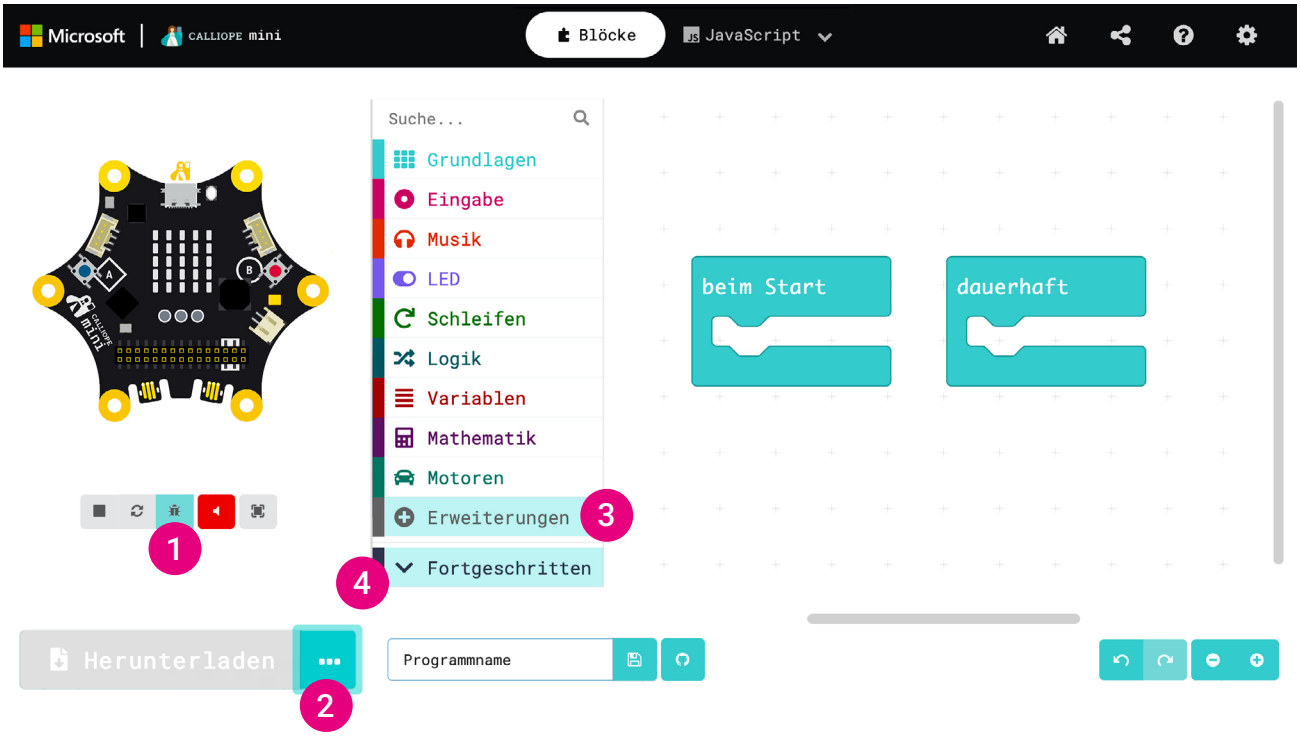
Hardware auswählen
Wähle zu Beginn eines neuen Projekts deine Calliope mini Version aus.

Tipp: Die Version deines Calliope mini erkennst du am Aufdruck auf der Rückseite. Dort steht rechts neben dem USB-Anschluss die Versionsnummer.

MAKECODE

Übersicht

3 Kennst du die verschiedenen Bereiche des Editors und kannst sie den Beschreibungen zuordnen?

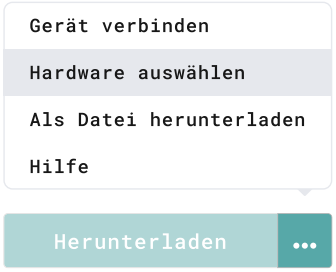


MakeCode bietet die Möglichkeit, Erweiterungen mit zusätzlichen Befehlsblöcken in die Bibliothek hinzuzufügen.

In dem Bereich „Fortgeschritten" findest du weitere nützliche Kategorien, wie z.B. Text oder Spiel mit entsprechenden Befehlsblöcken.

Über die Hardware-Auswahl wählst du deine Calliope mini Version aus, damit dein Programm auf deinem Calliope mini ausgeführt wird.

Mit dem Debugger kann das Programm nach Fehlern überprüft werden. Im Debugger werden die aktuellen Werte von allen Variablen im Programm angezeigt.



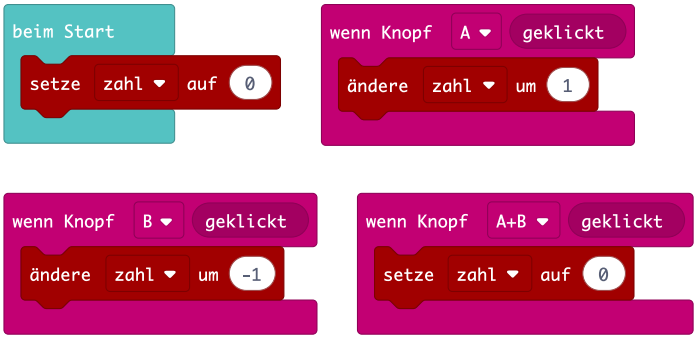
Hinweis: Stelle sicher, dass du deine Calliope mini Version im Simulator siehst. Du kannst jederzeit die Version anpassen. Klicke dazu auf die drei Punkte neben „Herunterladen“ und drücke „Hardware auswählen“.

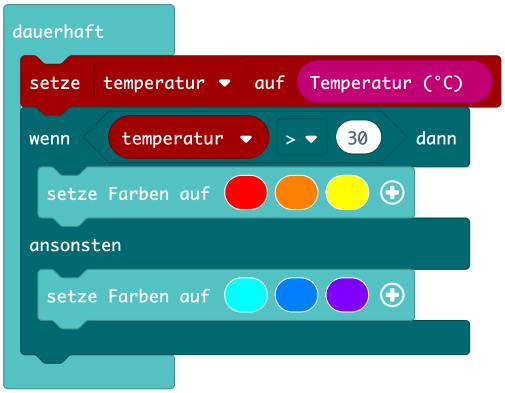
DEBUGGEN

Fehler finden im Code mit Hilfe des Debuggers.

1. Überprüfe im Debugger. Welche Funktionen haben die einzelnen Schaltflächen? Teste sie im Editor und notiere die Eigenschaften.

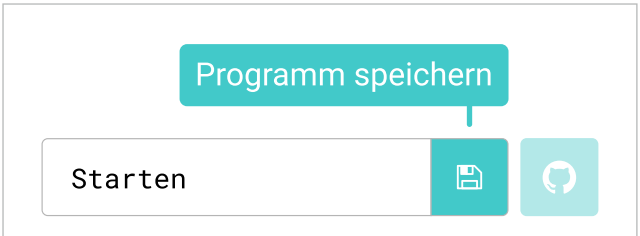
2. Erstelle dazu folgende Programme in MakeCode.
Starte den Debug-Modus und beschreibe, was passiert.



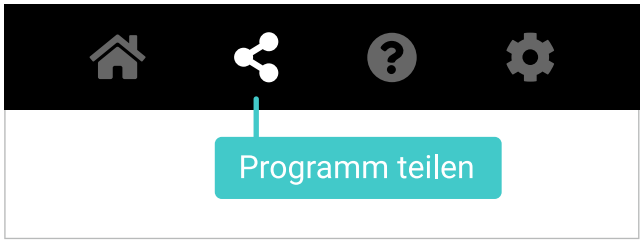


PROGRAMME VERWALTEN

Wie kann ich Programme speichern?

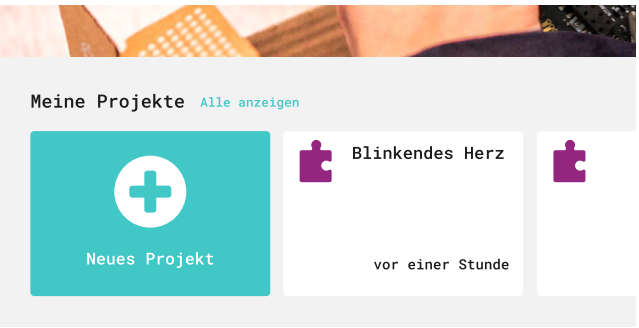


Durch einen Klick auf das **Speichern-Symbol** kannst du dein Programm als .hex Datei speichern. Die .hex Datei lässt sich wieder in MakeCode öffnen und bearbeiten.

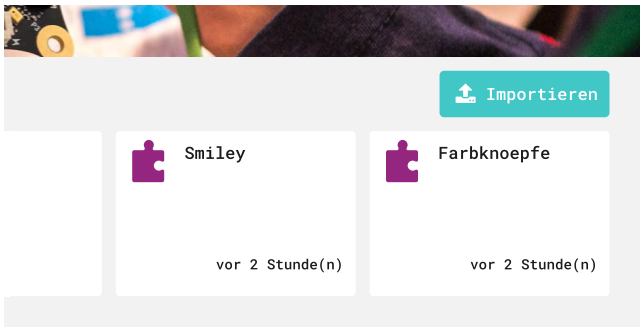


Über das **Teilen-Symbol** wird eine URL und ein QR-Code für dein Programm generiert. Du kannst diesen Link speichern oder mit anderen teilen.

Wie kann ich ein Programm öffnen?



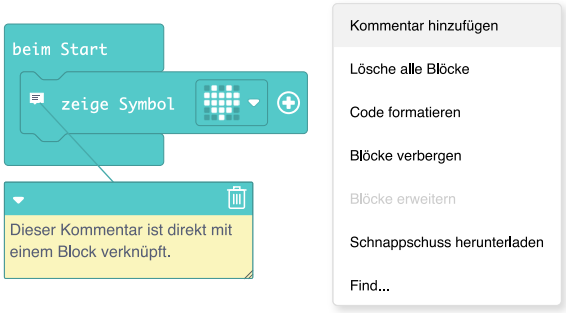
Deine bearbeiteten Projekte findest du auf der Startseite unter **„Meine Projekte“**.



Über **„Importieren“** kannst du gespeicherte Projekte öffnen.

Programme kommentieren

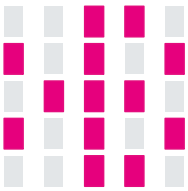
Kommentare helfen, das Programm zu dokumentieren, sodass du oder auch andere Personen den Ablauf nachvollziehen können. Kommentare können direkt an die Blöcke geheftet werden oder frei auf der Arbeitsfläche als Post-Its platziert werden.



MOBIL PROGRAMMIEREN

Der Calliope mini kann mobil per Bluetooth programmiert werden. Dazu wird die Calliope mini App benötigt. Um die Datei nach dem Programmieren zu übertragen, muss der Calliope mini in den Bluetooth-Modus versetzt und mit dem mobilen Gerät verbunden sein.

Calliope mini Bluetooth-Modus

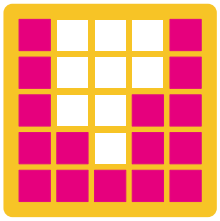


- Halte **Tasten A+B** gedrückt. Drücke zusätzlich die **Reset Taste** für 1 Sekunde. Halte die Tasten A+B so lange gedrückt, bis die Bluetooth-Animation beendet ist.

> **Tipp:** Ab dem Calliope mini 3 kannst du den Bluetooth-Modus starten, indem du 3x die Reset Taste drückst.

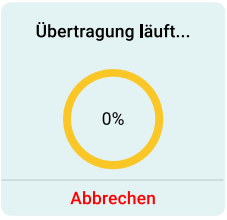
- Das individuelle **ID-Muster** erscheint auf deinem Calliope mini.

Verbindung mit einem mobilen Gerät



- Über das **rote Calliope mini Icon** im Bereich „Editoren und Programme“ öffnest du das Verbindungsfenster.
- Übertrage das individuelle ID-Muster des Calliope mini auf die Matrix.
- Wird der Calliope mini gefunden, verbindet er sich automatisch mit der App und ein Smiley erscheint.

Übertragung



- Jetzt kannst du Programme übertragen.
- Verwende den Editor MakeCode, um ein Programm zu erstellen.
- Tippe auf Herunterladen, um das Programm zu übertragen.
- Nach Abschluss erscheint das Programm auf dem Calliope mini!

Weitere Tipps zur Nutzung mit mobilen Geräten findest du unter:
https://calliope.cc/programmieren/mobil

SPIELE MIT DEM CALLIOPE MINI

Lerne Spielkonzepte kennen und entwickle eigene kleine Spiele.

SCHERE STEIN PAPIER

Zwei Personen spielen Schere, Stein, Papier gegeneinander. Pro gewonnener Runde gibt es einen Punkt. Wer zuerst 3 Punkte hat, hat gewonnen.

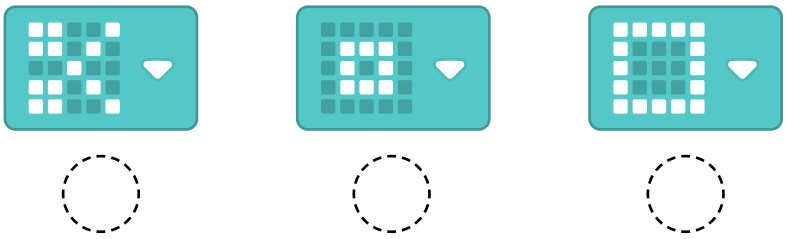
Spielablauf:
- 2 Personen spielen jeweils mit einem Calliope mini.
- Beide Calliope mini werden gleichzeitig geschüttelt.
- Per Zufall wird auf beiden Calliope mini eines von drei Zeichen angezeigt: Schere, Stein oder Papier.
- Die Punkte werden manuell über den Knopf A vergeben.

Regeln:
Schere schlägt Papier und verliert gegen Stein.
Stein schlägt Schere und verliert gegen Papier.
Papier schlägt Stein und verliert gegen Schere.

Eingabe: Lagesensor schütteln und Knopf A.
Verarbeitung: Zufällige Anzeige eines Zeichens und Punktevergabe.
Ausgabe: Die LED-Matrix, die RGB-LED und der Lautsprecher.

Zufällige Anzeige der Zeichen:

1 Weise jedem Zeichen eine Zahl von 1 - 3 zu.

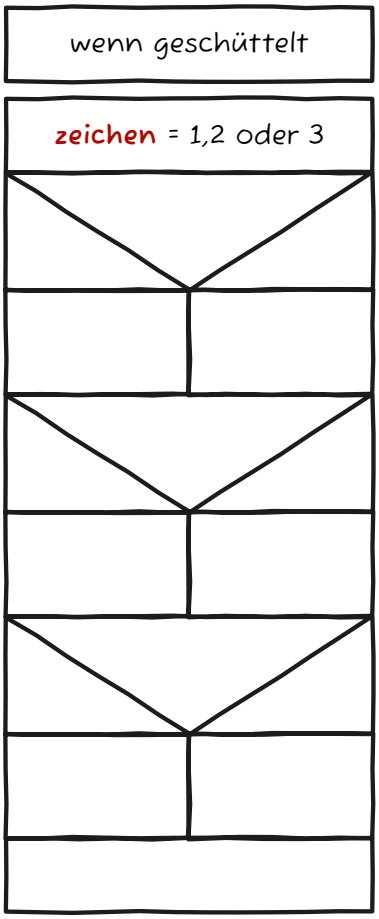


2 Ergänze das Struktogramm:

- zufall = 1
- zufall = 2
- zufall = 3

- zeige Symbol „Schere“
- zeige Symbol „Stein“
- zeige Symbol „Papier“

- Pausiere



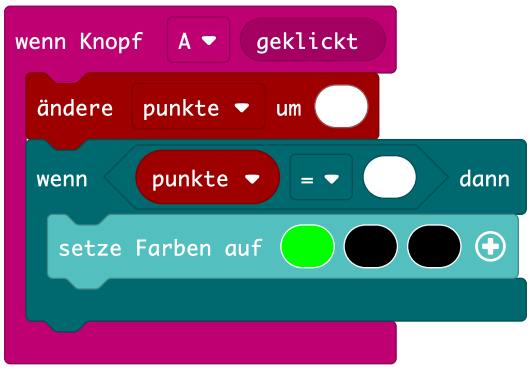
3 **Gehe Schritt für Schritt vor:**
a) Erstelle eine Variable mit dem Namen ***zeichen***.
b) Programmiere anhand deines Struktogramms.

Tipp: Füge eine Pause hinzu, um ein erneutes Schütteln pro Runde zu verhindern.

c) Überprüfe im Debugging Modus, ob alle Zahlen vorkommen.

Punktevergabe:
Die Person, die die Runde gewonnen hat, drückt den Knopf A und die Punkte werden um einen Punkt erhöht.
Pro Punkt leuchtet eine RGB-LED grün.
Sind drei Punkte erreicht, ertönt ein akustisches Signal.

4 Schau dir den Programmcode an und ergänze die fehlenden Angaben.



5 Übertrage das Programm in MakeCode und vervollständige die Abfragen für die möglichen Punktestände.

Tipp: Vergiss nicht, die Variable ***punkte*** zu definieren.

6 Diskutiert, welche Vor- und Nachteile die manuelle Vergabe der Punkte hat.

REAKTIONSSPIEL

Entdeckt den Punkt so schnell wie möglich, wenn er links oder rechts erscheint.

Spielablauf:
- Per Zufall leuchtet links oder rechts eine LED auf der LED-Matrix des Calliope mini auf.
- Mit den Knöpfen A und B wird die Seite markiert, auf der der Punkt entdeckt wurde
- Wurden 3 Versuche verspielt, ist das Spiel zu Ende.

Eingabe: Knöpfe A und B.
Verarbeitung: Zufällige Anzeige und Auswertung.
Ausgabe: Die LED-Matrix, die RGB-LED und der Lautsprecher.

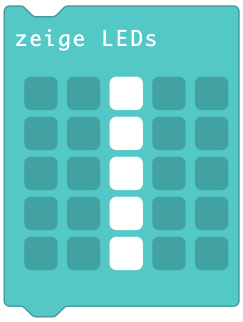
Entwickle das Programm in thematischer Reihenfolge:

1. Auftauchen des Punkts
2. Steuerung der Knöpfe
3. Auswertung

---

1 Auftauchen des Punkts

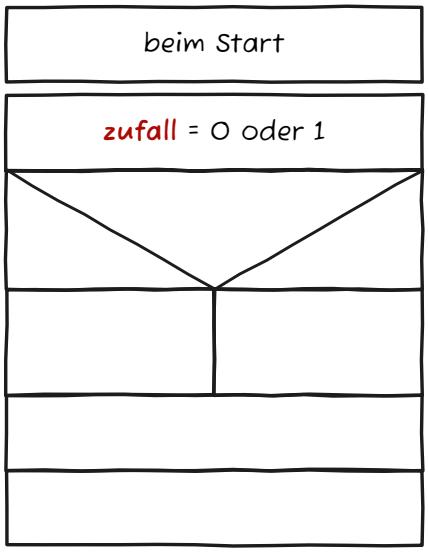
a) Lasse eine Mittellinie auf dem LED-Display leuchten.

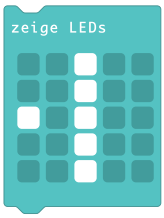


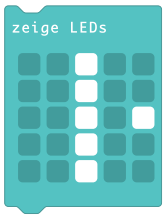
b) Jetzt kommt der Zufall dazu! Lasse dir eine zufällige Zahl aus einem 2er Zahlenraum ausgeben.

c) Über eine Verzweigung mit folgenden Bedingungen wird die Position des Punkts festgelegt.

Programmiere anhand des Struktogramms:
Wenn die zufällige Zahl eine 0 ist, lasse den Punkt links erscheinen.
Wenn die zufällige Zahl eine 1 ist, lasse den Punkt rechts erscheinen.





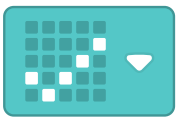


d) Lasse den Punkt nach 1 Sekunde wieder verschwinden.

e) Fasse die Anzeige-Mechanik in einer Funktion zusammen und rufe diese vorerst aus der Start-Funktion auf.

Hinweis: Funktionen sind eine zusammengefasste Einheit innerhalb eines Programms. Funktionen bestehen aus einer Reihe von Anweisungen, die ausgeführt werden, wenn die Funktion aufgerufen wird. Funktionen helfen dabei, den Code übersichtlicher zu machen. Anstatt denselben Code mehrmals zu schreiben, kannst du einfach die Funktion beliebig oft aufrufen. (Funktionen können auch Eingabewerte (Parameter) erhalten und Ausgabewerte (Rückgabewerte) liefern. So kannst du komplexe Programme in kleinere, gut verständliche Teile zerlegen.)

2 Steuerung der Knöpfe



a) Programmiere die beiden Knöpfe im ersten Schritt so, dass bei jedem Drücken ein Haken angezeigt wird.

b) Überprüfe im zweiten Schritt, ob der Punkt rechtzeitig entdeckt wurde, indem du abfragst, ob der richtige Knopf gedrückt wurde.

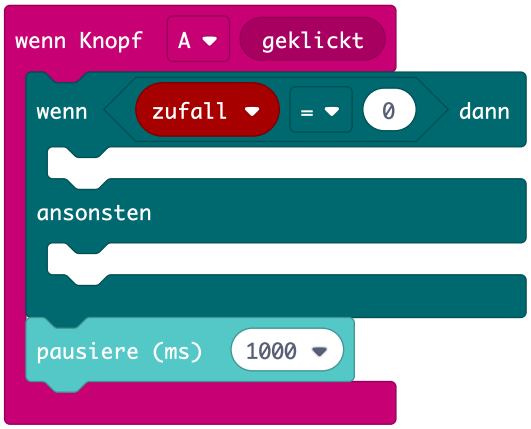
c) Für beide Knöpfe benötigst du die gleiche Logik: Vervollständige den folgenden Satz:

Wenn **zufall** = _____ ist, befindet sich der Punkt auf der _____ Seite und der Knopf A ist der richtige, ansonsten ist Knopf A der falsche.

d) Übertrage die Logik auf den Code auf der rechten Seite.

e) Erweitere dein Programm, sodass, wenn diese Bedingung erfüllt wird, ein Haken erscheint, wenn nicht ein X angezeigt wird und du weißt, dass das nicht der richtige Knopf war.

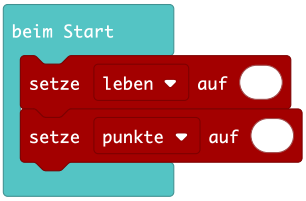
f) Wende die Logik von Knopf A auf Knopf B an.



3 Auswertung

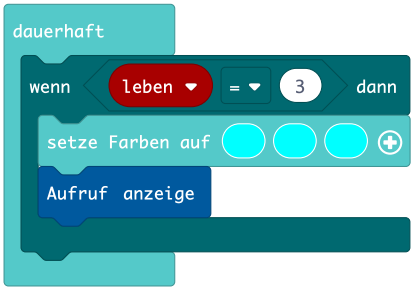
Für die Auswertung werden alle richtigen Versuche (***punkte***) und die falschen Versuche (***leben***) gezählt und in Variablen gespeichert.

a) Starte dein Programm mit den Starteinstellungen für die Anzahl der Punkte und der Leben sowie einem Countdown, bevor es richtig losgeht.

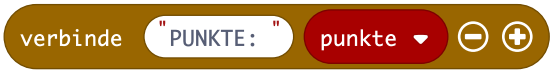


b) Erweitere die Funktionen für den Knopf A und den Knopf B um die Punktevergabe.

c) Programmiere eine Auswertung, die die Anzahl der verbleibenden Versuche dauerhaft mit den RGB-LED anzeigt.

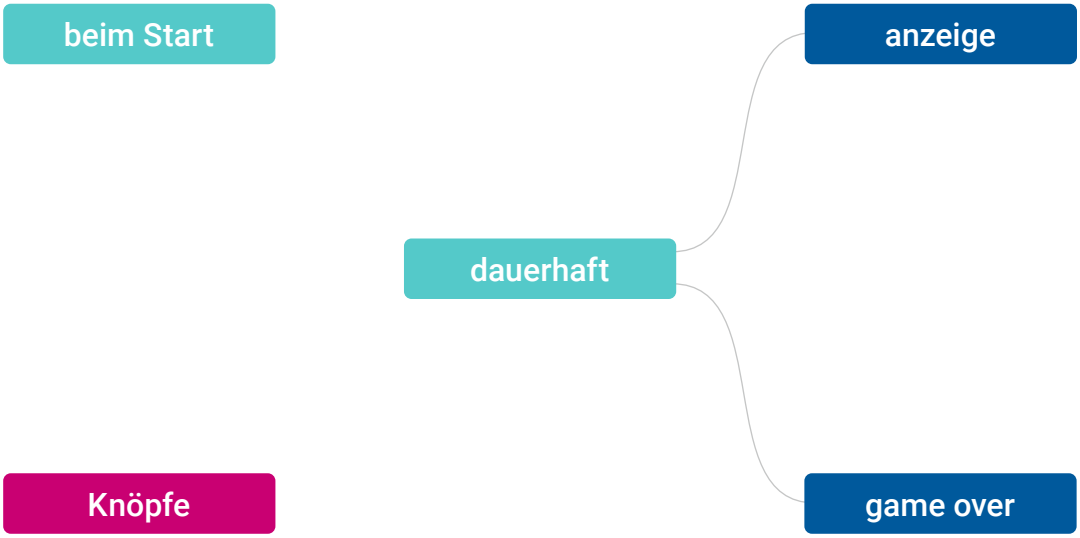


d) Schreibe eine weitere Funktion (**game over**) für den Spielabschluss. Verbinde dazu den Text „GAME OVER“ mit der Anzeige der Anzahl der richtigen Versuche.



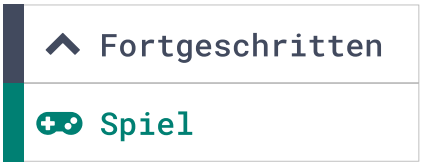
4 Programmfunktionen

Hier findest du eine Übersicht der gesamten Programmfunktionen. Ergänze in Stichpunkten, welche Aufgaben die 5 Funktionen haben.



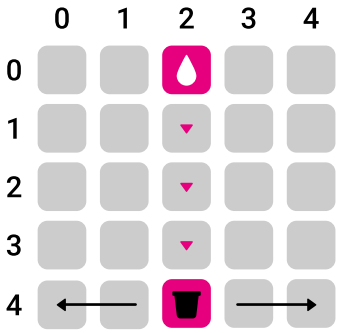
REGENTROPFEN FANGEN

Lerne die Codeblöcke in der Kategorie Spiel unter Fortgeschritten kennen.



Spielablauf:

- Regentropfen fallen von oben nach unten.
- Mit einem Eimer werden die Regentropfen gefangen.
- Wird ein Regentropfen gefangen, gibt es einen Punkt.
- Wird der Regentropfen verfehlt, wird ein Leben abgezogen.
- Sind alle Leben aufgebraucht, ist das Spiel zu Ende und die Punkte werden angezeigt.



Eingabe: Knöpfe A und B.
Verarbeitung: Steuerung, Abfrage und Auswertung.
Ausgabe: Die LED-Matrix und die RGB-LED.

Wichtige Blöcke:

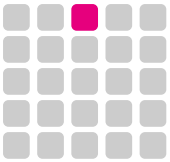
Hinweis: Ein Sprite (englisch „Kobold“) ist wie eine kleine Figur in Form einer LED, der du sagen kannst, was sie tun soll. Du kannst ihr sagen, dass sie sich bewegen und drehen soll. Du kannst abfragen, an welcher Position sie sich befindet und sogar prüfen , ob sie einen anderen Sprite berührt.
Damit ein Sprite angesprochen werden kann, benötigt es einen Namen. Dieser Name wird über eine Variable vergeben.

Entwickle das Programm in thematischer Reihenfolge:

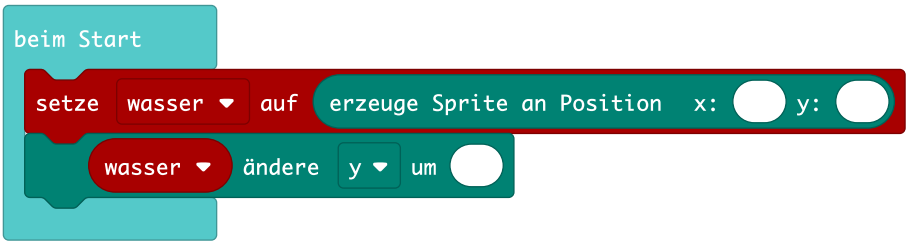
1. Regentropfen Steuerung
2. Eimer Steuerung
3. Starteinstellungen
4. Kollisionsabfrage und Auswertung

1 Regentropfen Steuerung

a) Erzeuge ein Sprite, positioniere es mittig ganz oben in der ersten Reihe der LED-Matrix und weise es einer Variablen mit dem Namen ***wasser*** zu.



b) Vervollständige den Code und teste im Simulator.

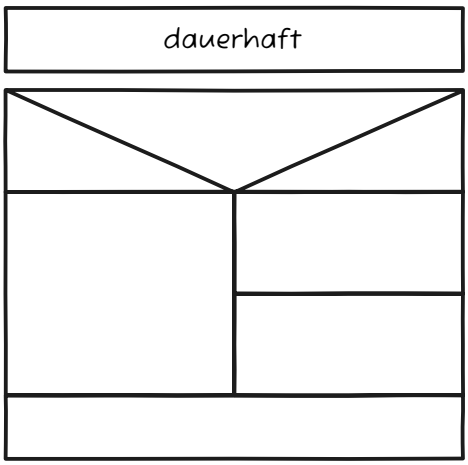


c) Beschreibe was passiert.

d) Vervollständige das Struktogramm anhand des Pseudocodes:

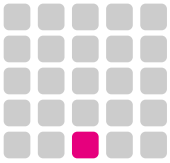
Prüfe dauerhaft:
Wenn die aktuelle y-Position kleiner als 4 ist, dann wird der Regentropfen 1 Position nach unten bewegt.
Ansonsten wird die y-Position wieder auf 0 gesetzt und eine zufällige Position auf der x-Achse definiert.

Füge eine Pause hinzu, um das Tempo einzustellen.



e) Programmiere anhand des Struktogramms.

2 Eimer Steuerung



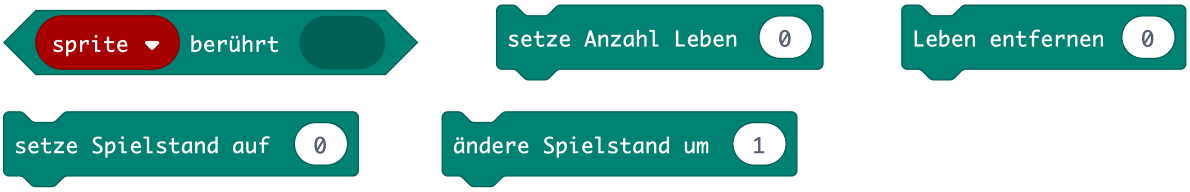
a) Erstelle einen Sprite mit dem Namen Eimer und positioniere ihn mittig ganz unten in der untersten Reihe der LED-Matrix.

b) Programmiere den Knopf A so, dass der Eimer sich nach links bewegt.

c) Programmiere den Knopf B so, dass der Eimer sich nach rechts bewegt.

3 Starteinstellungen

Vervollständige den Spielstart:



a) Setze den Spielstand auf 0.

b) Setze die Anzahl Leben auf 5.

4 Kollisionsabfrage und Auswertung

Erweitere dein Programm um folgende Logik:
Wenn sich die Sprites Eimer und Wasser berühren, wird der Spielstand um einen Punkt erhöht und die RGB-LED leuchten grün, ansonsten werden die Leben um einen Punkt reduziert und die RGB-LED leuchten rot.

Extra

Was würde das Spiel einfacher oder schwieriger machen?
An einer Stelle im Programmcode kann das Spiel über eine Zahl in der Schwierigkeit verändert werden. Weißt du, an welcher Stelle?

DEIN DIGITALES HAUSTIER

Essen, spielen, schlafen - was macht dein digitales Haustier glücklich?

DIGITALES HAUSTIER

Bedürfnisse

1 Haustiere haben Bedürfnisse, die erfüllt werden müssen, damit sie glücklich sind. Welche fallen dir ein? Schreibe sie auf:

2 Wie kannst du herausfinden, ob es deinem Haustier gut geht?
Schreibe deine Ideen auf.

a) Wie kannst du feststellen, ob die Umgebungstemperatur die richtige ist?

b) Wie kannst du feststellen, ob dein Haustier genug gestreichelt wurde?

c) Woher weißt du, ob es Hunger hat?

d) Wie findest du heraus, ob es genug Bewegung hatte?

e) Hat es heute schon etwas von einem Freund oder einer Freundin gehört?

f) Hatte es schon richtig Spaß beim Spielen heute?

g) Hat es genug Schlaf bekommen?

3 Welche Eingaben fallen dir ein, wie du mit deinem Haustier interagieren kannst?

4 Welche Ausgaben fallen dir ein, wie dein Haustier mit dir kommunizieren kann?

Digitales Haustier - Programmieren
In den folgenden Aufgaben entwickelst du Programme zu den einzelnen Bedürfnissen:

1. Temperatur
2. Streicheln
3. Bewegung
4. Spielen
5. Schlafen
6. Hunger
7. Soziale Kontakte und Freundschaften
8. Stimmung

Anschließend werden die einzelnen Bedürfnisse kombiniert und zeigen die Stimmung deines digitalen Haustiers an.

1 Temperatur

Die Temperatur kann durch den Temperatursensor des Calliope mini gemessen werden. Die richtige Temperatur ist wichtig, damit sich Haustiere wohlfühlen und gesund bleiben.

Eingabe: Temperatur und Pin P0
Verarbeitung: Messung der Temperatur und Vergleich mit Schwellenwerten
Ausgabe: LED-Matrix

Gehe Schritt für Schritt vor:

a) Berühre den Pin P0, um herauszufinden, ob es zu kalt oder zu warm ist. Lasse dir ein „OK“ bei einer Temperatur über 20° ausgeben und ein „Kalt“ bei Temperaturen darunter. Gib eine Warnung aus, wenn die Temperatur über 32° ist.

2 Streicheln

Dein Haustier liebt es, gestreichelt zu werden. Berühre den Pin P1, um es zu streicheln.
Wird dein Haustier gestreichelt, blinkt ein Herz auf der LED-Matrix.
Je mehr es gestreichelt wird, desto schneller ist die Blinkfrequenz.
Wurde es genug gestreichelt, erscheint das Herz permanent auf der LED-Matrix.
Berührst du es an Pin P2, wird es ärgerlich und die Blinkfrequenz verlangsamt sich.

Eingabe: Pin P1 und Pin P2
Verarbeitung: Vergleich mit Schwellenwerten und Berechnung einer Pausenlänge
Ausgabe: LED-Matrix

Gehe Schritt für Schritt vor:

a) Erstelle eine Variable mit dem Namen ***streicheln***.

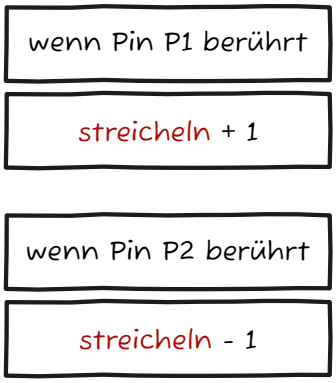
b) Verändere den Wert der Variable ***streicheln*** anhand des Struktogramms.

c) Überprüfe im Debugger.

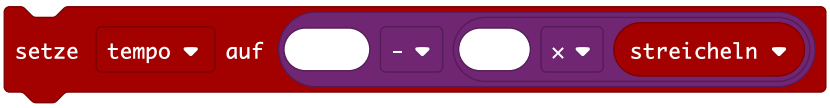
d) Erstelle eine dauerhaft blinkende Herz-Animation.

e) Füge Pausen hinzu.

f) Erstelle eine weitere Variable mit dem Namen ***tempo*** und ersetze die Angaben der Pausen durch die Variable ***tempo***.



g) Das Tempo verändert sich. Schaue in die Tabelle und stelle eine Gleichung auf:



streicheln	tempo
0	1000
1	800
2	600
3	400
4	200
5	0

h) Entferne das Blinken, sobald das Tempo kleiner oder gleich 0 ist.

3 Bewegung

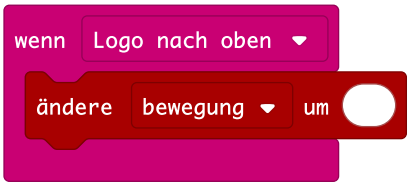
Lasse dir über den Pin P3 anzeigen, ob dein Haustier genug Bewegung hatte. Trainiere mit deinem Haustier. Rolle es nach links, nach rechts, lasse es einen Kopfstand machen oder lasse es sich kerzengerade hinstellen. Sind 5 Übungen gemacht, ertönt eine fröhliche Melodie.

Eingabe: Lagesensor und Pin P3
Verarbeitung: Erkennung verschiedener Gesten mit dem Lagesensor
Ausgabe: LED-Matrix und Lautsprecher

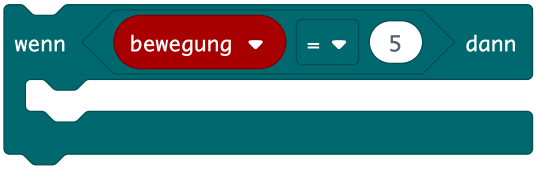
Gehe Schritt für Schritt vor:

a) Erstelle eine Variable mit dem Namen ***bewegung***.

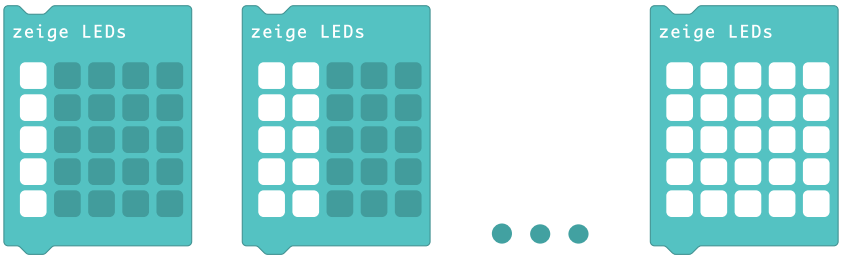
b) Erhöhe die Variable ***bewegung*** um 1, wenn eine Bewegung gemacht wird.



c) Spiele eine Melodie ab, wenn dein Haustier 5 mal bewegt wurde.



d) Visualisiere das Bewegungslevel auf der LED-Matrix, wenn **Pin P3** berührt wurde.



4 Spielen

Lasse dir über das Touch-Logo ausgeben, ob dein Haustier heute schon genug gespielt hat. Wirf dein Haustier in die Luft und fang es wieder auf. Vor Freude fängt es an zu quietschen.

Eingabe: Lagesensor und Touch-Logo Pin
Verarbeitung: Erkennung von Würfen mit dem Lagesensor
Ausgabe: LED-Matrix und Lautsprecher

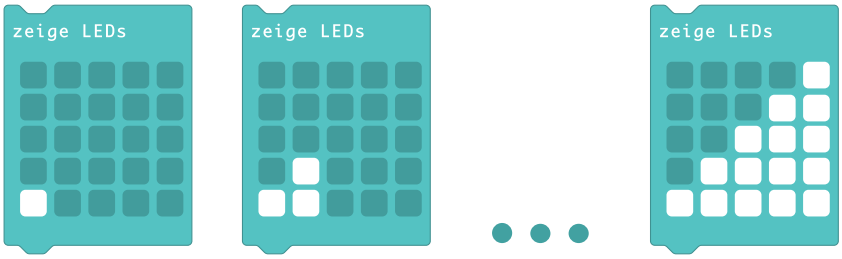
Gehe Schritt für Schritt vor:

a) Erstelle eine Variable mit dem Namen ***werfen***.

b) Erhöhe den Wert der Variable ***werfen*** für jeden Wurf um 1.

c) Begleite jeden Wurf mit einem Ton.

d) Visualisiere das Bewegungslevel auf der LED-Matrix, wenn der **Logo-Pin** berührt wurde. Verwende eine Art Treppe als Darstellung.



5 Schlafen

Steuere die Anzeige der Müdigkeit deines Haustiers über die Helligkeit der LED-Matrix. Lege dein Haustier für eine Zeit in einem abgedunkelten Raum und lasse es ausruhen. Nach einer kurzen Weile ist es wieder hellwach.
Lasse einen Regenbogen erscheinen, wenn es ausgeschlafen ist.

Eingabe: Lichtsensor
Verarbeitung: Messung der Lichtstärke und Vergleich mit Schwellenwerten
Ausgabe: LED-Matrix

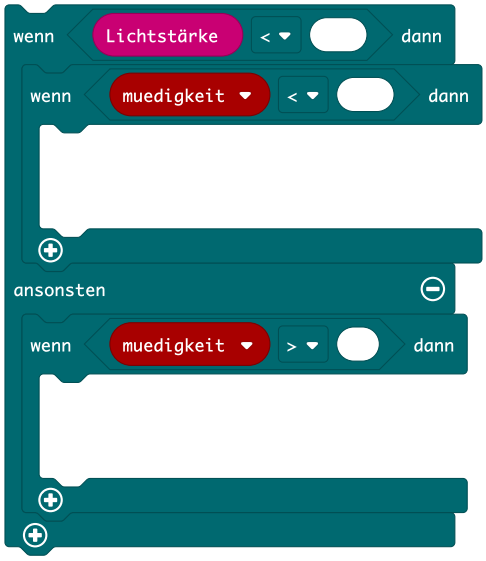
Gehe Schritt für Schritt vor:

a) Schalte alle LEDs der LED-Matrix an.

b) Erstelle eine Variable mit dem Namen ***muedigkeit*** und setze sie auf 0. Die Variable soll später Werte zwischen 0 (ganz müde) und 255 (hellwach) annehmen.

c) Erstelle eine Abfrage, die dauerhaft prüft, ob dein Haustier sich an einem dunklen Ort befindet.

d) Wenn es dunkel ist (**Lichtstärke unter 100**), ändere die Variable ***muedigkeit*** um 25 und warte für eine Sekunde.



e) Überprüfe im Debugger.

f) Ansonsten ändere die Variable ***muedigkeit*** um -25 und warte für eine Sekunde.

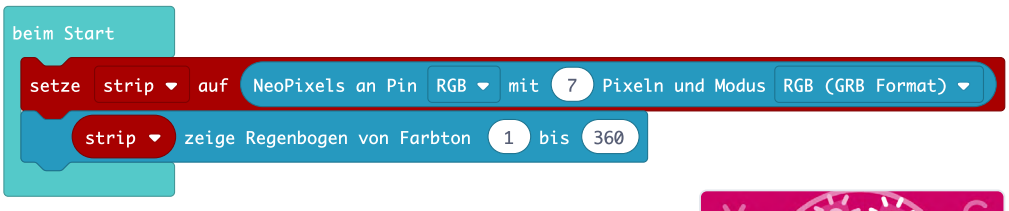
g) Erweitere deine Abfrage, so dass die Variable ***muedigkeit*** im Bereich von 0-255 bleibt.

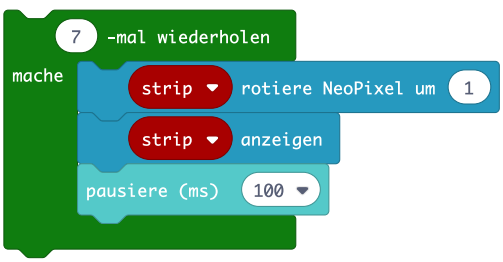
h) Verwende den Wert der ***muedigkeit*** für die Helligkeit der LEDs der LED-Matrix.

i) Lasse die RGB-LED grün leuchten, wenn dein Haustier ausgeschlafen ist (***muedigkeit*** > 200).

Extra

Du kannst anstelle der grünen RGB-LED auch einen Regenbogen Effekt programmieren. Benutze dafür die Neopixel Erweiterung.





6 Hunger

Drücke die Knöpfe A+B gleichzeitig, um festzustellen, ob dein Haustier hungrig ist. Wenn der Futternapf leer ist, dann ist die Matrix leer. Drücke die Knöpfe A+B gleichzeitig, um den Futternapf zu füllen. Ist die LED-Matrix gefüllt, ist dein Haustier gesättigt.

Eingabe: Taste A und Taste B
Verarbeitung: Vergleich mit Schwellenwerten
Ausgabe: LED-Matrix

Gehe Schritt für Schritt vor:

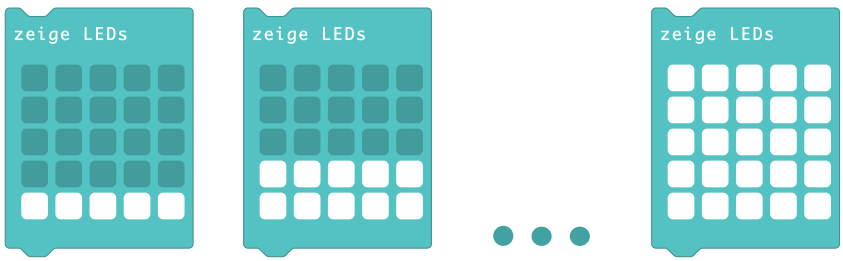
a) Erstelle zwei Variablen mit den Namen ***futter*** und ***anzeige***.

> Über die Variable anzeige wird unterschieden, ob der Futterstand angezeigt oder erhöht wird.
> ***anzeige*** = 0 : Anzeige Futterstand
> ***anzeige*** = 1 : Erhöhung Futterstand

b) Wenn die Tasten A und B zum ersten Mal gleichzeitig gedrückt werden, zeige den aktuellen Futterstand an.

c) Werden die Tasten A und B erneut gleichzeitig gedrückt, füllt sich der Futterstand um 1. Die Variable ***futter*** kann Werte von 1 bis 5 annehmen.

d) Visualisiere die Variable ***futter*** auf der LED-Matrix.



7 Soziale Kontakte und Freundschaften

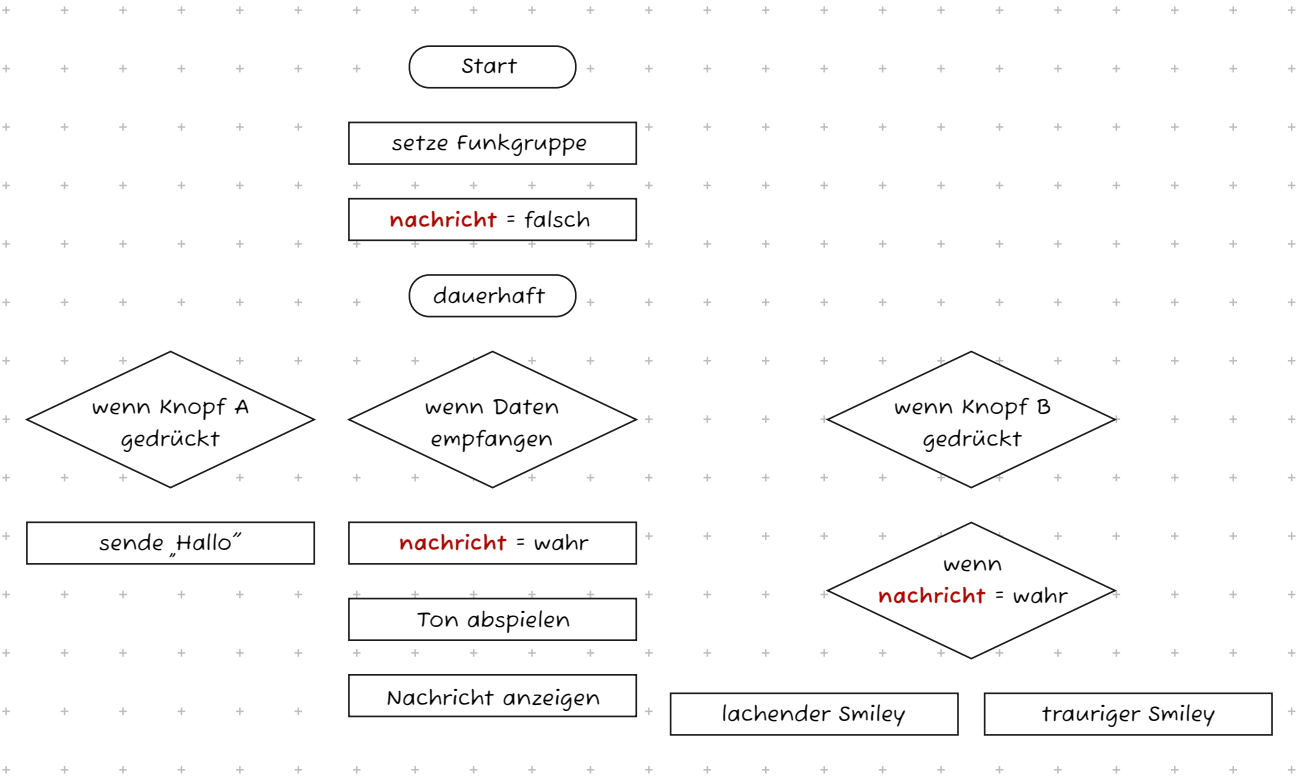
Drücke Knopf B, um festzustellen, ob dein Haustier einsam ist oder bereits eine Nachricht von einem anderen Haustier bekommen hat.
Dein Haustier schickt eine Nachricht an einen Freund oder eine Freundin.
Bekommt es eine Nachricht zurück, ertönt eine Melodie.

Wichtige Blöcke:

Eingabe: Taste B
Verarbeitung: Funknachrichten versenden und empfangen
Ausgabe: LED-Matrix

Gehe Schritt für Schritt vor.

a) Füge die Richtungspfeile in den Programmablaufplan ein.



b) Erstelle dein Programm anhand des Programmablaufplans und teste im Simulator.

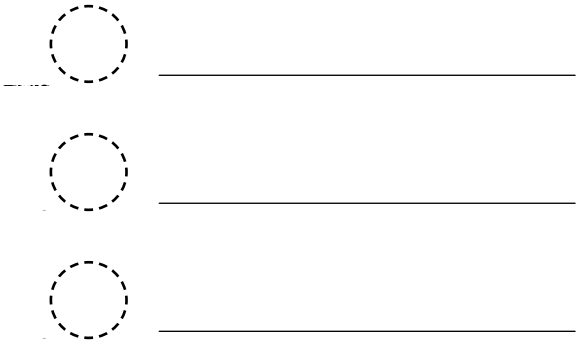
8 Stimmung

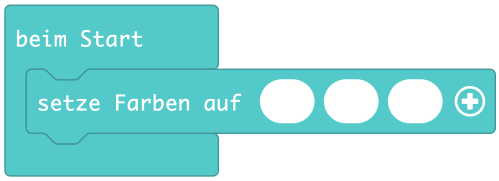
Wie bei einem Stimmungsring sagt dir dein Haustier über die Farbe, wie es ihm geht. Es begrüßt dich gut oder schlecht gelaunt. Finde heraus, was nicht stimmt.

Eingabe: keine
Verarbeitung: Vergleich einer Variable mit Schwellenwerten
Ausgabe: RGB-LED.

Gehe Schritt für Schritt vor:

a) Über die Farben sagt dir dein Haustier, wie es gelaunt ist.
Fülle die Kreise farbig aus und schreibe eine Legende für die Auswertung.





b) Programmiere einen Code, bei dem die Variable ***stimmung*** die verschiedenen Stimmungen differenziert und die RGB-LED leuchten lässt.

Eigenes Projekt: Digitales Haustier

Jetzt bist du dran, dir ein ganz individuelles Haustier zu programmieren.

1 Gehe Schritt für Schritt vor.
a) Erstelle ein Profil deines digitalen Haustiers:

Name: _____

besondere Merkmale: _____

Lieblingsessen/Lieblingstrinken: _____

viel/wenig Bewegung: _____

viel/wenig Schlaf: _____

Freunde: _____

_____

b) Kombiniere: Die Stimmungsanzeige mit zwei extra Bedürfnissen, z.B. Bewegung und Hunger.

d) Für die Stimmungsanzeige verfahre nach einem Punktesystem: Verteile Punkte für die unterschiedlichen Zustände der Bedürfnisse.

Bedürfnis	Stimmung
Bewegung	0 - 5
Hunger	0 - 5

e) Für die Stimmung werden die Punkte der Bedürfnisse addiert. Teile die Summe je nach Belieben in Stufen ein.

Stimmung	Farbe
< 3	pink
< 5	orange
< 7	grün
≤ 10	Regenbogen

Die Programmierung der unterschiedlichen Bedürfnisse werden kombiniert zu einem Programm. Verwende den Code der vorherigen Anwendungen.

2 Gehe Schritt für Schritt vor:

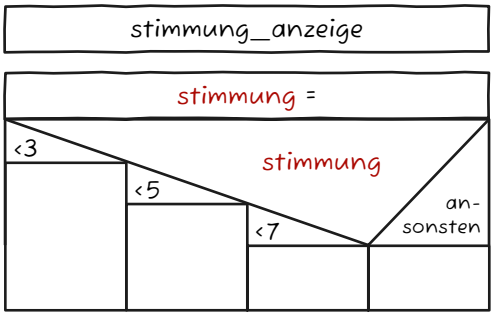
a) Definiere die benötigten Variablen in der Start-Funktion.

b) Lege die Eingaben fest. Achte darauf, dass Eingaben eindeutig sind und nicht doppelt belegt sind.

c) Befehle aus der Dauerhaft-Schleife werden in einer Dauerhaft-Schleife zusammengefügt.

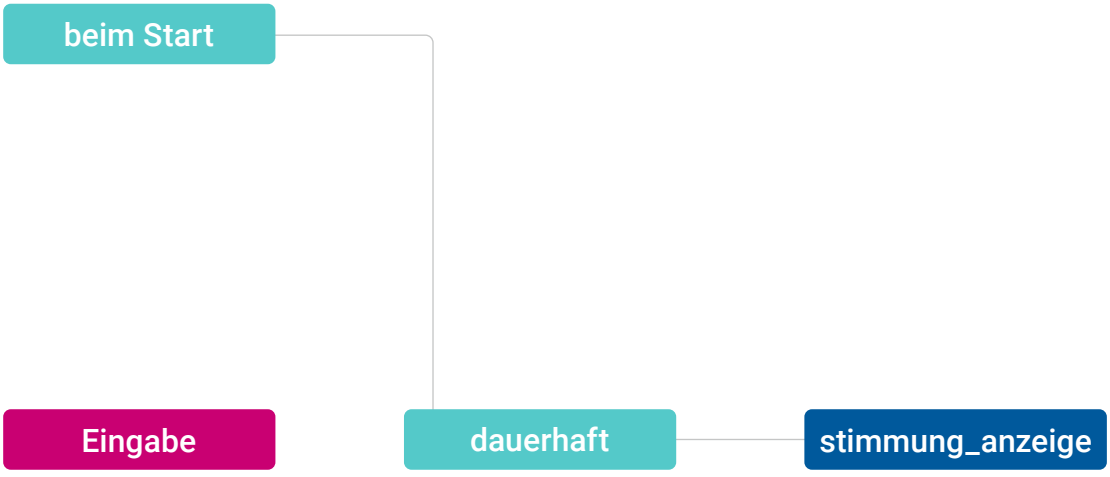
d) Integriere die Funktion **stimmung_anzeige** anhand des Struktogramms. Rufe diese Funktion aus der Dauerhaft-Schleife auf.

e) Überprüfe im Simulator.



3 **Programmfunktionen**

Hier findest du eine Übersicht der gesamten Programmfunktionen.
Ergänze in Stichpunkten, welche Aufgaben die 4 Funktionen haben.



DIGITALES HAUSTIER

Hier ist Platz für dein Struktogramm oder deinen Programmablaufplan:

DIGITALES HAUSTIER

Woran hast du länger getüftelt?

Was funktioniert schon ziemlich gut?

Was funktioniert noch nicht so gut?

DER KLASSENRAUM WIRD ZUM ESCAPE-ROOM

Durch das Lösen der einzelnen Aufgaben kommt ihr ans Ziel.

ESCAPE-ROOM

Erarbeitet anhand der Anleitungen die folgenden Aufgaben, um im Anschluss einen eigenen Escape-Room zu entwickeln.

In den folgenden Aufgaben entwickelt ihr einzelne Challenges:

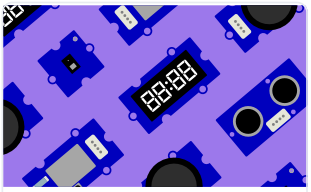
1. Entfernung messen
2. Lichtstärke messen
3. Lautstärke messen
4. Feuchtigkeit messen
5. Lichtschalter
6. Binärcode
7. Kompass

---

1 Entfernung messen

Aufgabe: Stellt den Entfernungsmesser so ein, dass die gemessene Entfernung zwischen 20-22 cm liegt.

Wichtige Blöcke:



Grove

MakeCode Paket für Seeed Studio Module

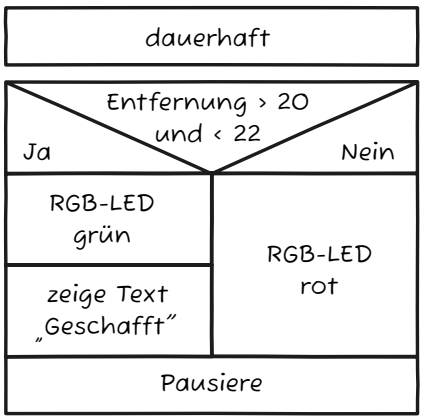
Eingabe: Ultraschallsensor
Verarbeitung: Auswertung gemessener Werte (Entfernung)
Ausgabe: RGB-LED und LED-Matrix

Geht Schritt für Schritt vor:

a) Ladet die Grove-Erweiterung in deine Bibliothek.

b) Lasst euch im ersten Schritt dauerhaft die Werte des Ultraschallsensor anzeigen, um ein Gefühl für die gemessenen Werte zu bekommen.

c) Programmiert anhand des Struktogramms.



2 Lichtstärke messen

Aufgabe: Findet eine dunkle Stelle im Raum.

Wichtige Blöcke:

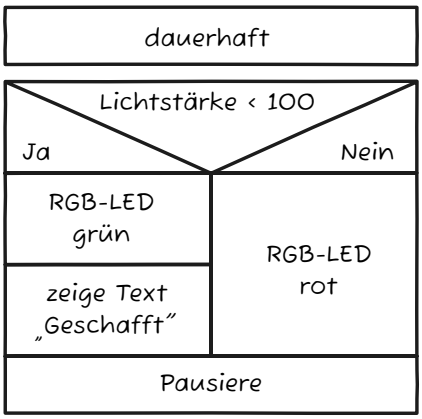
Eingabe: Lichtsensor
Verarbeitung: Auswertung gemessener Werte (Lichtstärke)
Ausgabe: RGB-LED

Geht Schritt für Schritt vor:

a) Lasst euch im ersten Schritt dauerhaft die Werte des Lichtsensors anzeigen, um ein Gefühl für die gemessenen Werte zu bekommen.

b) Programmiert anhand des Struktogramms.

c) Testet den Schwellenwert und passt diesen dem Raum entsprechend an.



Extra

Erweitere die Aufgabe um die Abfrage der Zeit.
Findet eine dunkle Stelle im Raum und verweilt dort für 10 Sekunden.

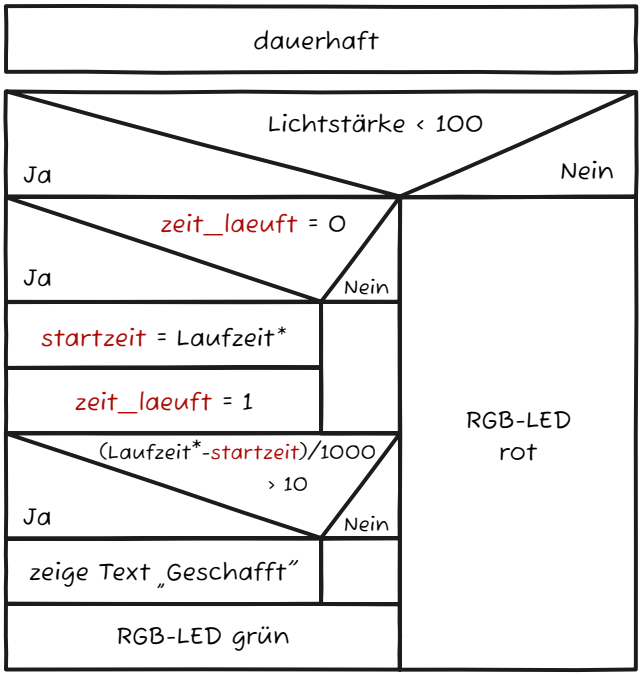
Geht Schritt für Schritt vor:

a) Für die Zeitabfrage werden zwei Variablen benötigt:
startzeit: Speichert den Zeitpunkt, ab wann die Zeit gestoppt wird.
zeit_laeuft: Gibt wieder, ob die Zeit gerade gemessen wird oder nicht. Sie kann wahr oder falsch (nicht wahr) sein.

b) Betrachtet den Code und beschreibe, was passiert.

c) Stellt eine Rechnung auf, mit der ihr abfragen könnt, ob 10 Sekunden vergangen sind. Programmiert mit Hilfe der folgenden Codeblöcke:

d) Programmiert anhand des Struktogramms.



* Laufzeit in ms

Hinweis:
Vergangene Zeit in Millisekunden = Laufzeit (ms) - **startzeit**

Vergangene Zeit in Sekunden = Vergangene Zeit in Millisekunden / 1000

3 Lautstärke messen

Aufgabe: Schafft ihr es, ganz still zu sein?

Wichtige Blöcke:

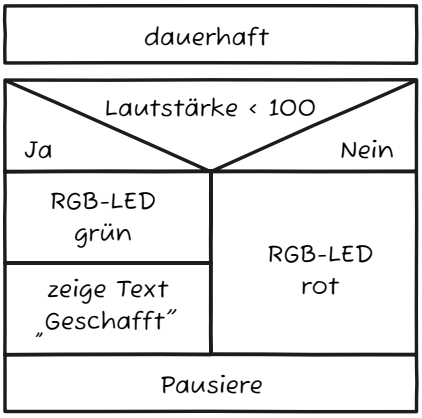
Eingabe: Lautstärke
Verarbeitung: Auswertung gemessener Werte (Lautstärke)
Ausgabe: RGB-LED

Geht Schritt für Schritt vor:

a) Um ein Gefühl für die gemessenen Werte zu bekommen, lasst euch im ersten Schritt dauerhaft die Lautstärke anzeigen.

b) Programmiert anhand des Struktogramms.

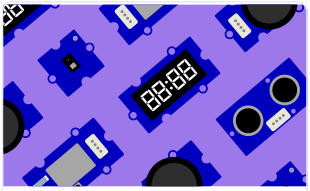
c) Testet den Schwellenwert und passt diesen dem Raum entsprechend an.



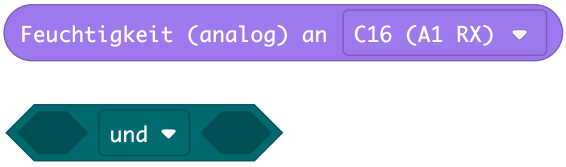
4 Feuchtigkeit messen

Aufgabe: Gießt die Blume im Klassenraum, bis der gemessene Feuchtigkeitswert > 300 liegt.

Wichtige Blöcke:



Grove
MakeCode Paket für Seeed Studio Module



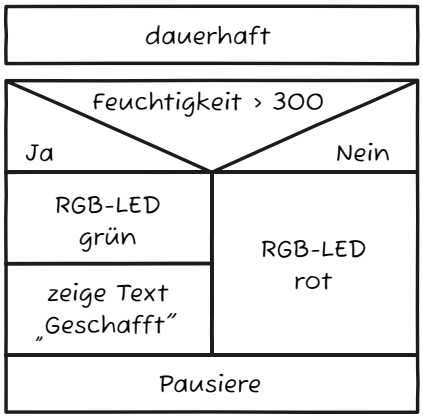
Eingabe: Feuchtigkeitssensor
Verarbeitung: Auswertung gemessener Werte (Feuchtigkeit)
Ausgabe: RGB-LED

Geht Schritt für Schritt vor:

a) Ladet die Grove-Erweiterung in eure Bibliothek.

b) Lasst euch im ersten Schritt dauerhaft die Werte des Ultraschallsensor anzeigen, um ein Gefühl für die gemessenen Werte zu bekommen.

c) Programmiert anhand des Struktogramms.



5 Lichtschalter

Aufgabe: Schaltet die LED-Matrix durch ein Klatschen an und durch ein weiteres Klatschen wieder aus.

Wichtige Blöcke:

Eingabe: Mikrofon
Verarbeitung: Auswertung gemessener Werte (Lautstärke)
Ausgabe: LED-Matrix und RGB-LED

Geht Schritt für Schritt vor:

a) Erstellt ein Struktogramm anhand folgender Logik.
Beim Start:
- Setze die Variable ***schalter*** auf falsch.
- Licht (LED-Matrix) ist aus.

Prüfe dauerhaft:
- Wenn geklatscht wird, prüft, ob das Licht an oder aus ist.
- Ob das Licht an oder aus ist, wird in der Variablen ***schalter*** gespeichert.
- Wenn geklatscht wird und das Licht ist aus, schaltet es an.
- Wenn geklatscht wird und das Licht ist an, schaltet es aus.

b) Programmiert anhand des Struktogramms.
Hinweis: Geschafft ist die Aufgabe, wenn das Licht an und wieder ausgeschaltet wurde.

6 Binärcode

Aufgabe: Schreibe die Zahl 36 im Binärcode.
Gib über die Tasten den Binärcode ein: Knopf A = 0 und Knopf B = 1.
Schließe deine Eingabe mit dem gleichzeitigen Drücken der Tasten A + B ab.

a) Vervollständigt die Tabelle.

Dezimal-zahlen	Zweierpotenzen							Binärcode
	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
	64	32	16	8	4	2	1	
12				1	1	0	0	8 + 4 -> **1100**
24			1	1	0	0	0	16 + 8 -> **11000**
76	1	0	0	1	1	0	0	64 + 8 + 4 -> **1001100**
36								

Wichtige Blöcke:

Eingabe: Taste A und Taste B
Verarbeitung: Speicherung der Tastenkombination, Vergleich Zeichenketten
Ausgabe: LED-Matrix und RGB-LED

Gehe Schritt für Schritt vor.
b) Setze den Binärcode der Zahl 36 als Wert der Variable ***ziel***.

c) Starte mit einem leeren String in der Variable ***code.***

d) Wird die Taste A gedrückt, verbinde den aktuellen Wert der Variable ***code*** mit einer „0“.

e) Verwende die gleiche Logik für die Taste B.

f) Werden die Tasten A+B gleichzeitig gedrückt, vergleiche die Variable ***ziel*** mit der Variable ***code***.

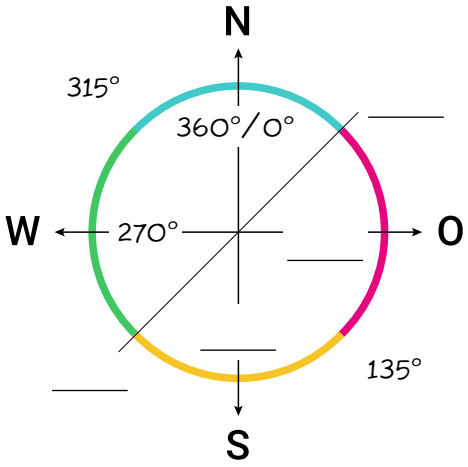
g) Sind die Werte der Variablen identisch, ist die Aufgabe gelöst.

h) Ist die Bedingung nicht erfüllt, leere die Variable ***code***, sodass eine erneute Eingabe möglich ist.

7 Kompass

Aufgabe: Kalibriert den Calliope mini Kompass und stellt euch so hin, dass ihr in Richtung Westen guckt.

Teilt 360° in die vier Himmelsrichtungen ein und ergänze die fehlenden Zahlen.



Wichtige Blöcke:

`Kompassausrichtung (°)`

Eingabe: Kompass
Verarbeitung: Abfrage
Ausgabe: LED-Matrix

Gehe Schritt für Schritt vor.

a) Schreibt die aktuelle Kompassausrichtung in die Variable ***kompass***.

b) Fragt die Kompassausrichtung für die unterschiedlichen Himmelsrichtungen ab.
Tipp: Teile den Norden in zwei Bereiche ein.

Hinweis: Zu Anfang muss der Kompass einmalig kalibriert werden. Folge der Anweisung auf der LED-Matrix: „Tilt to fill Screen". Drehe und wende den Calliope mini solange bis die komplette LED-Matrix gefüllt ist.

Eigenes Projekt: Escape-Room

Setzt euch im Team zusammen und entwickelt eine Escape-Room Anwendung für ein weiteres Team. Verändert die Aufgaben nach euren Ideen, legt die Reihenfolge fest und programmiert euer Programm so, dass die Aufgaben hintereinander bearbeitet werden können.

Der Spielablauf
Das Lösungsteam erhält eine Aufgabenkarten und löst die einzelnen Aufgaben hintereinander mit Hilfe des Calliope mini. Ist die Aufgabe erfolgreich gelöst, lasst auf der LED-Matrix eine Ziffer zwischen 0-3 ausgeben. Diese Ziffern werden notiert und ergeben am Ende den Lösungscode. Abgeschlossen wird der Escape-Room mit der Eingabe über die Touch-Pins.

1 **Vorbereiten**
Listet bis zu 5 unterschiedliche Aufgaben auf und notiert die jeweiligen Lösungsziffern:

1. _____ ◯
2. _____ ◯
3. _____ ◯
4. _____ ◯
5. _____ ◯

Programmfunktionen

beim Start Starteinstellungen

dauerhaft Abfrage und Aufruf der aktuellen Aufgaben-Funktion

aufgabe 1, 2, ... Funktionen der einzelnen Aufgaben

Wenn die Aufgabe gelöst ist, dann wird die Lösungsziffer ausgegeben.

naechsteAufgabe Start der nächsten Aufgabe

Wenn die letzte Aufgabe gelöst ist, wird die Funktion loesungscode aufgerufen.

loesungscode Eingabe des Lösungscodes

2 Programmierung

Programmiert anhand der Struktogramme.

beim Start

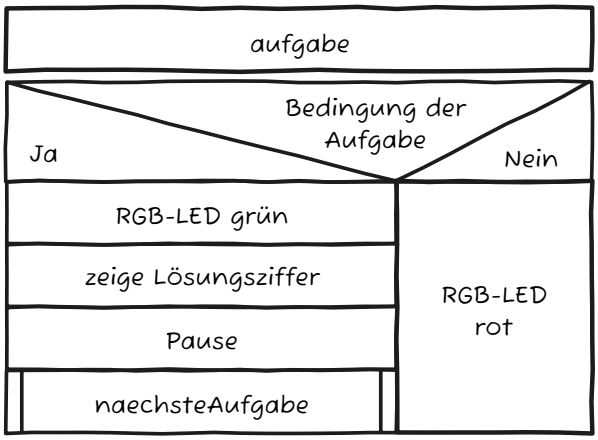
a) In der Start-Funktion werden alle Starteinstellungen festgelegt und Variablen definiert.

Achtet auch darauf, die Variablen aus euren gewählten Aufgaben zu definieren.

Die Variable ***ziel*** setzt sich zusammen aus euren Lösungsziffern.

aufgabe 1, 2, ...

b) Jede Aufgabe wird in einer eigenen Funktion geschrieben. Innerhalb der Funktion wird abgefragt, ob die Aufgabe erfüllt ist. Hier könnt ihr auf den Code der vorherigen Aufgaben zurückgreifen.

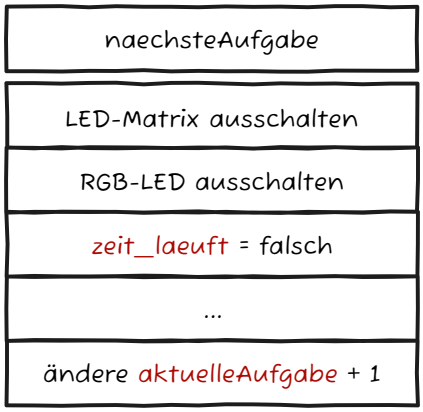


naechsteAufgabe

c) Die Ausgaben, wie die LED-Matrix und die RGB-LED, werden ausgeschaltet.

Die aufgabenspezifische Variablen werden zurückgesetzt.

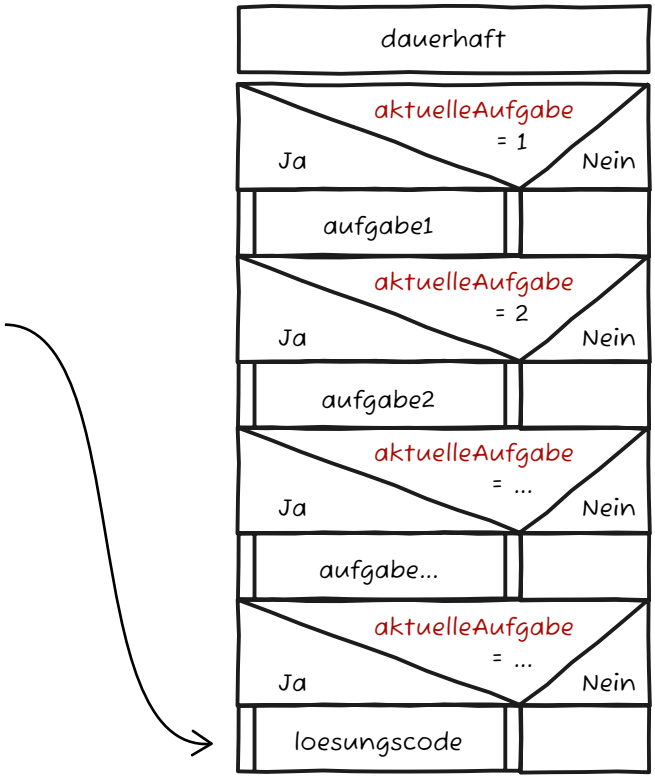
Die Variable ***aktuelleAufgabe*** wird um 1 erhöht.



dauerhaft

d) In der Dauerhaft-Schleife wird die aktuelle Aufgabennummer abgefragt und die dazugehörige Funktion aufgerufen.

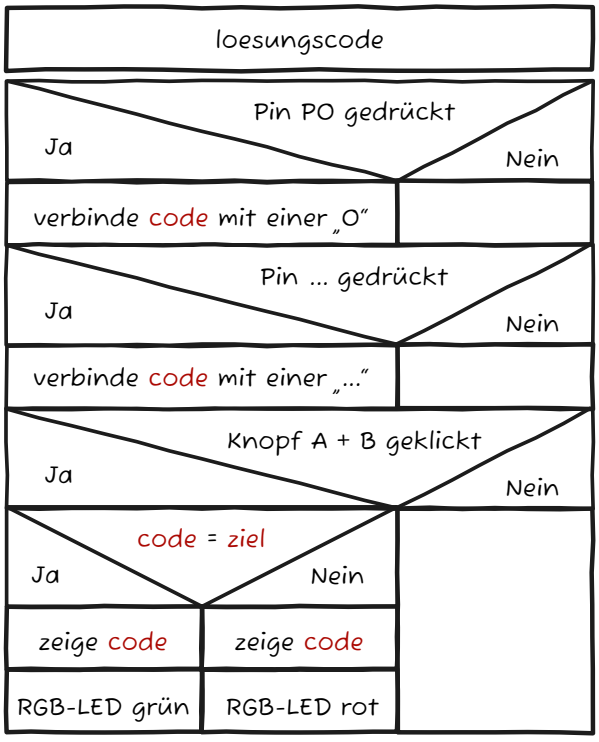
Beachtet, dass die letzte Aufgabe immer die Eingabe des Lösungscodes ist. Ruft dazu die Funktion **loesungscode** auf.



loesungscode

e) In der Funktion **loesungscode** wird die Eingabe gesteuert und der Lösungscode mit dem Ziel-Code verglichen.

Erweitert die Funktion **loesungscode** um die Eingaben für die Pins P0 - P3.



Escape-Room - Aufgabenummer

Aufgabentitel: _____

Aufgabenstellung:

Eingabe: _____

Ausgabe: _____

Aufgabe gelöst? Die Lösungsziffer lautet:

Wird vom Lösungsteam ausgefüllt!

Escape-Room - Aufgabenummer

Aufgabentitel: _____

Aufgabenstellung:

Eingabe: _____

Ausgabe: _____

Aufgabe gelöst? Die Lösungsziffer lautet:

Wird vom Lösungsteam ausgefüllt!

ESCAPE-ROOM

Hier ist Platz für dein Struktogramm oder deinen Programmablaufplan:

ESCAPE-ROOM

Woran hast du länger getüftelt?

Was funktioniert schon ziemlich gut?

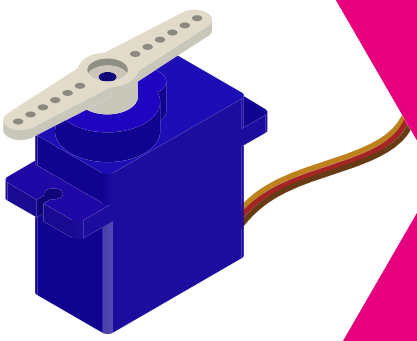
Was funktioniert noch nicht so gut?

BEWEGUNG DURCH MOTOREN

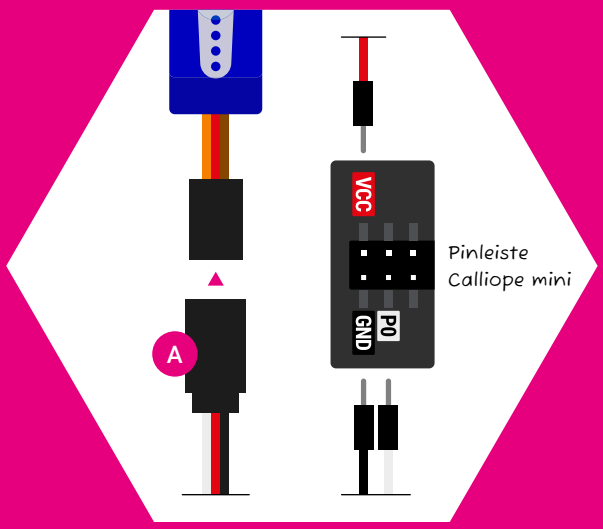
Mit Hilfe von Motoren kommt Bewegung ins Spiel.

SERVOMOTOREN

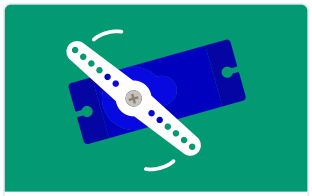
Servomotoren sind kleine elektrische Motoren und können einfach an den Calliope mini angeschlossen werden. Durch einen eingebauten Positionssensor erkennen die Servomotoren den Drehwinkel des Motors und können so sehr präzise gesteuert werden.



Verwende Servomotoren, um Objekte in Bewegung zu setzen.



1. Verbinde das Adapter Kabel (A) mit dem Kabel des Servomotors.
2. Schließe das schwarze Kabel (Minus) an den GND-Pin und das rote Kabel (Plus) an den VCC-Pin. Das weiße Kabel kann an einen beliebigen Pin gesteckt werden.



Servo

Stellt die Servo-Blöcke bereit

Importiere die benötigten Programmblöcke:
Servo
(Erweiterungen)

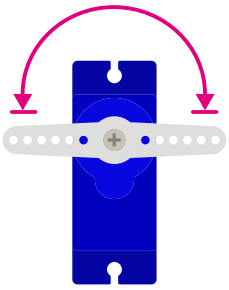
Wichtige Blöcke:

1. Erweiterungen
2. Servos

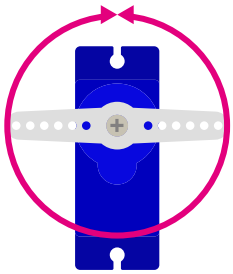
Unterscheidung zwischen Servomotoren

Es gibt zwei Arten von Servomotoren: **180° und 360° Servomotoren.**
Die Gradzahl gibt an, wie weit sich der Motor drehen kann.

180°

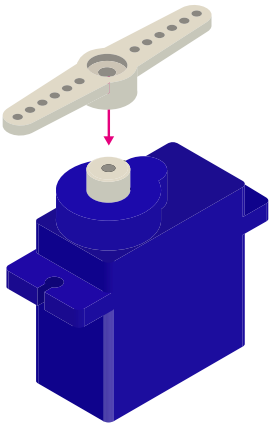


360°



Ob es sich um einen 180° oder 360° Servomotor handelt, erkennst du entweder am Aufkleber am blauen Gehäuse oder du probierst es aus:

Benutze dafür einen der beiliegenden weißen Aufsätze, stecke ihn auf die Motorwelle und drehe vorsichtig. Wenn es irgendwann nicht mehr weiter geht, hast du einen 180° Servomotor. Wenn du den Aufsatz komplett herum drehen kannst, hast du einen 360° Grad Servomotor.

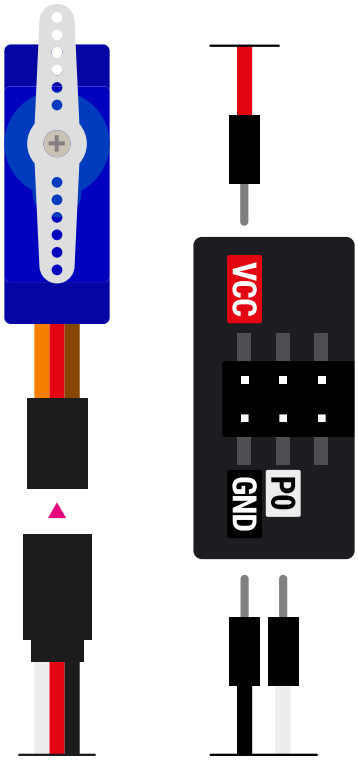


Hinweis: Bei der Programmierung wird die Position des 180° Servomotors über die Gradzahl bestimmt. Beim 360° Servomotor steuerst du die Geschwindigkeit in Prozent (-100% bis 100%).

1 Fallen dir Beispiele für (maximal) 180° und 360° Bewegungen ein.

180°	360°
Schranke	Windmühle

Programmieren mit 180° Servomotoren



Anschließen
Stecke einen der weißen Aufsätze auf die Motorwelle. Verbinde die Kabel des Servomotors mit dem Jumper-Adapter Kabel (schwarz-rot-weiß). Stecke die Jumperkabel mit den Metallstiften in die Pinleiste des Calliope mini:

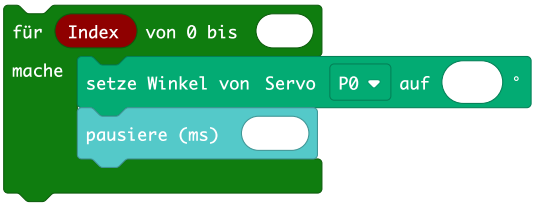
Schwarzes Kabel: **GND**
Rotes Kabel: **VCC**
Weißes Kabel: **Pin P0**

1. Lade die Servo Erweiterung in MakeCode und programmiere eine Steuerung über die Knöpfe:
   a) Wenn Knopf A gedrückt wird, setze den Winkel auf 0°.

   b) Wenn Knopf B gedrückt wird, setze den Winkel auf 180°.

Geschwindigkeit steuern
Die Geschwindigkeit von 180° Servomotoren kann nicht verändert werden. Allerdings kannst du Pausen und Schleifen einsetzen, um die Bewegung zu beschleunigen oder zu verlangsamen.

2. Erweitere dein Programm so, dass der Servomotor mindestens 3 Sekunden für eine Bewegung von 0° bis 180° braucht.



3. Steuere den Servomotor über den Lagesensor des Calliope mini.
   a) Wenn du den Calliope mini nach links neigst, drehe den Servomotor nach links.

   b) Wenn du den Calliope mini nach rechts neigst, drehe den Servomotor nach rechts.

Tipp: Benutze den **Drehung (°) rollen** Block aus der Eingabe Kategorie und verteile die Werte entsprechend. Der Lagesensor gibt Werte von -90 bis 90 aus.

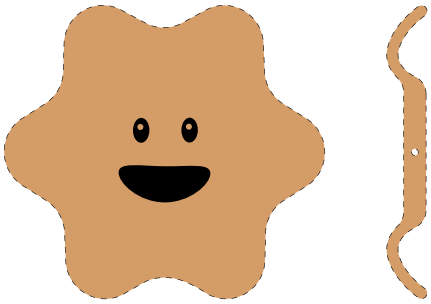
4. Programmiere eine Sequenz für die folgende Bastelanleitung.
   a) Wenn Knopf A gedrückt, lasse den Servo dreimal hintereinander zwischen 70° und 110° bewegen. Benutze Pausen und Schleifen.

BASTELANLEITUNG

Bringe dein digitales Haustier in die reale Welt und animiere die Beine oder Arme mit einem 180° Servomotor:

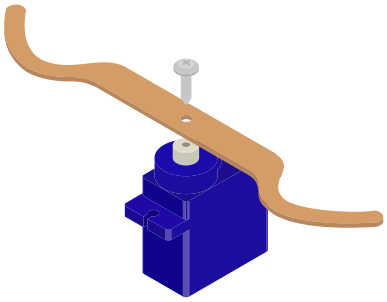
1

Male ein Tier mit extra Armen oder Beinen und schneide es aus. Du kannst auch die Bastelvorlage benutzen. Für mehr Stabilität kannst du die Motive auch auf Pappe übertragen und dann ausschneiden.



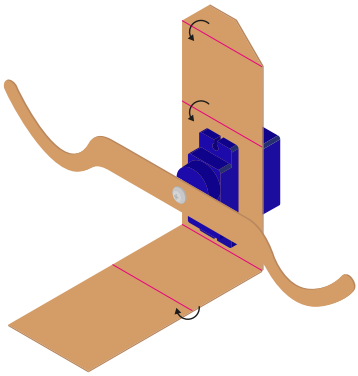
2

Stecke einen weißen Aufsatz auf die Motorwelle und lege deine Arme oder Beine darauf. Befestige alles mit einer Schraube.



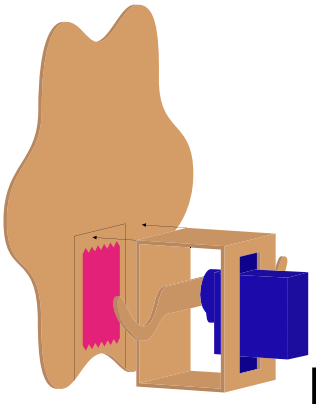
3

Schneide die Servohalterung aus. Führe den Servomotor und die Kabel durch die Öffnung und falte den Streifen anhand der Markierungen, sodass eine Box entsteht. Klebe die Box zusammen.

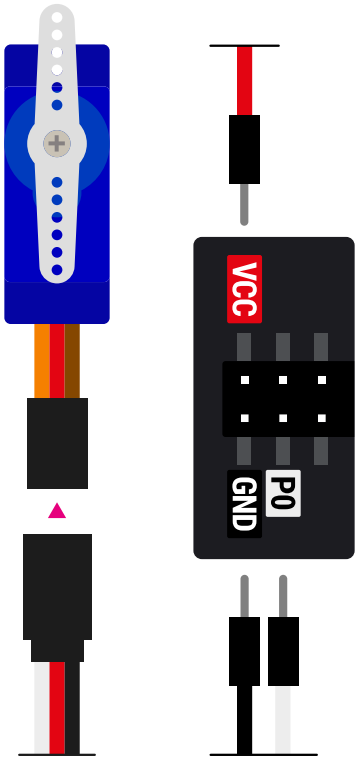


4

Klebe die Box an die Rückseite von deinem Tier.



Programmieren mit 360° Servomotoren



Anschließen
Stecke einen der weißen Aufsätze auf die Motorwelle.
Verbinde die Kabel des Servomotors mit dem Jumper-Adapter Kabel (schwarz-rot-weiß). Stecke die Jumperkabel mit den Metallstiften in die Pinleiste des Calliope mini:

Schwarzes Kabel: **GND**
Rotes Kabel: **VCC**
Weißes Kabel: **Pin P0**

1. Programmiere eine Steuerung für den Servomotor über die Knöpfe:
   a) Wenn Knopf A gedrückt wird, drehe den Servo mit 50% Geschwindigkeit gegen den Uhrzeigersinn.

   b) Wenn Knopf B gedrückt wird, drehe den Servo mit -50% Geschwindigkeit im Uhrzeigersinn.

   c) Wenn Knopf A+B gedrückt wird, stoppe den Servo.

2. Steuere den Servomotor durch die Lautstärke in der Umgebung. Starte den Servomotor durch ein Klatschen und stoppe ihn durch ein weiteres Klatschen.

   Tipp: Du hast auf Seite 42 einen Lichtschalter programmiert, der das gleiche Prinzip anwendet.

3. Steuere den Servomotor durch die Lichtstärke der Umgebung.
   So kannst du das Tempo durch die Lichtstärke beeinflussen.
   Verteile den Wertebereich der Lichtstärke (0-255) auf die Geschwindigkeit (0-100).

4. Programmiere ein Kreisel-Orakel:
   a) Wenn Knopf A gedrückt, drehe den Servo mit 100% Geschwindigkeit in eine Richtung.

   b) Stoppe den Servo zu einem zufälligen Zeitpunkt.

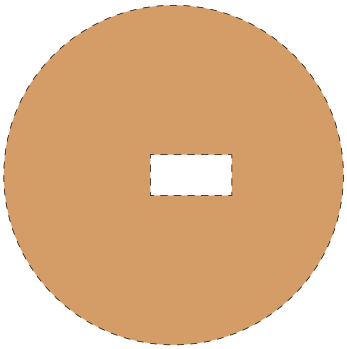
   In der folgenden Bastelanleitung teilst du einen Kreis in verschiedene Segmente und kannst dein Kreisel-Orakel richtig ausprobieren.

BASTELANLEITUNG

Baue ein Kreisel-Orakel mit dem 360° Servomotor.

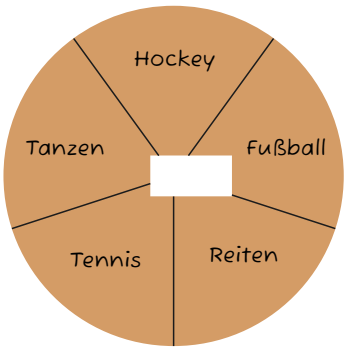
1

Schneide die Kreisvorlage aus. Für eine größere Stabilität kannst du die Vorlage auch auf Pappe übertragen und ausschneiden.



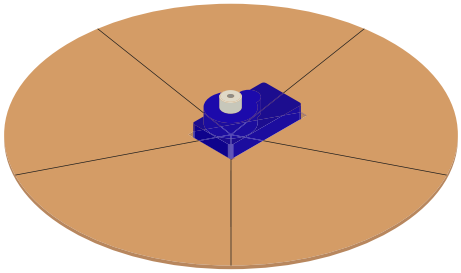
2

Beschrifte die Vorderseite mit deinen Auswahl-Möglichkeiten für das Orakel.



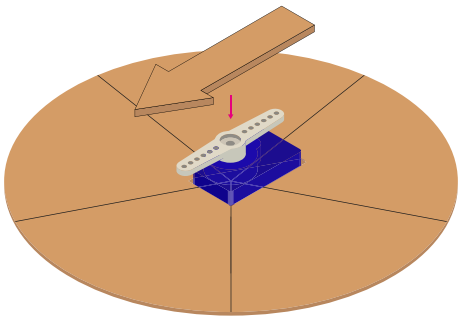
3

Stecke den Servomotor von unten in die Kreisvorlage. Achte darauf, dass die weiße Motorwelle in die Mitte zeigt.



4

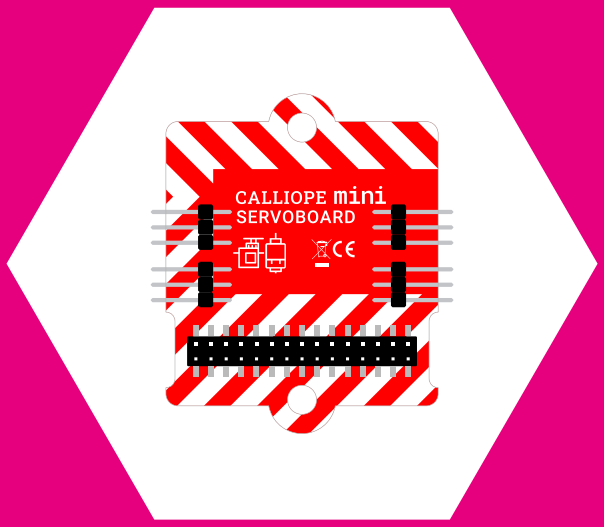
Schneide einen Pfeil aus und befestige ihn zusammen mit einem weißen Aufsatz auf dem Servomotor. Benutze dafür die Schraube oder Klebeband.



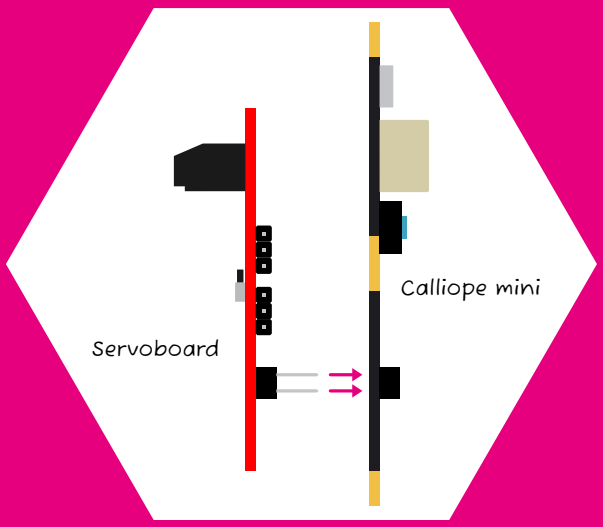
SERVOBOARD

Servo- und Gleichstrommotoren anschließen!

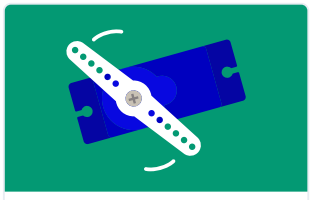
An das Servoboard können bis zu 4 Servomotoren und 2 Gleichstrommotoren angeschlossen werden. In Betrieb können 4 Aktoren gleichzeitig bewegt werden.



Verwende das Servoboard, um Motoren zu steuern.



Stecke das Servoboard von hinten in die Pinleiste des Calliope mini.

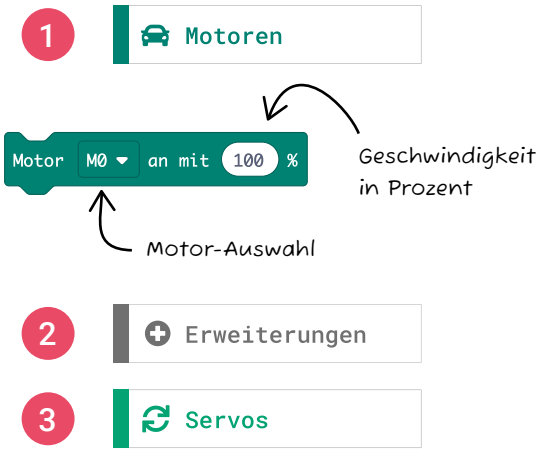


Servo

Stellt die Servo-Blöcke bereit

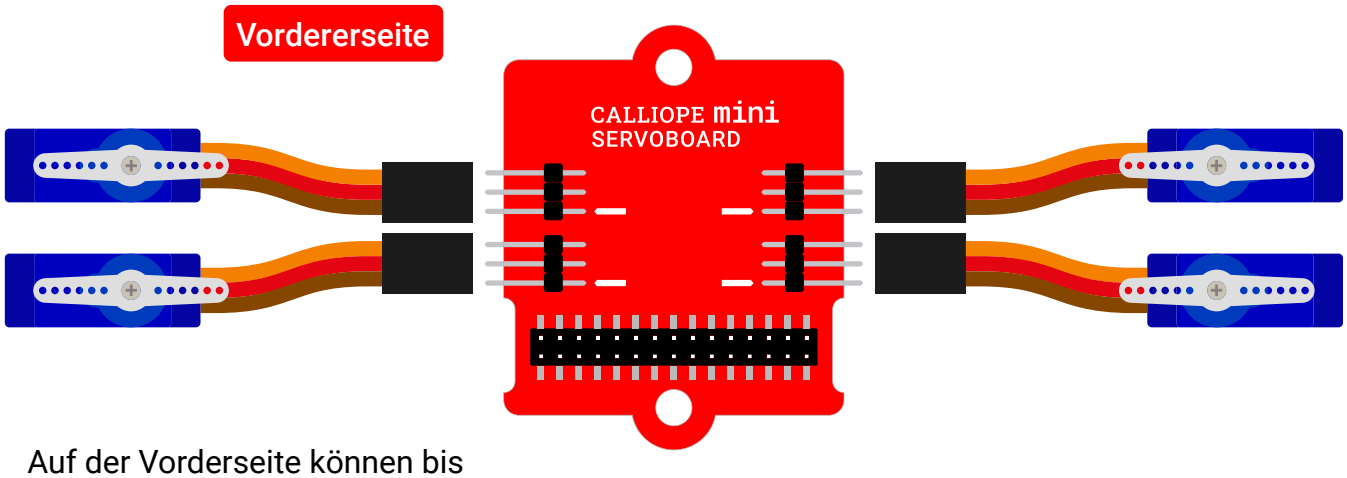
Importiere die benötigten Programmblöcke:
Servo
(Erweiterungen)

Wichtige Blöcke:

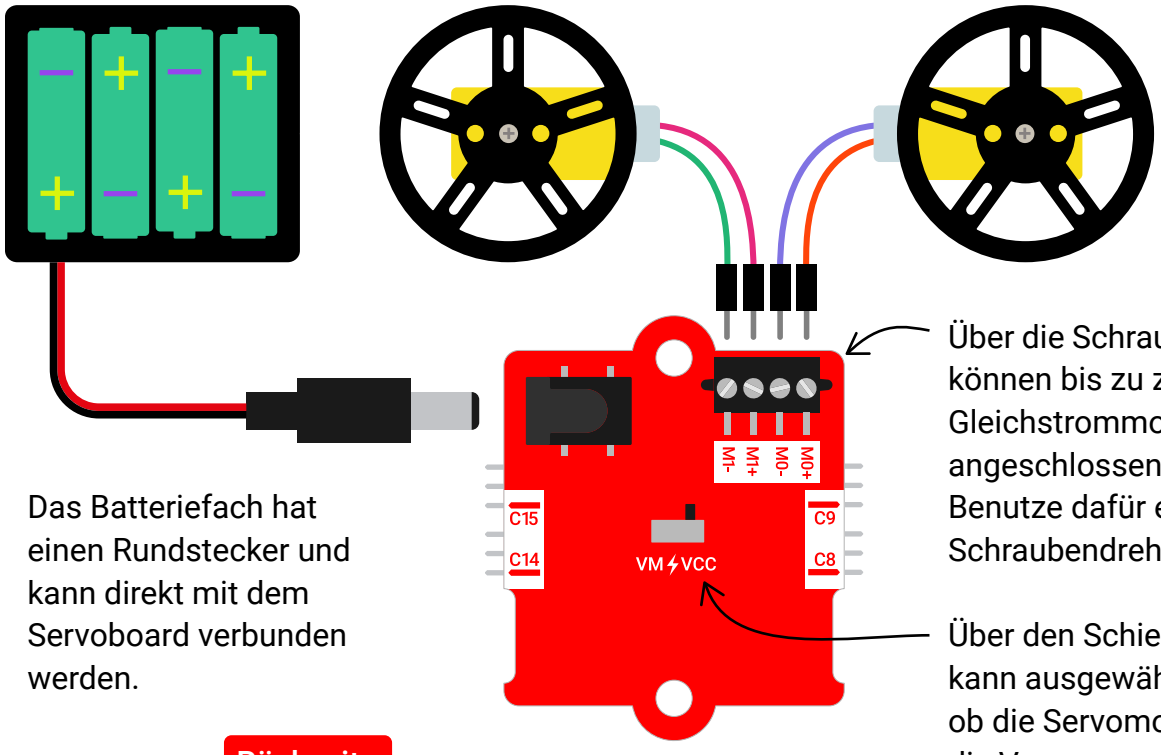


Anschlussmöglichkeiten

Mit Hilfe des Servoboards lassen sich Servo- und Gleichstrommotoren einfach und unkompliziert an den Calliope mini anschließen.



Auf der Vorderseite können bis zu 4 Servomotoren angeschlossen werden.
Achte auf die richtige Ausrichtung der Kabel. Die Pfeile auf der Rückseite des Servoboards zeigen auf den GND-Pin (braunes Kabel).



Das Batteriefach hat einen Rundstecker und kann direkt mit dem Servoboard verbunden werden.

Über die Schraubklemmen können bis zu zwei Gleichstrommotoren angeschlossen werden. Benutze dafür einen Schraubendreher.

Über den Schiebeschalter kann ausgewählt werden, ob die Servomotoren über die Versorgungsspannung des Calliope mini (VCC) oder die Motorspannung (VM) betrieben werden.

Rückseite

Hinweis: Verwende standardmäßig die VCC Einstellung. Sollte der Servomotor sich zu langsam drehen, ändere den Schalter auf VM.

Eigenes Projekt: Seifenblasenmaschine

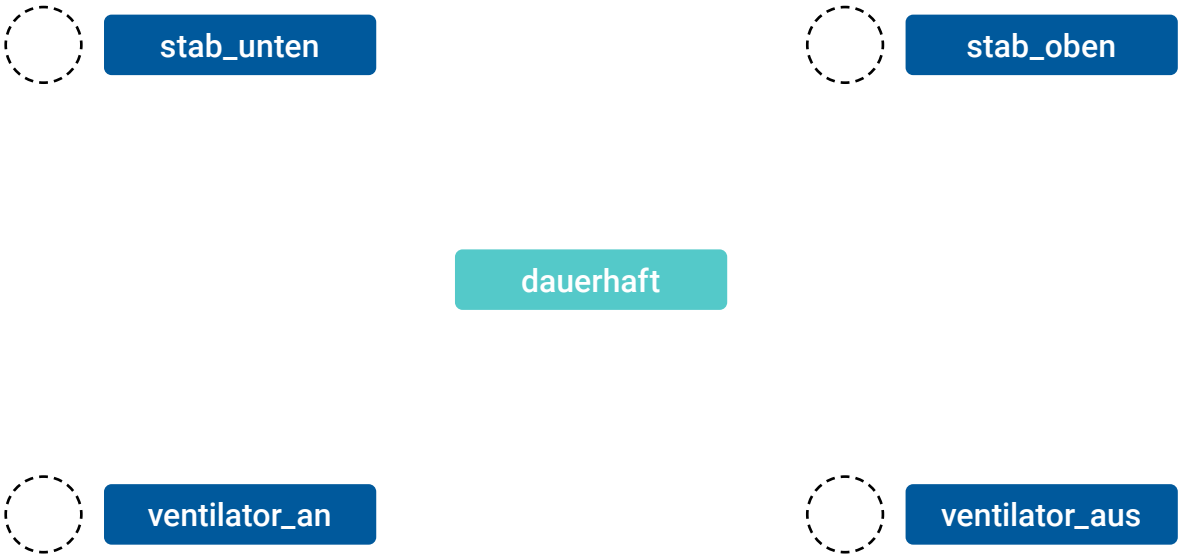
Setzt euch im Team zusammen und entwickelt eine Maschine, die über einen Ultraschallsensor gesteuert Seifenblasen produziert.

Was benötigt ihr?

- Einen **180° Servomotor**, der den Seifenblasenstab steuert:
  a) Eintauchen in das Seifenblasenwasser
  b) Positionierung vor dem Ventilator
- Einen **Ventilator,** der die Luft für die Seifenblasen liefert.
- Ein **Servoboard** inklusive Batterfach, das den Strom liefert.
- Einen **Ultraschallsensor**, der die Maschine startet.
- Einen Seifenblasenstab, ein Behältnis für das Seifenblasenwasser und natürlich einen Calliope mini.

1 Geht Schritt für Schritt vor:

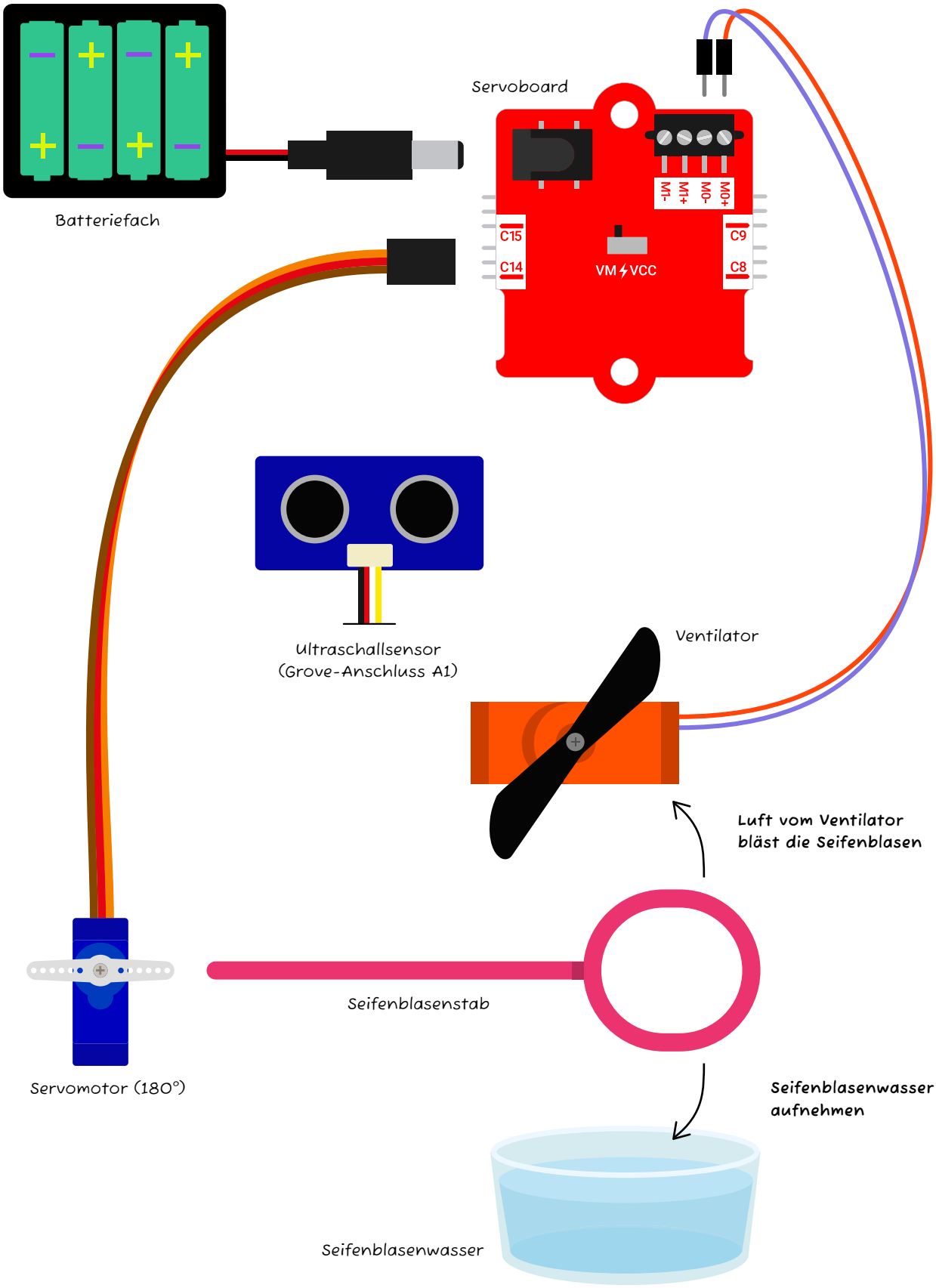
a) Überlege dir den Ablauf der Seifenblasenmaschine und vervollständige die Übersicht der Funktionen.



b) Schaut euch die Skizze an und erstellt einen Programmablauf, programmiert anhand des Ablaufplans die Seifenblasenmaschine.
Testet die Aufrufe der einzelnen Funktionen und optimiert die Bewegung des Servomotors, die Geschwindigkeit des Ventilators sowie die zeitliche Abfolge.

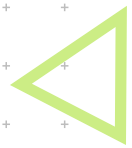
BASTELANLEITUNG

Befestigt die Bauteile an einer dicken Pappe oder an einem Karton und baut euch eine automatische Seifenblasenmaschine.



SEIFENBLASENMASCHINE

Hier ist Platz für dein Struktogramm oder deinen Programmablaufplan:



SERVOBOARD

Woran hast du länger getüftelt?

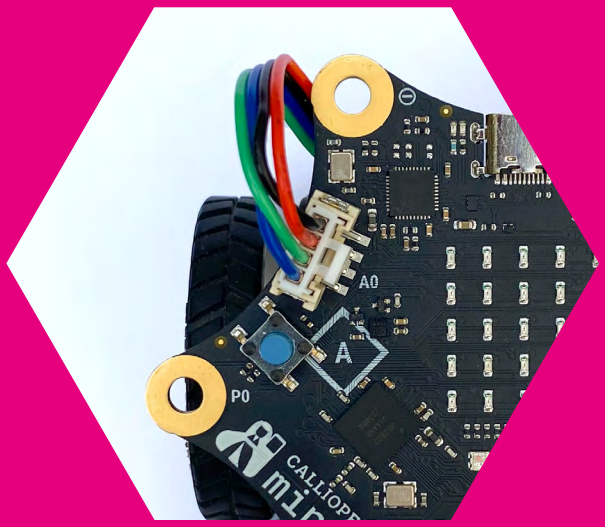
Was funktioniert schon ziemlich gut?

Was funktioniert noch nicht so gut?

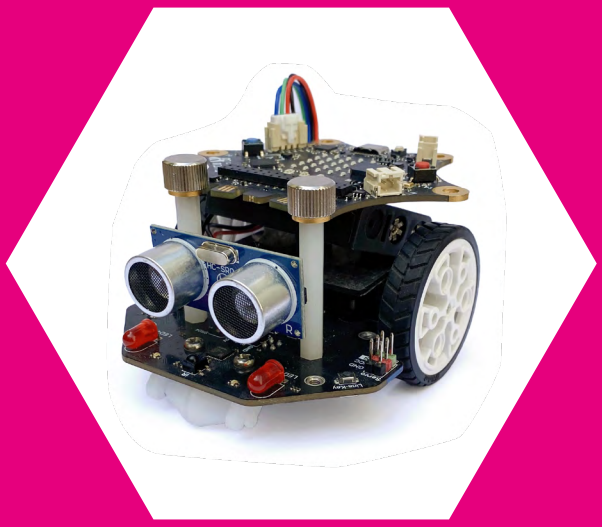
MOTIONKIT

Zusammenbauen und losfahren!

Der Calliope mini bringt das MotionKit ins Rollen. Die Motoren bewegen die Räder und setzen den Calliope mini in Bewegung. Beide Motoren können unabhängig in Richtung (vorwärts, rückwärts, anhalten) und Leistung (0-255) gesteuert werden.



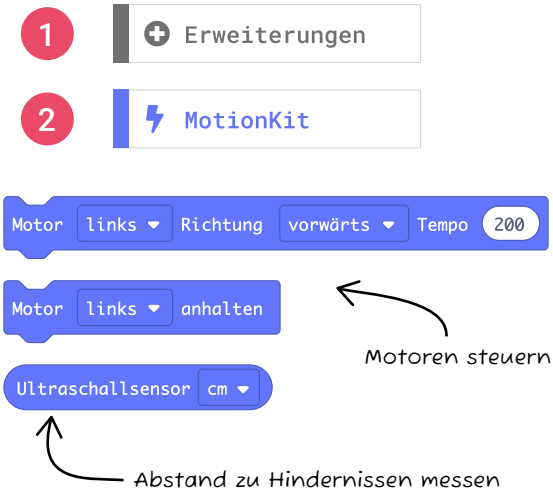
Mit dem Grove-Kabel wird das MotionKit über den Grove-Anschluss A0 mit dem Calliope mini verbunden.



Steuere das MotionKit selbst oder programmiere eine autonome Steuerung.

Importiere die benötigten Programmblöcke:
MotionKit V2
(Erweiterungen)

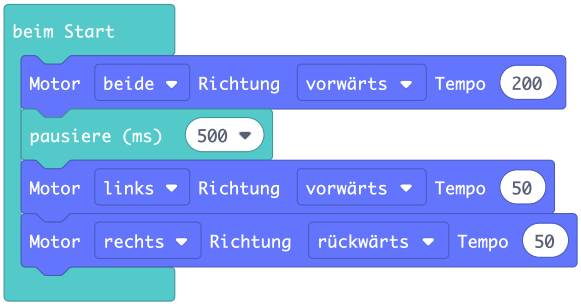
Wichtige Blöcke:



Fahrbewegungen

Die Fahrbewegung wird bestimmt über die Bewegungsrichtung, die Geschwindigkeit und die Dauer der Bewegung.

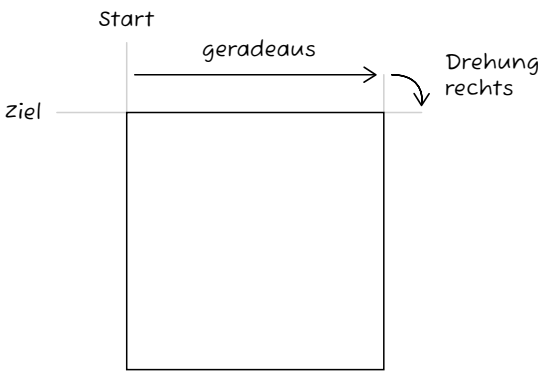
1 Schau dir die Programmcodes genau an und beschreibe was passiert.



2 Passe die Parameter **Tempo** und **Zeit** so an, dass das MotionKit eine 90° Kurve fährt.

3 Erstelle ein Programm, dass den Calliope mini ein Quadrat fahren lässt.
Gehe Schritt für Schritt vor:

a) Unterteile die Strecke in einzelne Einheiten und kennzeichne Wiederholungen.



b) Erstelle ein Struktogramm.

c) Programmiere anhand des Struktogramms.

d) Übertrage es auf deinen Calliope mini. Teste und beobachte die Fahrweise.

Extra

e) Strukturiere deinen Code, indem du Funktionen verwendest.

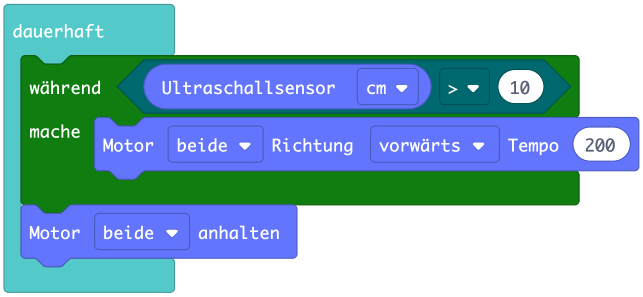
f) Vereinfache deinen Code durch Wiederholungen.

Selbstfahrender Roboter

Eine Aufgabe des autonomen Fahrens ist es, Hindernisse zu erkennen und diesen auszuweichen. Mithilfe des Ultraschallsensors lassen sich Hindernisse erkennen.

1. Beschreibe drei verschiedene Anwendungen von selbstfahrenden Robotern, bei denen das Erkennen von Hindernissen eine Rolle spielt.

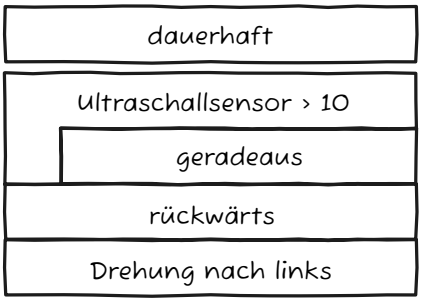
2. Stecke den Ultraschallsensor in den vorgesehenen Anschluss und übertrage den Programmcode auf deinen Calliope mini.



3. Beschreibe das Verhalten des Calliope mini.

4. Erweitere das Programm so, dass das MotionKit solange geradeaus fährt, bis es auf ein Hindernis trifft, dann ein Stück rückwärts fährt, eine Drehung nach links macht und dann weiterfährt.

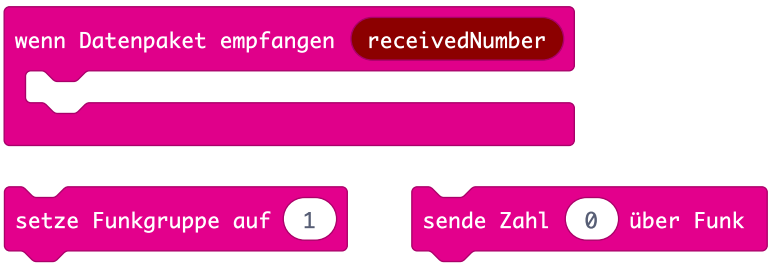
   Programmiere anhand des Struktogramms.



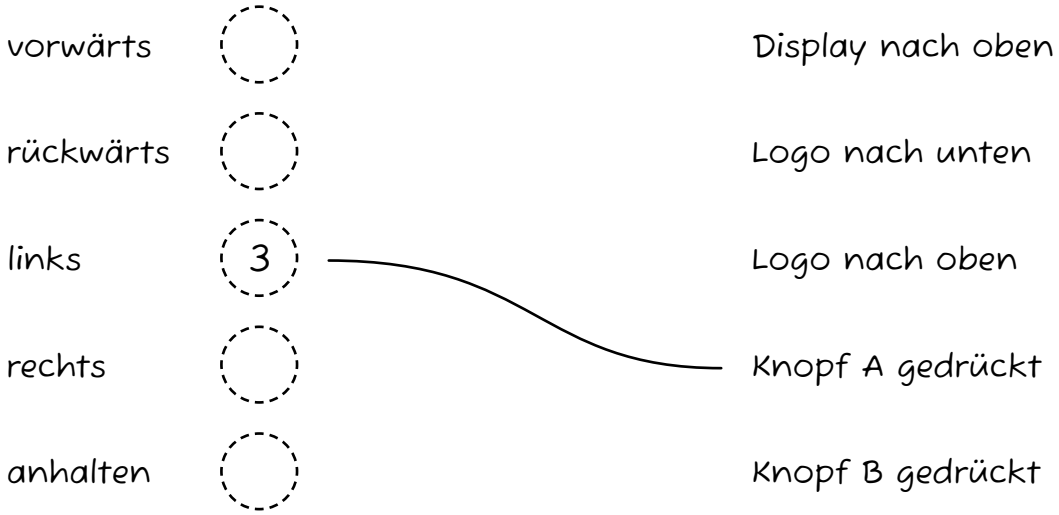
Ferngesteuerter Roboter

Über Funk lassen sich Roboter auch fernsteuern. Ein Calliope mini ist der Sender, der die Steuerbefehle als Nachricht sendet. Diese werden von einem weiteren Calliope mini am MotionKit empfangen, interpretiert und ausgeführt. Für das Senden und Empfangen von Nachrichten muss einmalig in der Startfunktion eine Funkgruppe festgelegt werden. Einigt euch auf eine pro Team und los geht's!

Wichtige Blöcke:



1 **Vorbereiten**
Legt folgende Funktionen für die Fernsteuerung fest und weist jeder Funktion eine intuitive Eingabe und eine der vorab definierten Nummern zu.



2 Programmiere als erstes die **Fernsteuerung - Daten senden**.
Gehe Schritt für Schritt vor:

a) Beschreibt das Programm auf der rechten Seite.

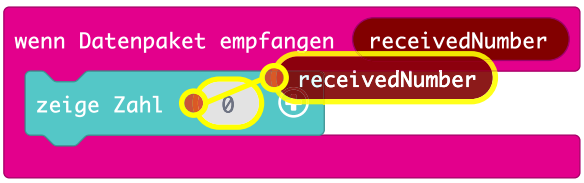
b) Programmiert den Calliope mini so, dass bei der jeweiligen Eingabe die zugewiesene Nummer auf der LED-Matrix angezeigt wird.

c) Kontrolliert zuerst die Zuweisung der Eingaben über die Ausgabe.

d) Sendet per Funk die zugeordneten Nummern aus der vorherigen Aufgabe 1.

3 Programmiere das **MotionKit - Daten empfangen**.
Gehe Schritt für Schritt vor:

a) Programmiert den Calliope mini Roboter so, dass die empfangende Zahl auf der LED-Matrix angezeigt wird. **Wichtig: Funkgruppe definieren!**



b) Definiert 5 Funktionen für das jeweilige Verhalten: vorwärts, rückwärts, links, rechts, anhalten.

c) Erstellt eine Mehrfachverzweigung und weist jeder Nummer eine auszuführende Aktion durch einen Funktionsaufruf zu.

d) Übertrag das jeweilige Programm auf den entsprechenden Calliope mini und teste deinen ferngesteuerten Roboter.

MOTIONKIT

Hier ist Platz für dein Struktogramm oder deinen Programmablaufplan:

MOTIONKIT

Woran hast du länger getüftelt?

Was funktioniert schon ziemlich gut?

Was funktioniert noch nicht so gut?

WEITERE INFORMATIONEN

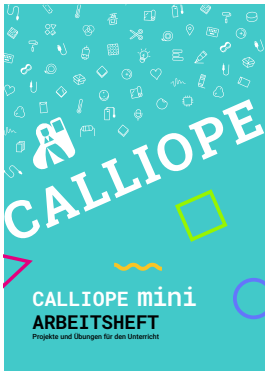
Arbeitsheft 2
Diese Webseite enthält alle Informationen zum Calliope mini Arbeitsheft 2, Zusatzmaterial sowie Vorlagen zum Ausdrucken bereit:
https://calliope.cc/schulen/arbeitsheft-2

Material für Lehrkräfte
Nach der Registrierung auf der Webseite haben Lehrkräfte Zugriff auf Zusatzmaterial mit Lösungen und Tipps für den Unterricht:
1) Registrierung mit der Email-Adresse
2) Passwort erhalten
3) Log in mit Passwort
https://calliope.cc/schulen/arbeitsheft-2

Programmieren mit dem iPad
Auf der Webseite gibt es nützliche Tipps zum Programmieren mit dem iPad: *https://calliope.cc/programmieren/mobil/ipad*

Glossar
Das Glossar auf der Webseite bietet eine Übersicht über die wichtigsten Begriffe und Konzepte rund um den Calliope mini:
https://calliope.cc/begriffe

Arbeitsheft
Das Calliope mini Arbeitsheft bietet eine allgemeine Einführung sowie Aufgaben zum Kennenlernen und Steuern der einzelnen Komponenten.



4.3 Regentropfen fangen

Idee basierend auf dem Material von: Karsten Beuche (calliopemini.info)

CALLIOPE mini
ARBEITSHEFT 2

Projekte und Übungen

- Die Programmierumgebung MakeCode
- Spiele mit dem Calliope mini
- Digitales Haustier
- Escape Room
- Der Calliope mini in Bewegung

CALLIOPE.CC

ISBN 978-3-9825596-2-9