

Calliope Workshop

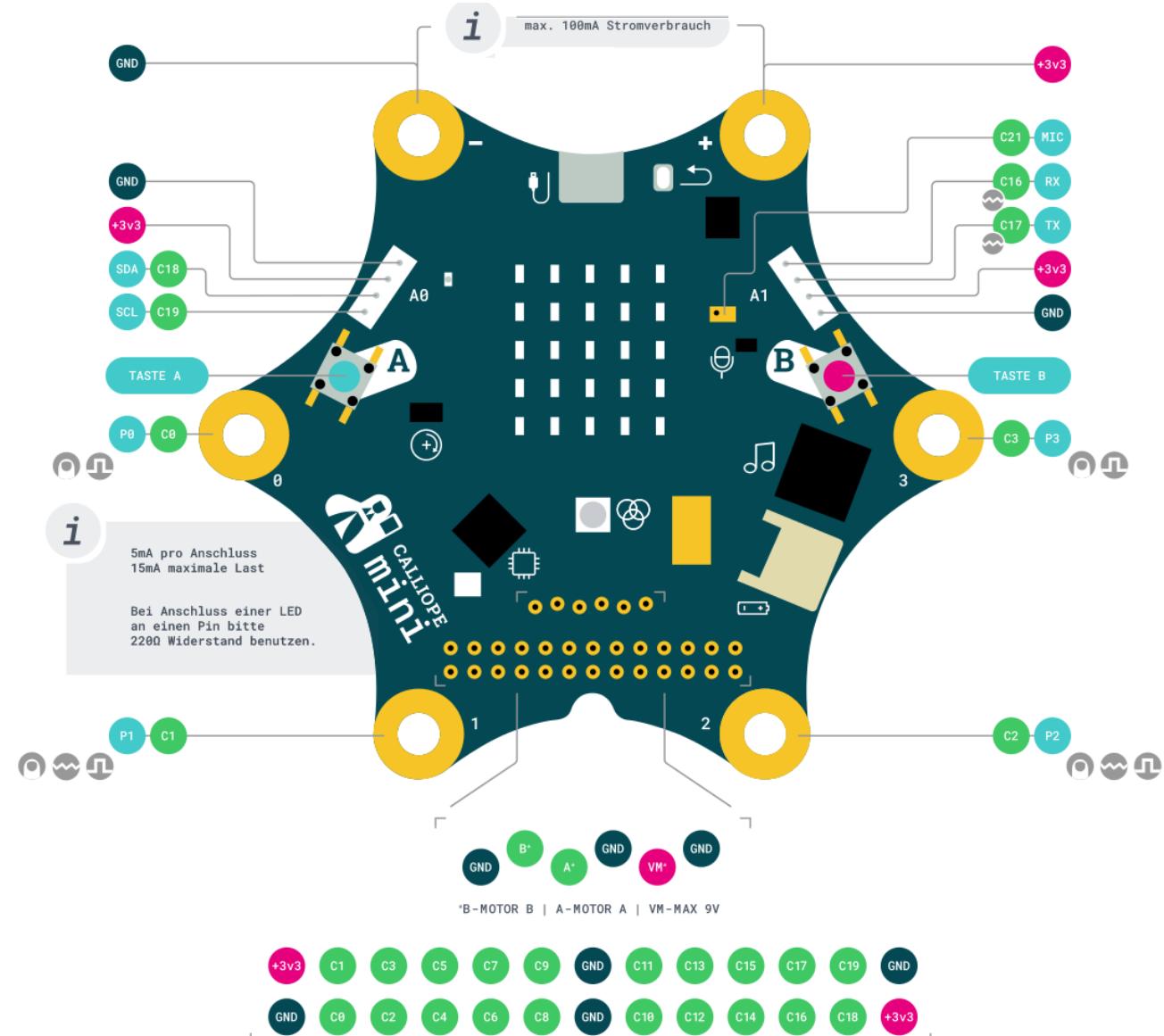


Willkommen im



Calliope Workshop

Die Hardware



Calliope Workshop

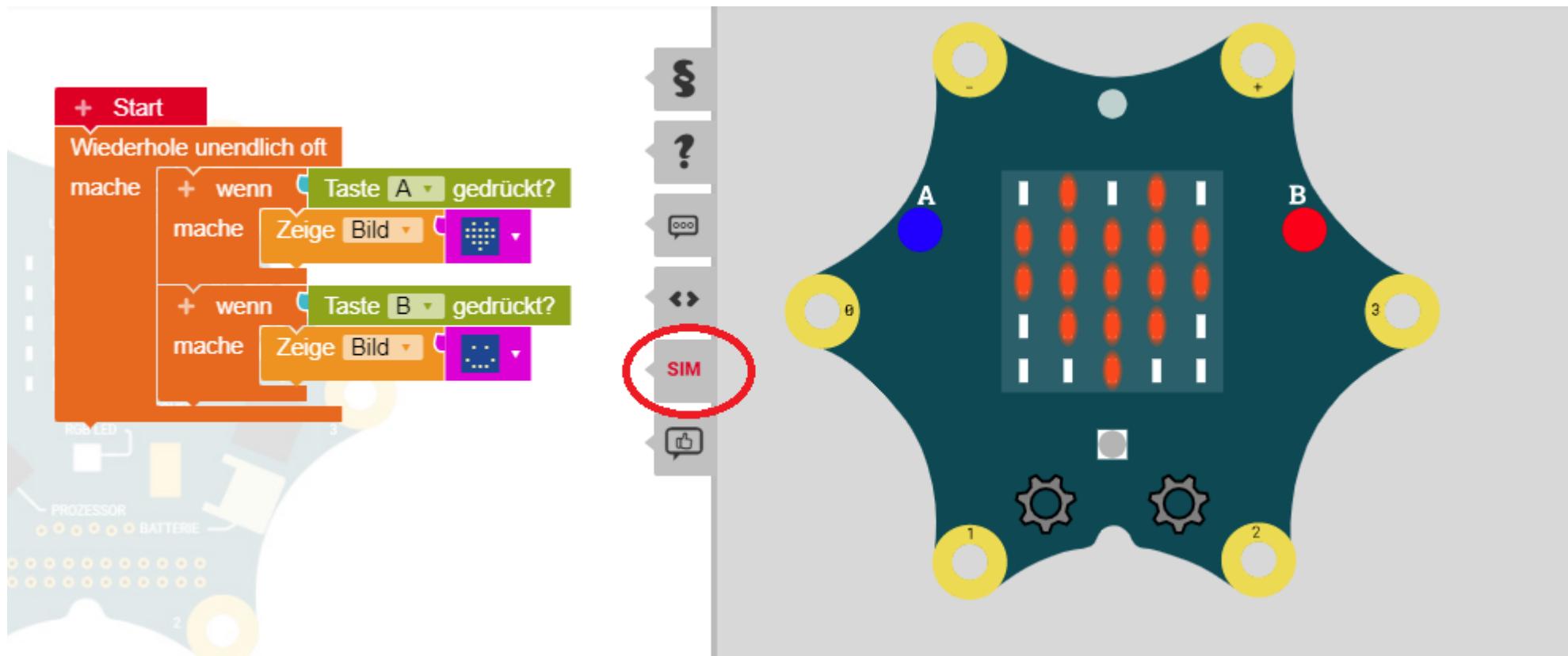


Mein erstes Programm – Online-Editoren:

<https://lab.open-roberta.org/>

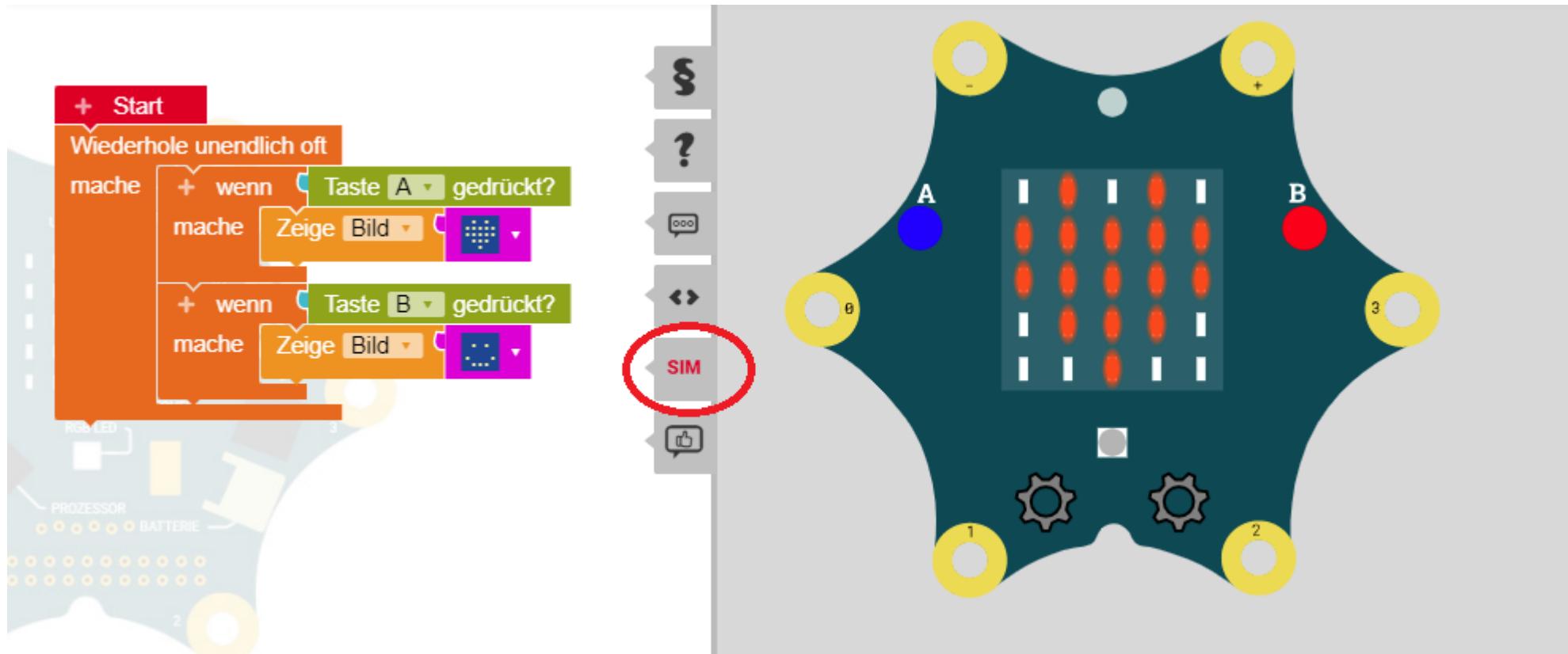
Calliope Workshop

1 Mein erstes Programm – Taste A zeigt uns ein Herz, Taste B zeigt uns einen Smiley



Calliope Workshop

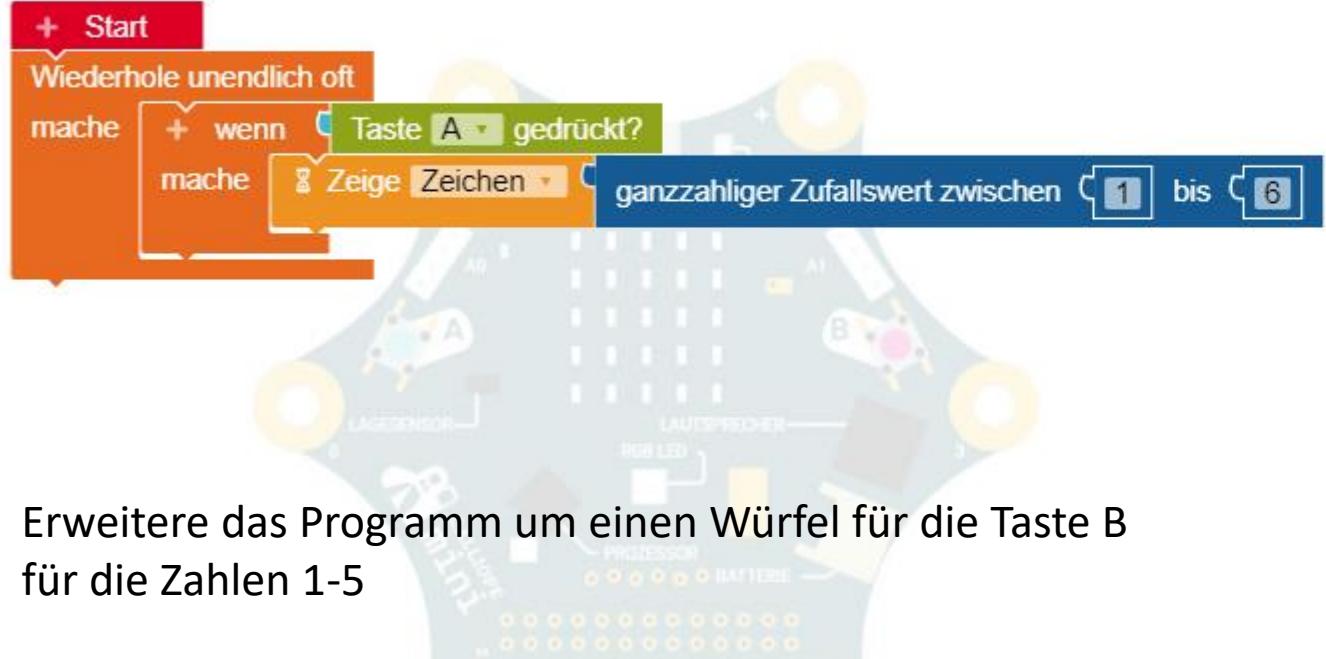
1 Mein erstes Programm – Taste A zeigt uns ein Herz, Taste B zeigt uns einen Smiley



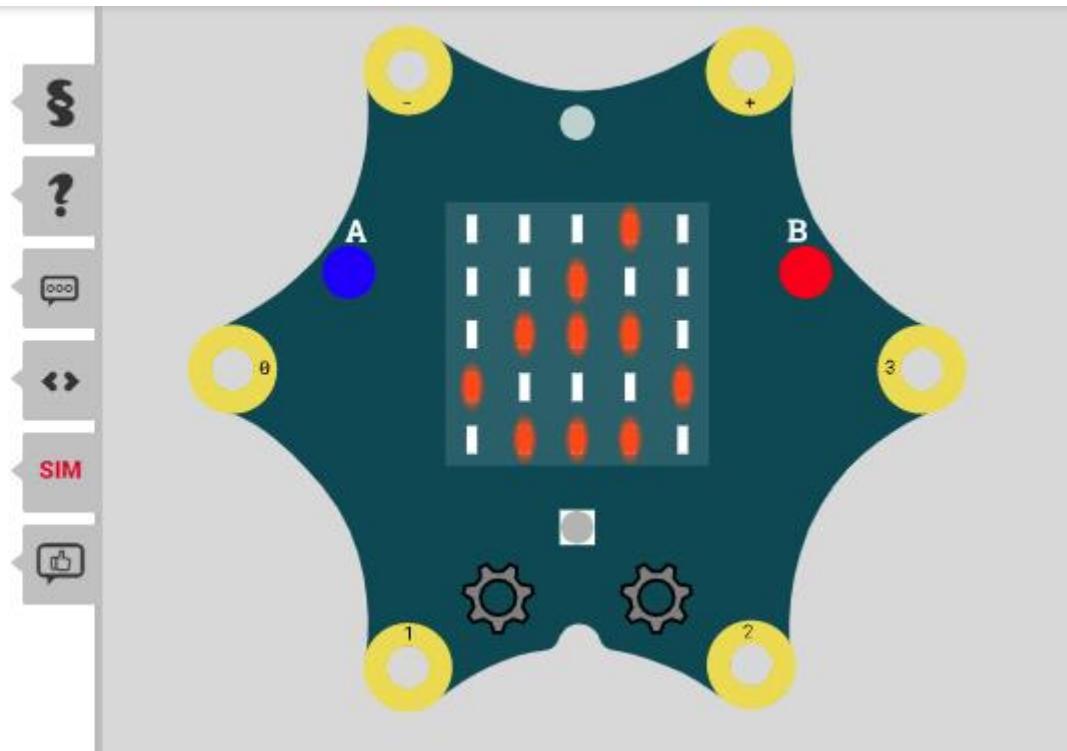
Calliope Workshop



2 Würfel - Taste A zeigt Zahlen von 1-6



Erweitere das Programm um einen Würfel für die Taste B
für die Zahlen 1-5



Calliope Workshop



Bei Gruppe anmelden

The screenshot shows the Calliope software interface. On the left, there is a vertical menu bar with colored buttons for different programs: Aktion (orange), Sensoren (green), Kontrolle (red), Logik (blue), Mathematik (dark blue), Text (light green), Farben (yellow), Bilder (pink), and Variablen (purple). Above this menu, there are four small icons: a document, a person, a lightbulb, and a square with dots. To the right of these icons is a button labeled "Benutzer" with a user icon. A dropdown menu is open from this button, containing the following options: "anmelden ...", "Bei Gruppe anmelden ...", "abmelden", "Gruppen verwalten ...", "ändern ...", "Benutzer löschen ...", and "Statusinformationen". The main workspace area has a title "CALLIOPEbasic" and some placeholder text "...". Overlaid on the workspace is a modal dialog titled "Bei Gruppe anmelden". This dialog contains several input fields:

- Benutzername des Gruppeneigentümers: A field with a graduation cap icon containing the text "franky".
- Name der Gruppe: A field with a group icon containing the text "Oberlab".
- Benutzername: A field with a person icon containing the text "Julia".
- Passwort: A field with a lock icon containing the text "Oberlab:Julia".

At the bottom of the dialog are two buttons: "OK" and "Passwort zurücksetzen ...".

ändere Passwort ...

Das Passwort deines Benutzerkontos ist noch nicht sicher. Setze dein eigenes Passwort, damit nur du dich bei deinem Konto anmelden kannst.

Neues Passwort

Passwort wiederholen

OK **Abbrechen**

Calliope Workshop

Programme speichern



Calliope Workshop

Die Online-Hilfe nutzen:

rechte Maustaste



»Wiederhole unendlich oft«

Mit dem Block »Wiederhole unendlich oft« kannst du Blöcke endlos von deinem Roboter ausführen lassen. Alle Blöcke, die sich im »Wiederhole unendlich oft«-Block befinden, werden immer wieder ausgeführt. Die Blöcke werden dabei von oben nach unten ausgeführt. Sobald der letzte Block ausgeführt wurde, beginnt das Programm wieder mit dem ersten Block. Deshalb wird dieser Block auch »Endlosschleife« genannt.

Eingaben:

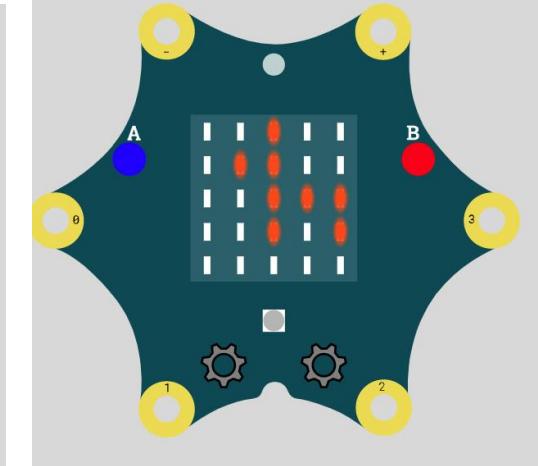
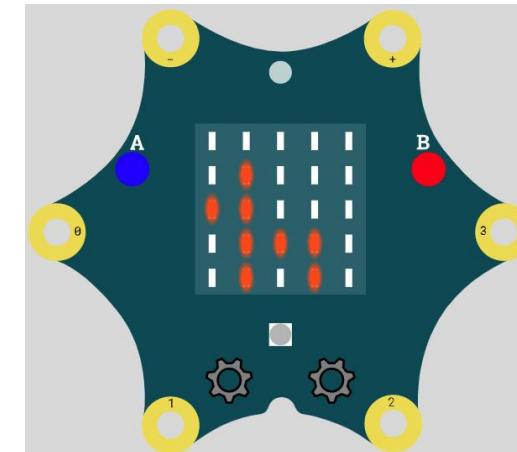
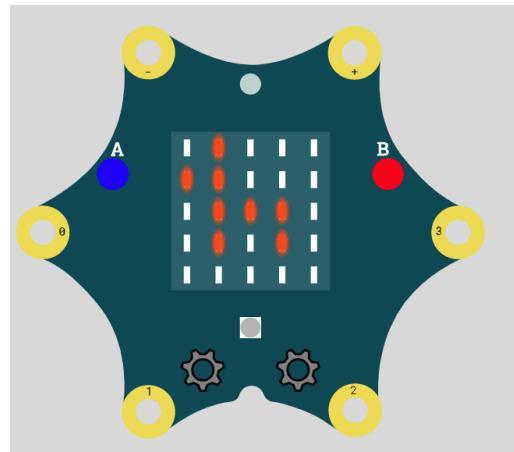
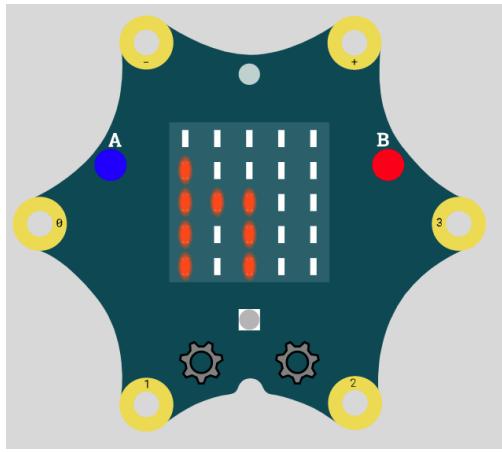
→ Blöcke, die wiederholt ausgeführt werden.

»Wiederhole n mal«

Mit dem Block »Wiederhole n mal« kannst du andere Blöcke beliebig oft von deinem Roboter ausführen lassen. Alle Blöcke, die sich im Block »Wiederhole n mal« befinden,

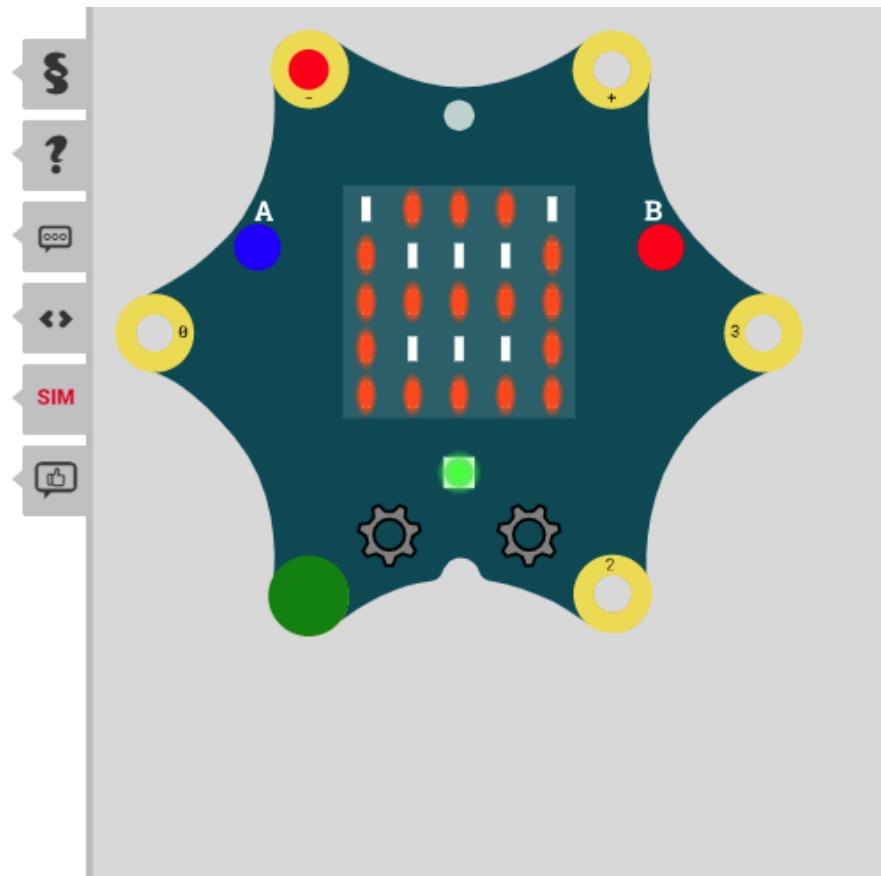
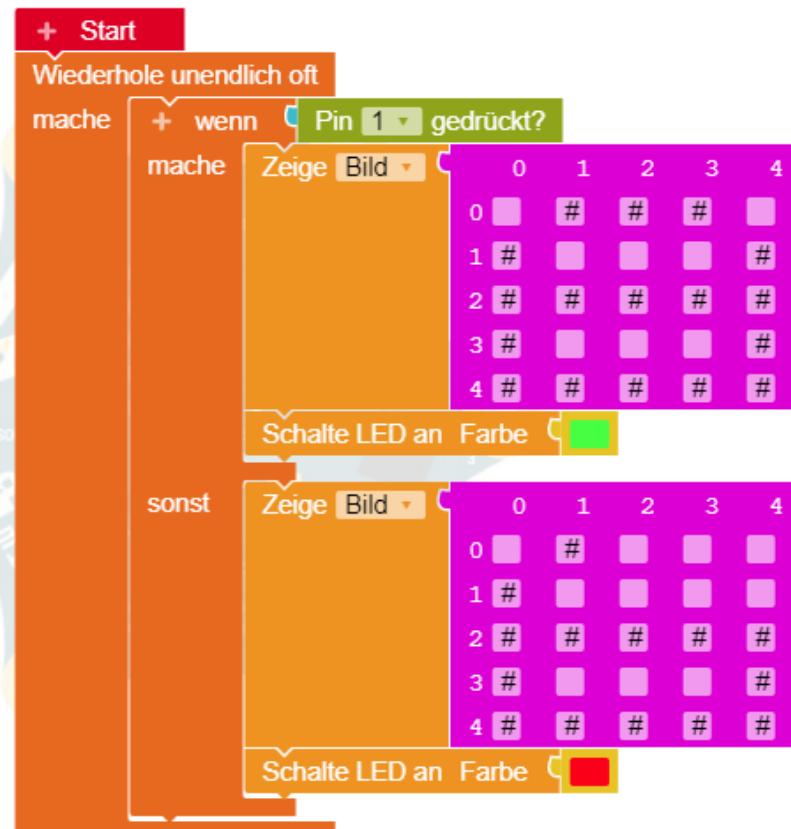
Calliope Workshop

3 Aufgabe Mit Open Roberta programmieren wir ein Daumenkino und lassen einen Hund mit 8 Bildern von rechts nach links springen.



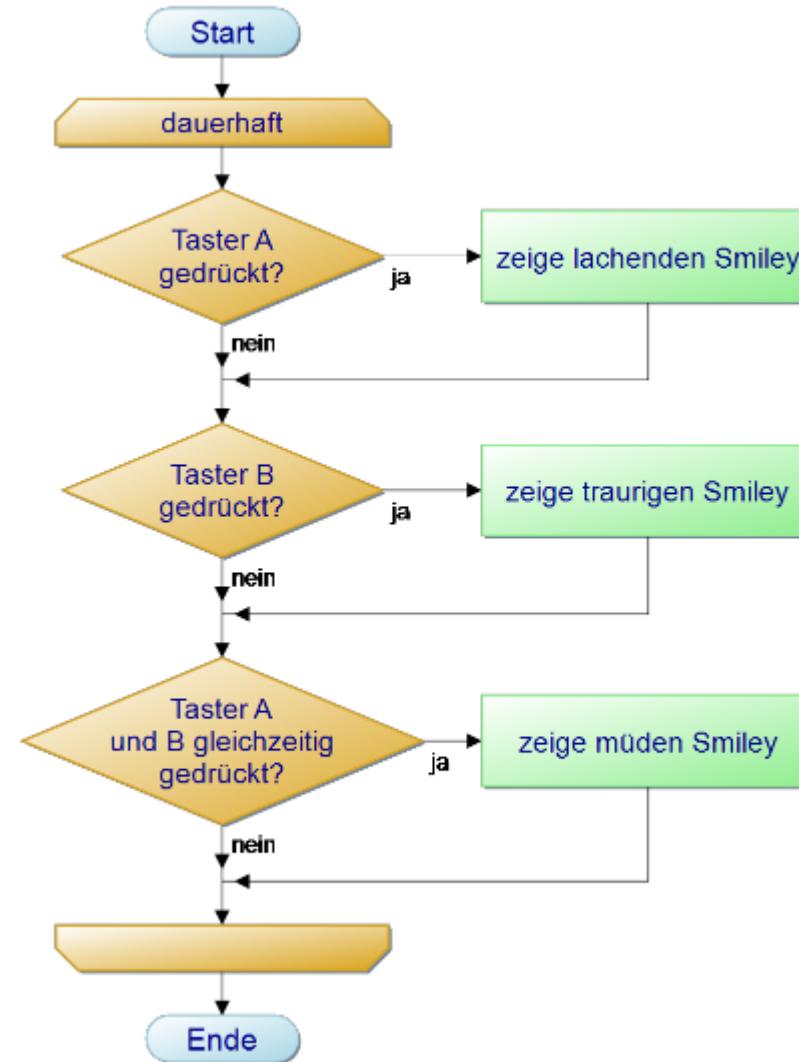
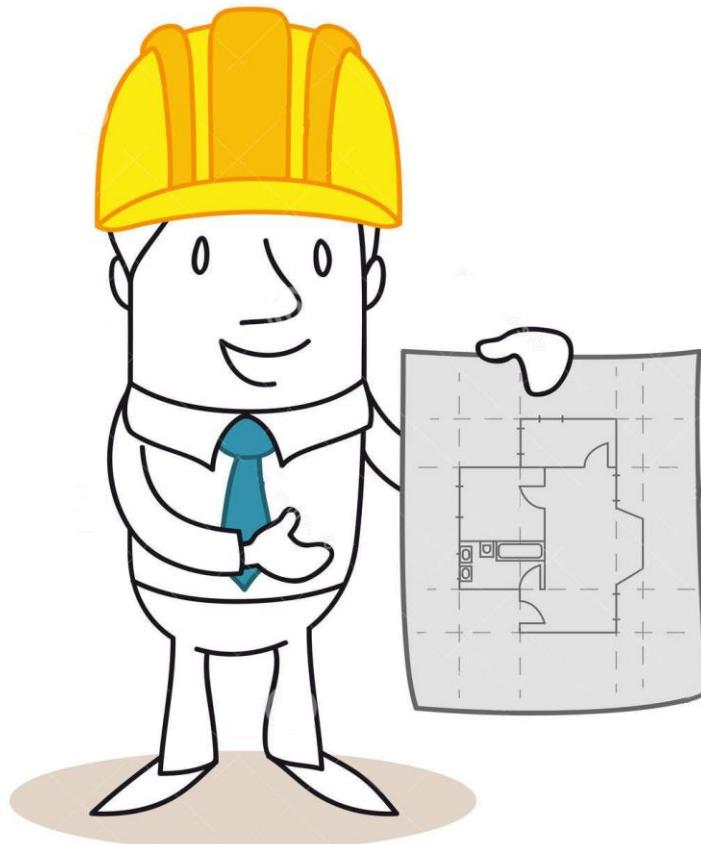
Calliope Workshop

4 Keksdose offen? - Anzeige ob Kontakt (PIN P1) geschlossen oder offen ist
PIN P1 gedrückt: Anzeige Schloss geschlossen und LED=grün; ansonsten Symbols Schloss offen und LED=rot.



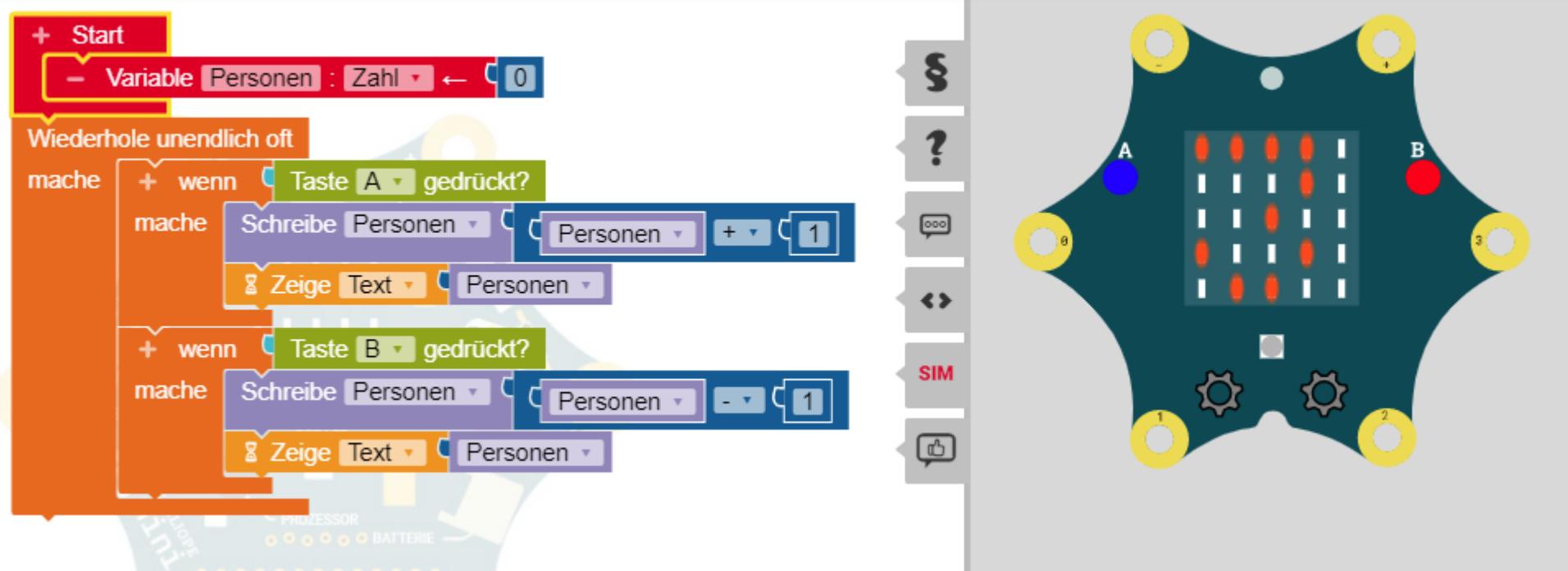
Calliope Workshop

Einen Plan haben



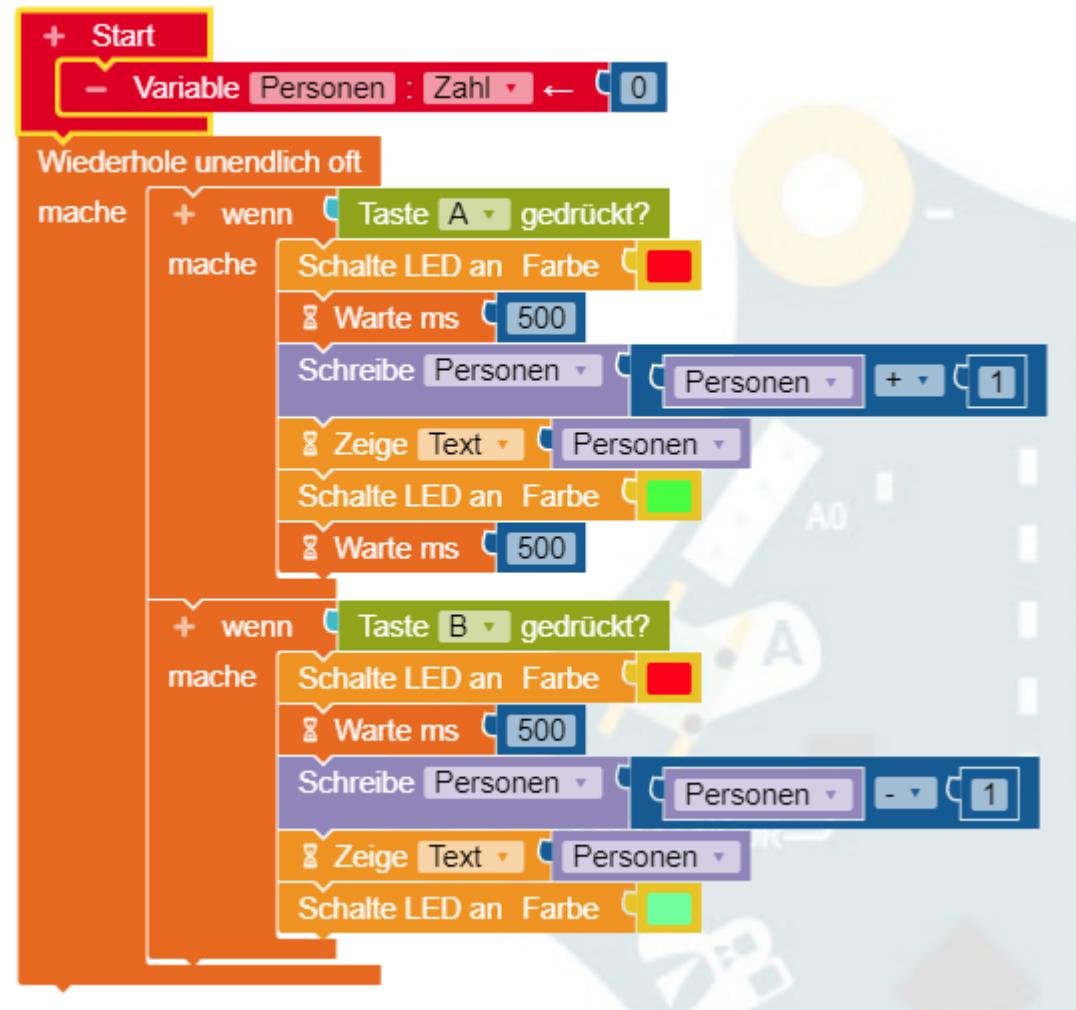
Calliope Workshop

5 Personenzähler - Beim Eintreten in ein Freibad (Taste A - Drehkreuz) wird ein Zähler um +1 erhöht, beim Austritt (Taste B - Drehkreuz) wird der Zähler um 1 subtrahiert. Zwischen den Zählvorgängen soll 1 sec Pause liegen und in der Zeit eine Ampel von grün auf rot schalten.



Calliope Workshop

5 Personenzähler - mit LEDs:



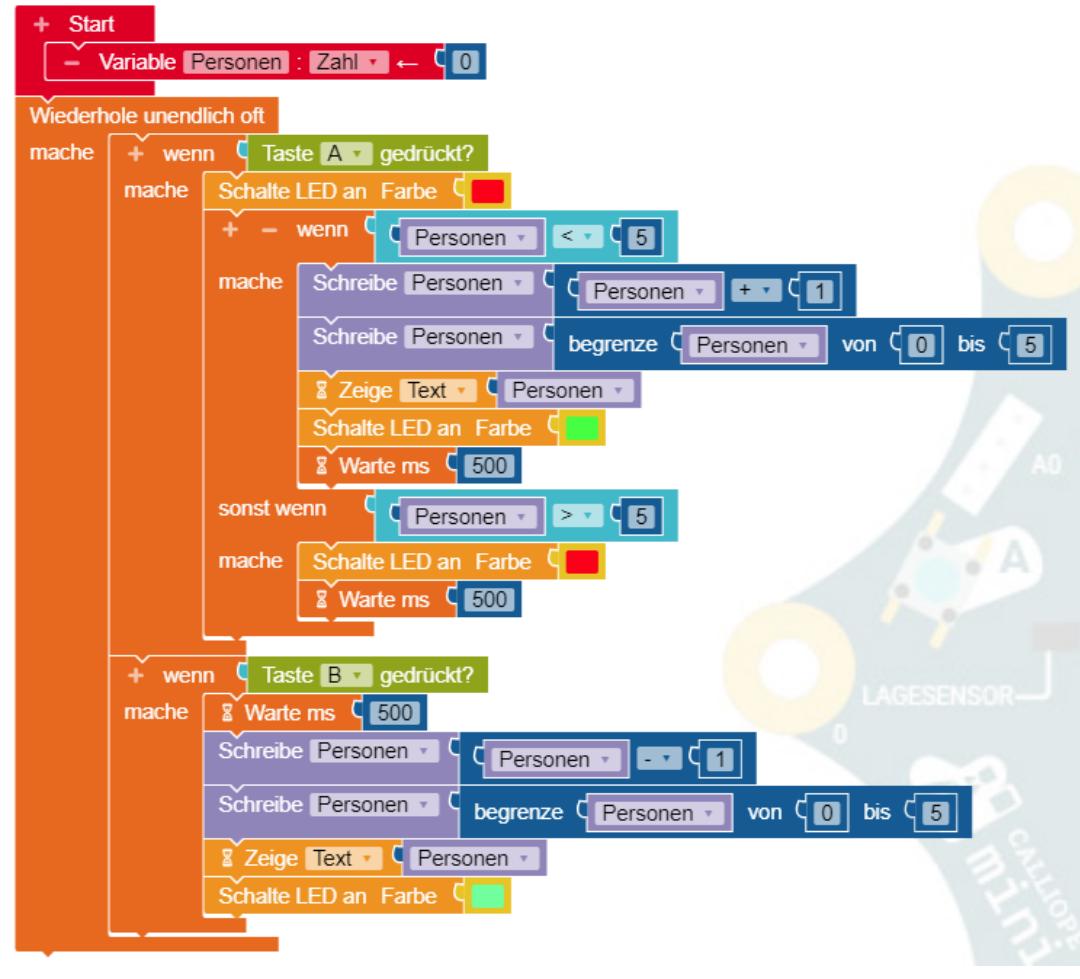
Calliope Workshop

6 Personenzähler – Erweiterung 1 Beim vorigen Personenzähler ist es möglich, dass eine Minus- Anzeige zustande kommt, wenn beim Zählerstand Null die Taste B noch einmal gedrückt wird. Ändere das Programm ab, dass eine Anzeige unter Null nicht möglich ist.



Calliope Workshop

7 Personenzähler – Erweiterung 2: Wird eine Höchstzahl von Besuchern (5) überschritten, soll der Eingang gesperrt werden und die rote LED leuchten.



Calliope Workshop



8 Zufallsspiel: Schere, Stein, Papier - Programmiere das Schere, Stein, Papier Spiel



Calliope Workshop



Fehlersuche: Der Debug-Modus

The image shows a Scratch script on the left and a Calliope microcontroller board diagram on the right.

Scratch Script:

- + Start
- Wiederhole unendlich oft
- mache + wenn Taste A gedrückt?
- mache + wenn ganzzahliger Zufallswert zwischen 1 bis 3 = 1
- mache Zeige Bild [0 1 2 3 4]
- mache Zeige Bild [0 1 2 3 4]
- + wenn ganzzahliger Zufallswert zwischen 1 bis 3 = 2
- mache Zeige Bild [0 1 2 3 4]
- + wenn ganzzahliger Zufallswert zwischen 1 bis 3 = 3
- mache Zeige Bild [0 1 2 3 4]

Calliope Board Diagram:

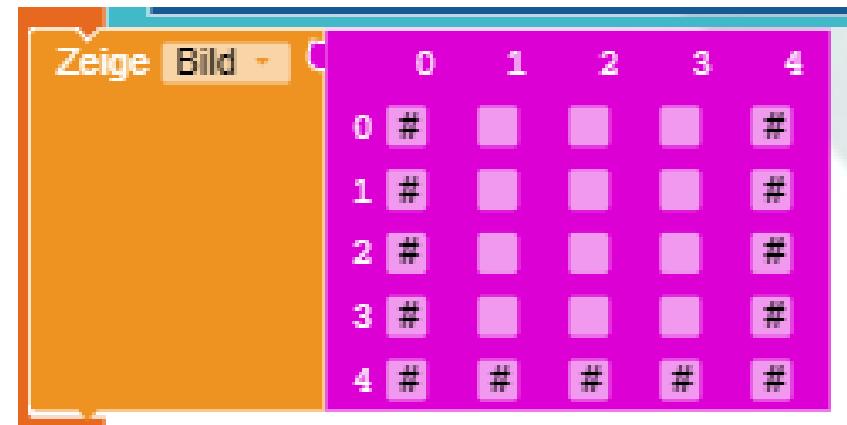
The diagram illustrates the Calliope board with various components and connection points. Key features include:

- A central 5x5 grid of red LED lights.
- Four yellow circular pads labeled 0, 1, 2, and 3, each with a corresponding digital pin number (0, 1, 2, 3) nearby.
- Two circular pads labeled A and B, each with a corresponding analog pin number (A0, A1) nearby.
- Four grey circular pads labeled -, +, 1, and 2, each with a corresponding analog pin number (-, +, 1, 2) nearby.
- Two small grey circles representing capacitive touch pads.
- Two grey gears representing motor drivers.

The bottom toolbar includes icons for play, cloud, search, and trash, with the key icon circled in red.

Calliope Workshop

9 Zufallsspiel: Schere, Stein, Papier - füge noch einen Brunnen hinzu



Calliope Workshop



Calliope C++ Editor und Compiler mit Open Roberta:

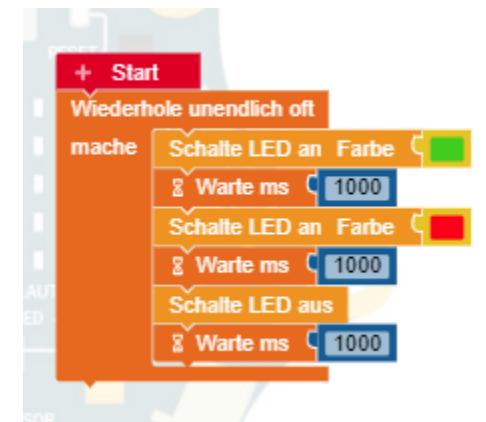
<https://lab.open-roberta.org/>

The screenshot shows the Open Roberta C++ Editor interface. At the top, there are several icons: a document, a gear, a lightbulb, a person, a house, a square grid, and a circular arrow. To the right of these are four colored squares (blue, cyan, green, orange). Below the icons, the text "PROGRAMM Sens_Zeitgeber" and "ROBOTERKONFIGURATION" is visible. The main area contains a code editor with the following C++ code:

```
1. 
2. #include "MicroBit.h"
3. #include "NEPODefs.h"
4. #include <list>
5. #include <array>
6. #include <stdlib.h>
7. MicroBit _uBit;
8.
9. int _initTime = _uBit.systemTime();
10.
11.
12.
13. int main()
14. {
15.     _uBit.init();
16.
17.     while ( true ) {
18.         _uBit.rgb.setColour(MicroBitColor(51, 204, 0, 255));
19.         _uBit.sleep(1000);
20.         _uBit.rgb.setColour(MicroBitColor(255, 0, 0, 255));
21.         _uBit.sleep(1000);
22.         _uBit.rgb.off();
23.         _uBit.sleep(1000);
24.         _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
25.     }
26.     release_fiber();
27. }
```

To the right of the code, there are four buttons with labels: "HEX-Datei erzeugen", "Quellcode erzeugen", "Quellcode auf PC laden", and "CPP-Datei hochladen". Below these buttons is a button labeled "Importiere aktuelles Programm". At the bottom right of the editor window, there is a vertical stack of colored squares (blue, cyan, green, orange) matching the ones at the top.

C++ Beispiel
LED schalten im
Open Roberta Editor



Calliope Workshop



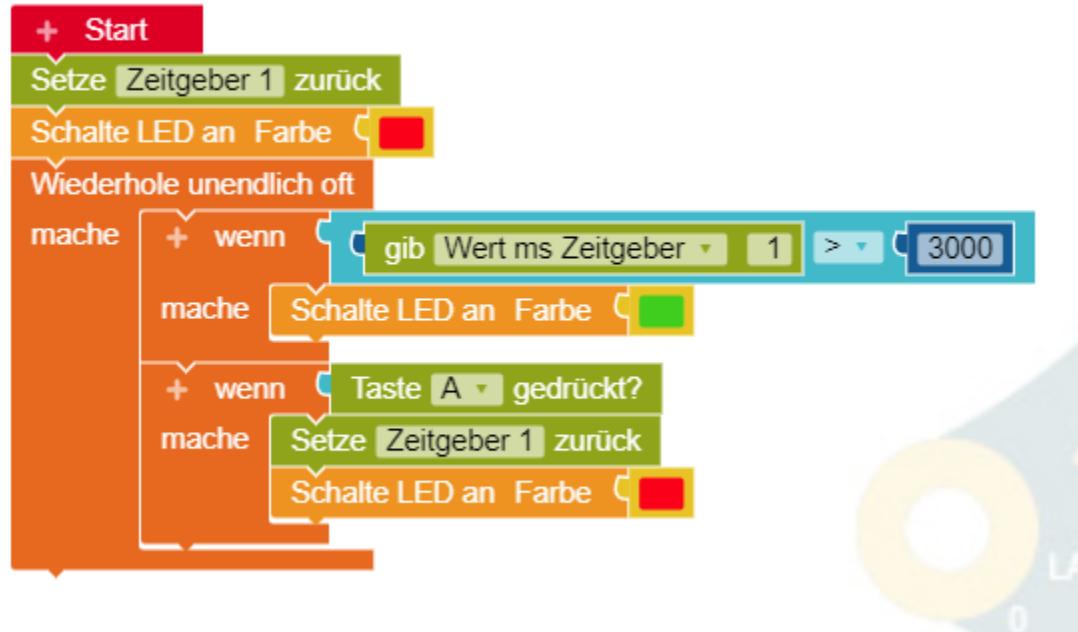
10 Aufgabe Händewaschtimer

Wir programmieren uns einen Zeitgeber. Beim Start des Calliope leuchtet die rote LED 30 Sekunden (zum testen 3 Sekunden), danach wird die LED grün. Mit der Taste A können wir den Zeitgeber neu starten. Löse die Aufgabe mit Open Roberta.



Calliope Workshop

10 Lösung Händewaschtimer



C++

```
#include "MicroBit.h"
#include "NEPODefs.h"
#include <list>
#include <array>
#include <stdlib.h>
MicroBit_uBit;

int _initTime = _uBit.systemTime();

int main()
{
    _uBit.init();

    _initTime = _uBit.systemTime();
    _uBit.rgb.setColour(MicroBitColor(255, 0, 255));
    while ( true ) {
        if ( ( _uBit.systemTime() - _initTime ) > 3000 ) {
            _uBit.rgb.setColour(MicroBitColor(51, 204, 0, 255));
        }
        if ( _uBit.buttonA.isPressed() ) {
            _initTime = _uBit.systemTime();
            _uBit.rgb.setColour(MicroBitColor(255, 0, 0, 255));
        }
        _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
    }
    release_fiber();
}
```

Calliope Workshop



11 Aufgabe Automatisches Rollerlicht

Wir programmieren uns ein automatisches Rollerlicht. Wenn es dunkel wird und der Lichtwert unter 30% sinkt, soll der Calliope automatisch das Licht einschalten. Löse die Aufgabe mit Open Roberta.



Calliope Workshop

11 Lösung Automatisches Rollerlicht



C++

```
#include "MicroBit.h"
#include "NEPODefs.h"
#include <list>
#include <array>
#include <stdlib.h>
MicroBit_uBit;

int main()
{
    _uBit.init();

    while ( true ) {
        if ( round(_uBit.display.readLightLevel() * _GET_LIGHTLEVEL_MULTIPLIER) < 30 ) {
            _uBit.rgb.setColour(MicroBitColor(255, 255, 255, 255));
        } else {
            _uBit.rgb.setColour(MicroBitColor(0, 0, 0, 255));
        }
        _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
    }
    release_fiber();
}
```

Calliope Workshop



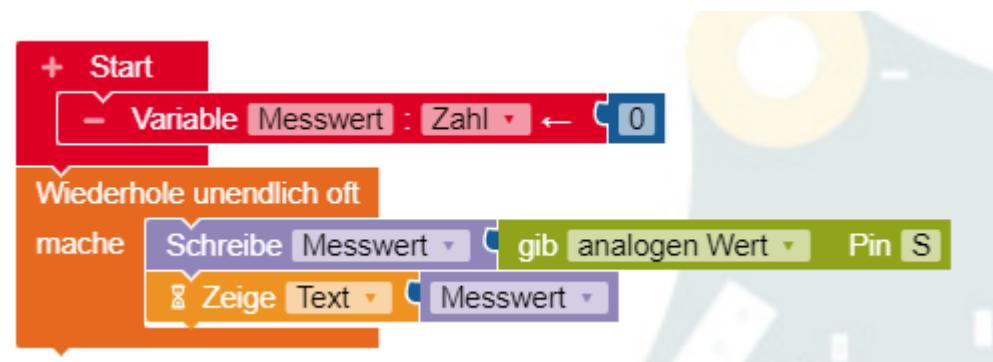
12 Aufgabe Luftqualität-Anzeige

Mit einem Open Roberta Programm soll die Luftqualität vom Sensoreingang A1 (Pin C16) mit einem Grove Luftqualitätssensor angezeigt werden. Die Anzeige erfolgt über die LED-Matrix.



Calliope Workshop

12 Lösung Luftqualität-Anzeige



Calliope Workshop



13 Aufgabe Temperatur-Anzeige

Mit einem Open Roberta Programm soll die Temperatur vom Sensoreingang A1 (Pin C16) mit einem Grove Temperatursensor angezeigt werden. Die Anzeige erfolgt über die LED-Matrix.

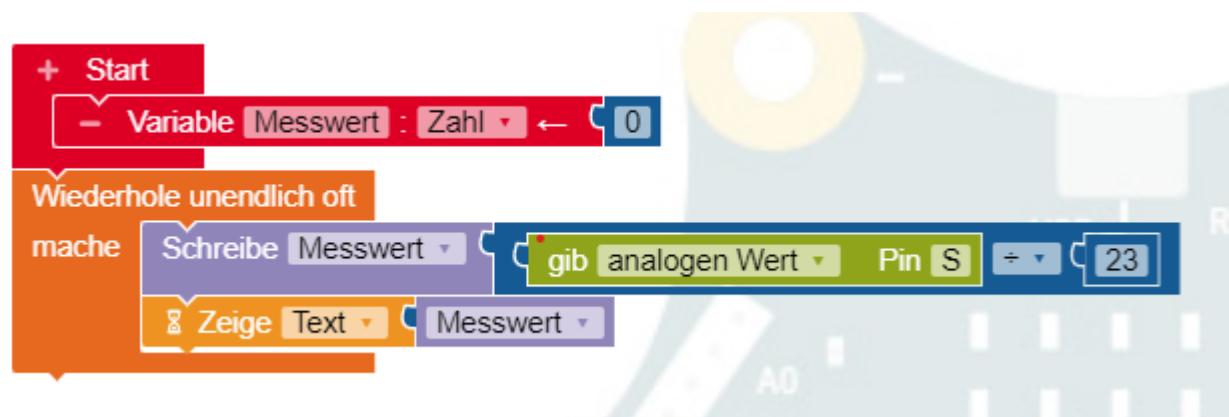


A screenshot of an Open Roberta block diagram. It shows a purple "Schreibe" block connected to a green "gib analogen Wert" block, which is then connected to a green "Pin S" block. The "gib" block has a dropdown menu showing "analogen Wert".

Calliope Workshop



13 Lösung Temperatur-Anzeige



Calliope Workshop

14 Aufgabe

Programmiere eine Raumluftüberwachung für einen Sensor am A1

Mit einem Open Roberta Programm soll eine Raumluftüberwachung vom Sensoreingang A1 (Pin16) erfolgen. Als Sensor dient zur Simulation ein Poti. Der Sensorwert soll uns 3 Werte anzeigen.

Der eingelesenen Sensorwert soll uns 3 Werte anzeigen:

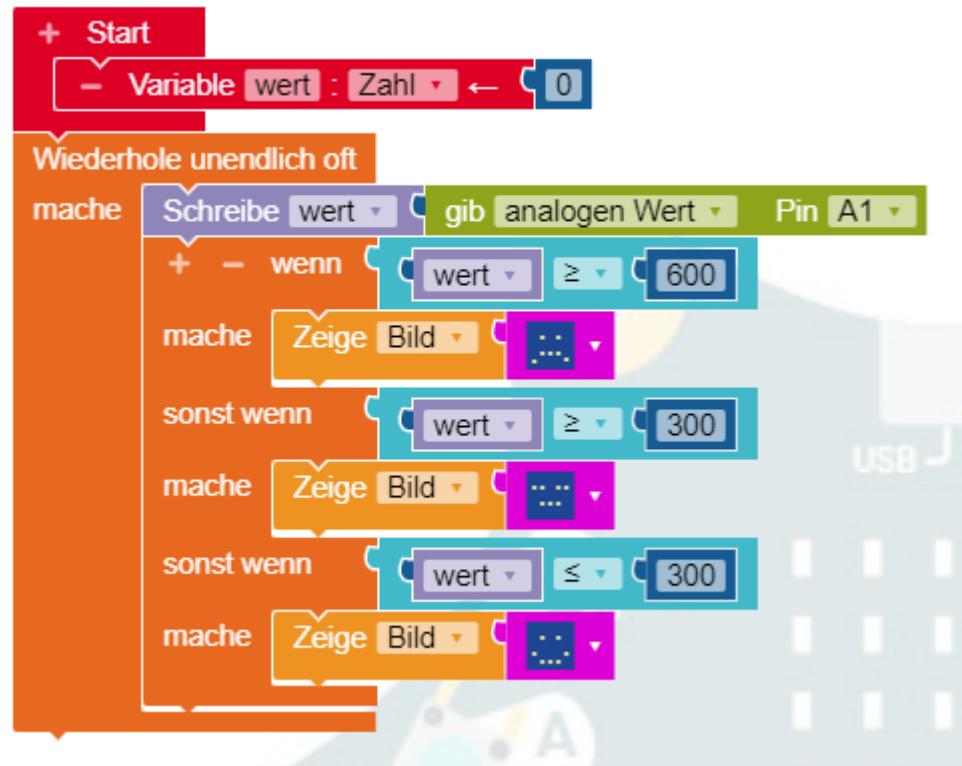
000-300:		<table border="1"><thead><tr><th>Bit</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th></tr></thead><tbody><tr><td>Wertigkeit</td><td>2^{10}</td><td>2^9</td><td>2^8</td><td>2^7</td><td>2^6</td><td>2^5</td><td>2^4</td><td>2^3</td><td>2^2</td><td>2^1</td><td>2^0</td></tr><tr><td>Dezimal</td><td>1024</td><td>512</td><td>256</td><td>128</td><td>64</td><td>32</td><td>16</td><td>8</td><td>4</td><td>2</td><td>1</td></tr></tbody></table>	Bit	11	10	9	8	7	6	5	4	3	2	1	Wertigkeit	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Dezimal	1024	512	256	128	64	32	16	8	4	2	1
Bit	11	10	9	8	7	6	5	4	3	2	1																											
Wertigkeit	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0																											
Dezimal	1024	512	256	128	64	32	16	8	4	2	1																											
301-600:																																						
600-1024:																																						

Bits und ihre Wertigkeit



Calliope Workshop

14 Lösung Programmiere eine Raumluftüberwachung für einen Sensor am A1

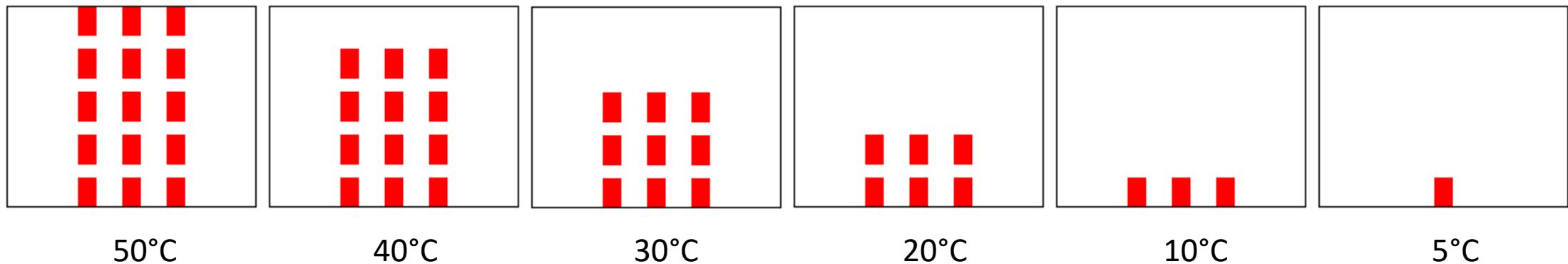


Calliope Workshop

15 Aufgabe Temperaturanzeige

Mit einem Open Roberta Programm soll die Temperatur vom Sensoreingang A1 (Pin C16) angezeigt werden. Als Temperatursensor dient zur Simulation ein Poti. Die Temperaturanzeige erfolgt über die LED-Matrix.

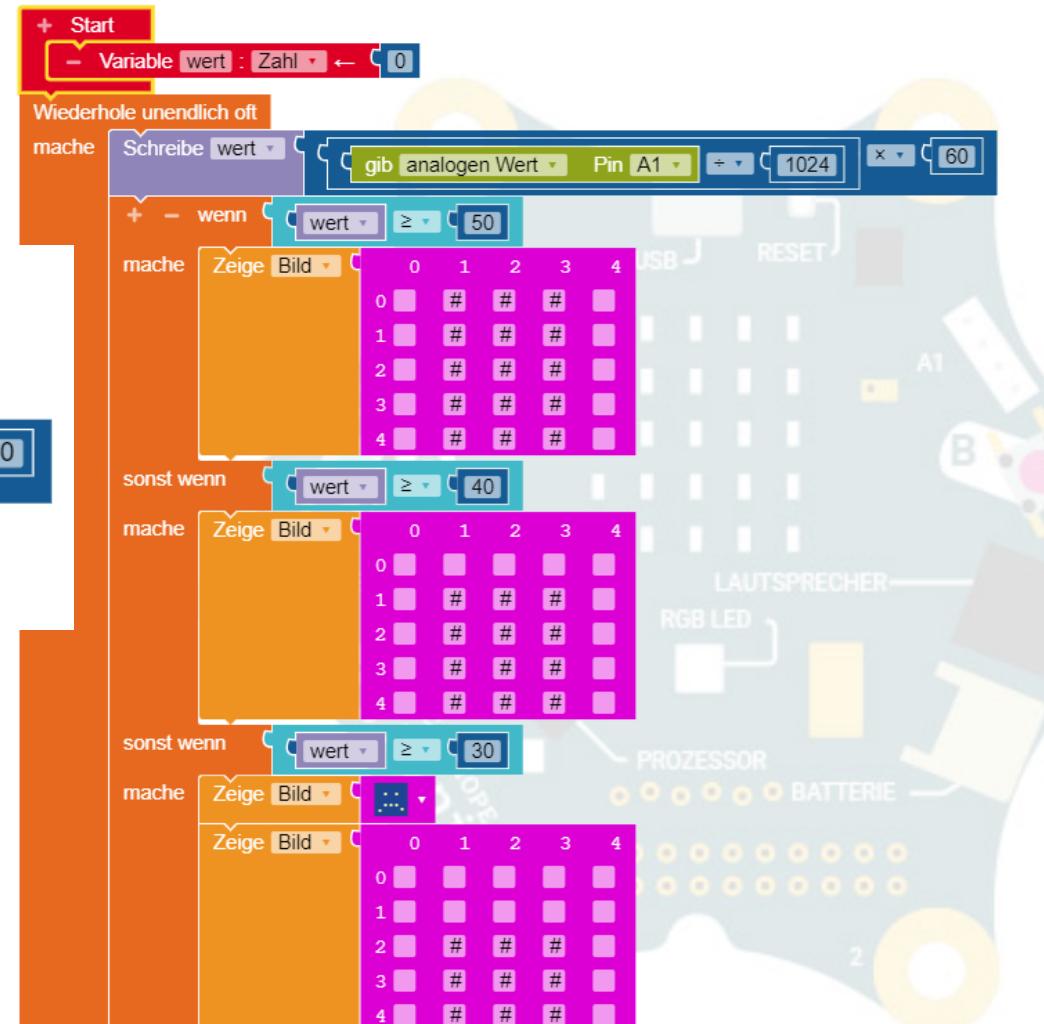
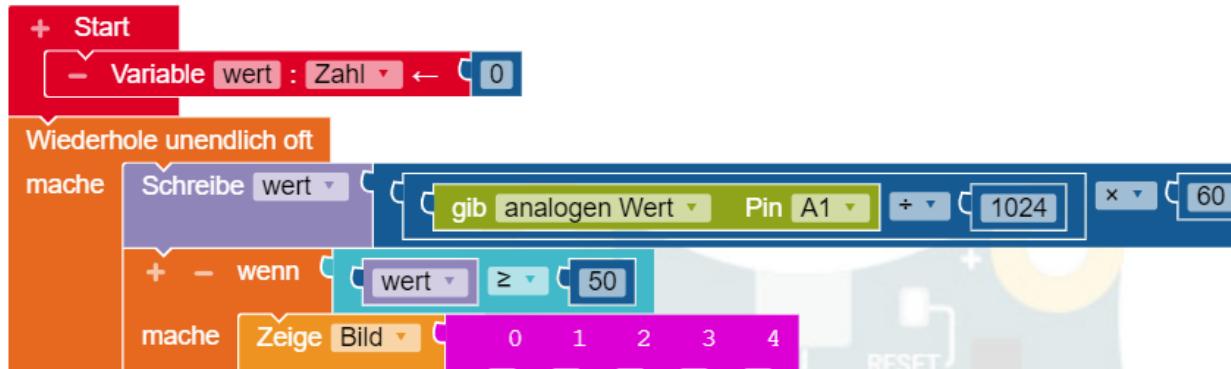
Anzeige:



Calliope Workshop



15 Lösung Temperaturanzeige



Calliope Workshop



16 Aufgabe Programmiere eine Raumluftüberwachung für einen Sensor am A1



Testprogramm: Poti (Raumluft-Sensor) von A1 (C16) einlesen.

Der eingelesenen Sensorwert soll uns 3 Werte anzeigen:

000-300:

301-600:

600-1024:

Bit	11	10	9	8	7	6	5	4	3	2	1
Wertigkeit	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Dezimal	1024	512	256	128	64	32	16	8	4	2	1

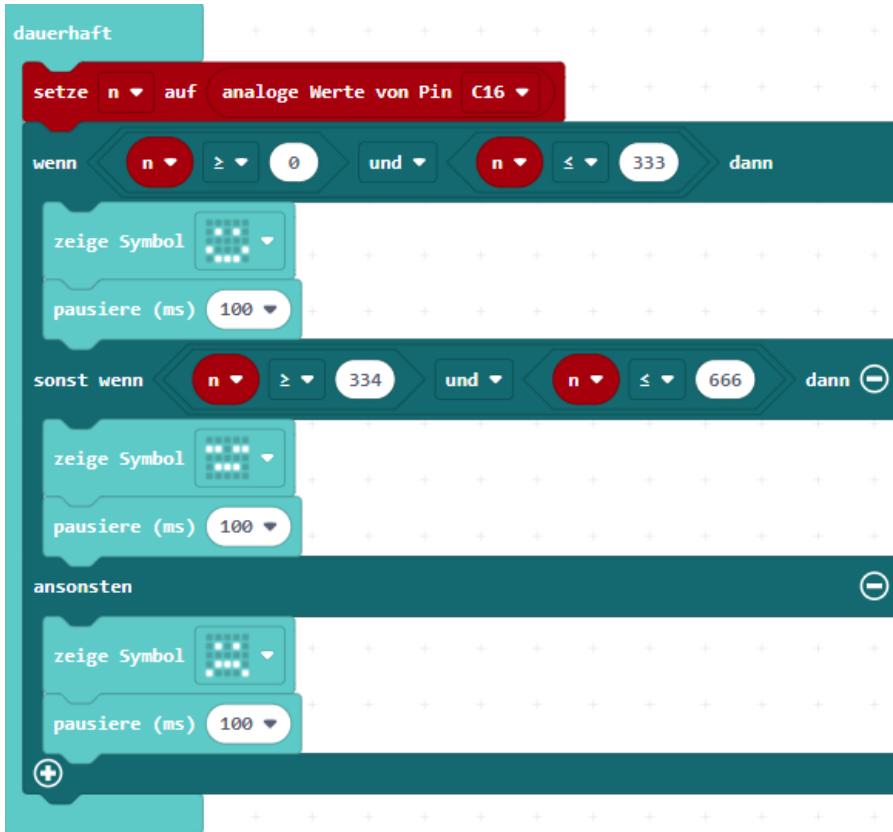
Bits und ihre Wertigkeit



Calliope Workshop



16 Lösung Raumluftüberwachung mit einem Poti an A1 simuliert



Java

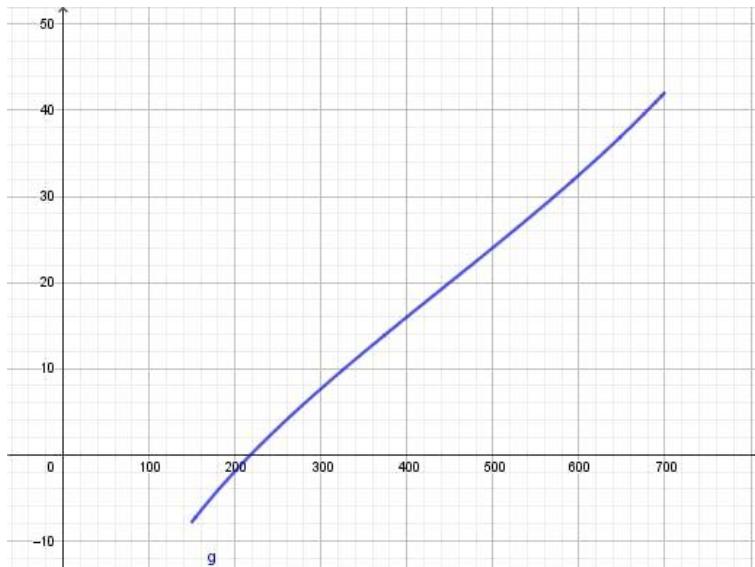
```
let n = 0  
basic.forever(function on_forever() {  
  
    n = pins.analogReadPin(AnalogPin.C16)  
    if (n >= 0 && n <= 333) {  
        basic.showIcon(IconNames.Happy)  
        basic.pause(100)  
    } else if (n >= 334 && n <= 666) {  
        basic.showIcon(IconNames.ASLEEP)  
        basic.pause(100)  
    }  
})
```

Python

```
n = 0  
def on_forever():  
    global n  
    n = pins.analog_read_pin(AnalogPin.C16)  
    if n >= 0 and n <= 333:  
        basic.show_icon(IconNames.HAPPY)  
        basic.pause(100)  
    elif n >= 334 and n <= 666:  
        basic.show_icon(IconNames.ASLEEP)  
        basic.pause(100)  
    else:  
        basic.show_icon(IconNames.SAD)  
        basic.pause(100)  
basic.forever(on_forever)
```

Calliope Workshop

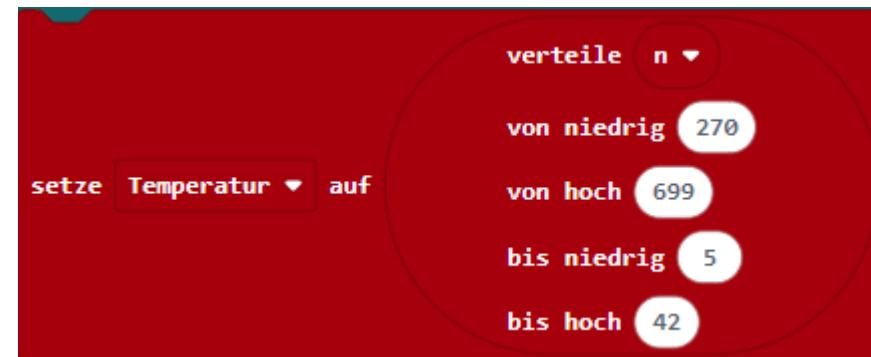
17 Aufgabe Programmiere ein Thermometer mit dem Grove Temperatur-Sensor



Kennlinie des Grove Temperatur-Sensor

Kennlinie: <https://calliope.cc/en/projekte/grove-temperatur-sensor-v1-2>

Der Grove Temperatur-Sensor kann über den Pin C16 eingelesen werden (Anschluss A1 - rechts)

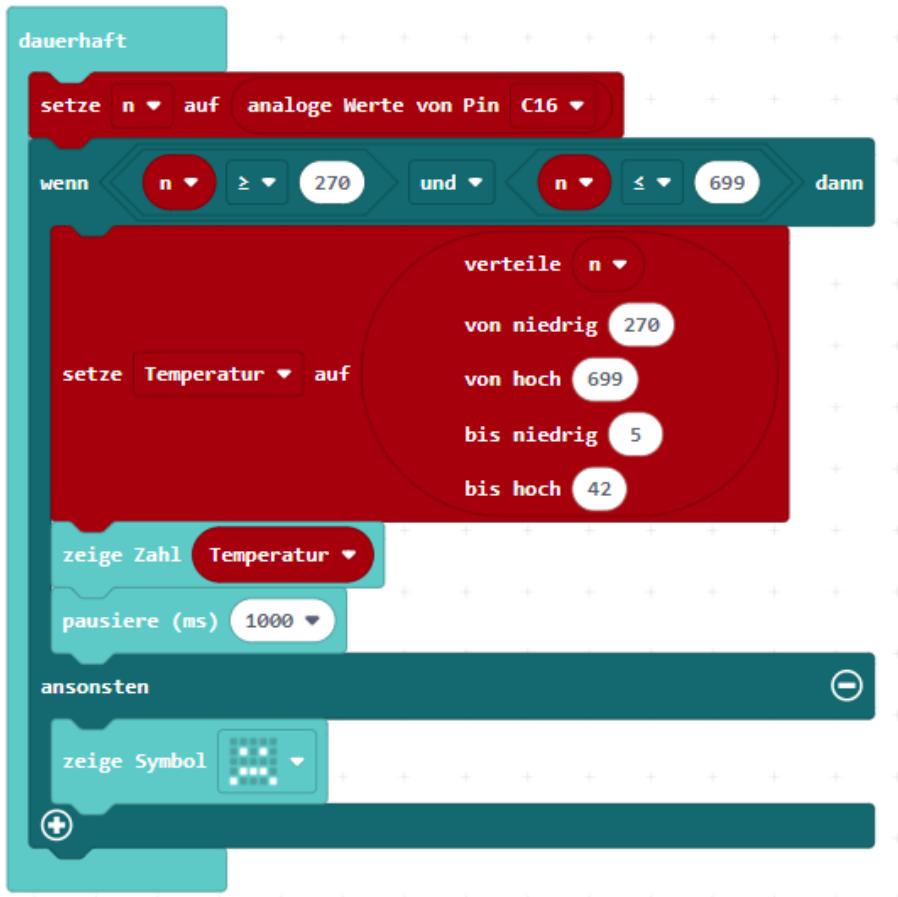


Verwende den MAP-Befehl (verteile-Befehl)

Calliope Workshop



17 Lösung Thermometer



Python

```
n = 0
Temperatur = 0

def on_forever():
    global n, Temperatur
    n = pins.analog_read_pin(AnalogPin.C16)
    if n >= 270 and n <= 699:
        Temperatur = pins.map(n, 270, 699, 5, 42)
        basic.show_number(Temperatur)
        basic.pause(1000)
    else:
        basic.show_icon(IconNames.SAD)
basic.forever(on_forever)
```

Java

```
let n = 0
let Temperatur = 0
basic.forever(function () {
    n = pins.analogReadPin(AnalogPin.C16)
    if (n >= 270 && n <= 699) {
        Temperatur = pins.map(
            n,
            270,
            699,
            5,
            42
        )
        basic.showNumber(Temperatur)
        basic.pause(1000)
    } else {
        basic.showIcon(IconNames.Sad)
    }
})
```

Calliope Workshop

18 Aufgabe Alarmanlage

Mit einem Open Roberta Programm soll eine Alarmanlage programmiert werden.

Der Calliope wird mit der X-Achse (parallel mit dem USB-Anschluss) auf der Türklinke befestigt.

Nach dem Einschalten leuchtet die gelbe LED. Wird die Türklinke betätigt und der Calliope neigt sich $> 15^\circ$, schaltet die LED für 1sec auf rot und ein Warnsignal ertönt. Danach schaltet die LED auf grün und zeigt an, dass ein Alarm ausgelöst wurde.



Calliope Workshop



18 Lösung Alarmanlage



C++

```
#include "MicroBit.h"
#include "NEPODefs.h"
#include <list>
#include <array>
#include <stdlib.h>
MicroBit _uBit;

int main()
{
    _uBit.init();

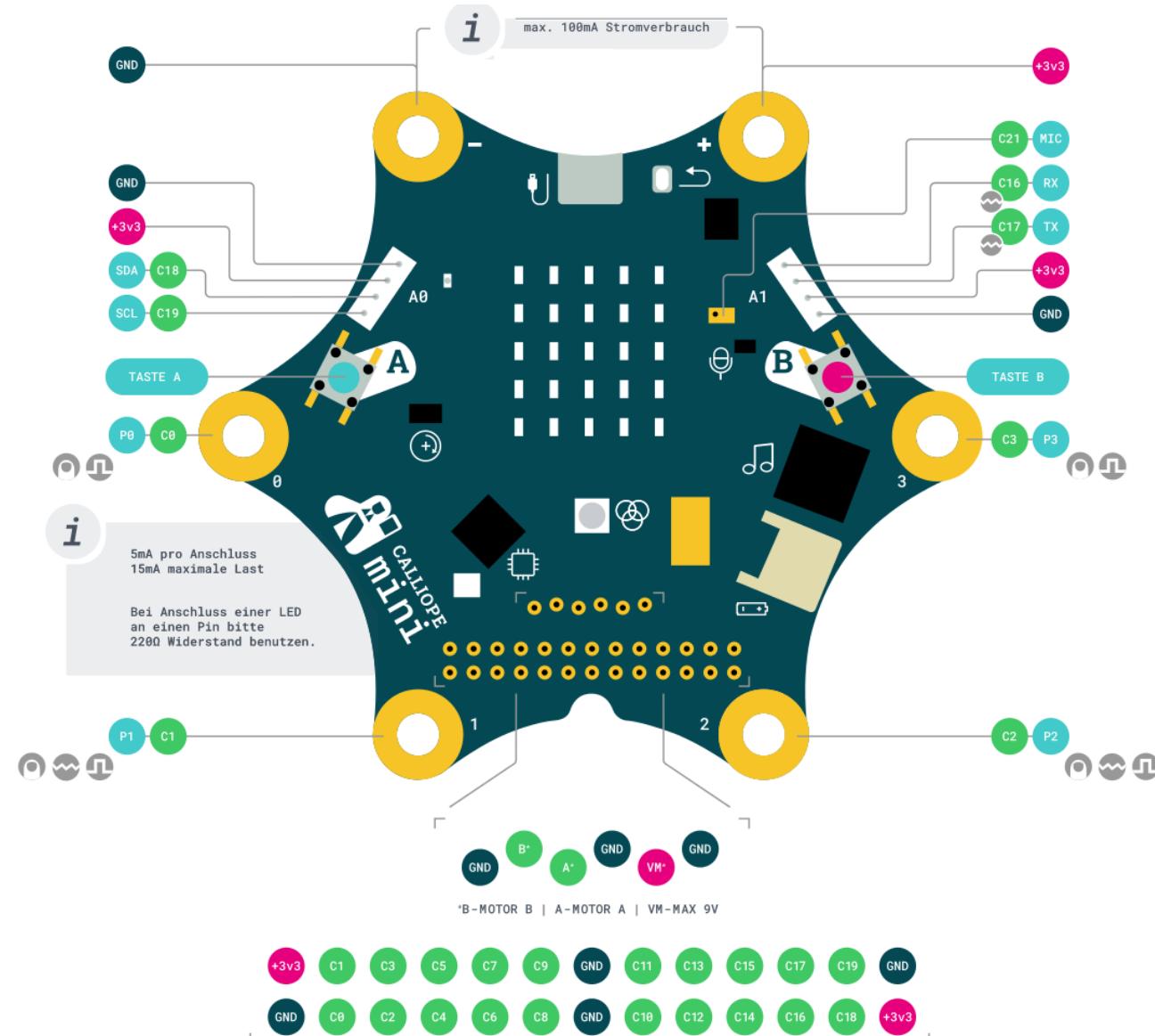
    _uBit.accelerometer.updateSample();
    _uBit.rgb.setColour(MicroBitColor(255, 255, 0, 255));
    while ( true ) {
        if ( _uBit.accelerometer.getPitch() > 15 ) {
            _uBit.rgb.setColour(MicroBitColor(255, 0, 0, 255));
            _uBit.soundmotor.soundOn(261.626); _uBit.sleep(2000);
        }
        _uBit.soundmotor.soundOff();
        _uBit.sleep(1000);
        _uBit.rgb.setColour(MicroBitColor(0, 153, 0, 255));
    }
    _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
}
release_fiber();
```

Calliope Workshop

Techn. Info

[v1.3 | Calliope Dokumentation \(calliope-mini.github.io\)](#)

[v2.0 | Calliope Dokumentation \(calliope-mini.github.io\)](#)



Calliope Workshop

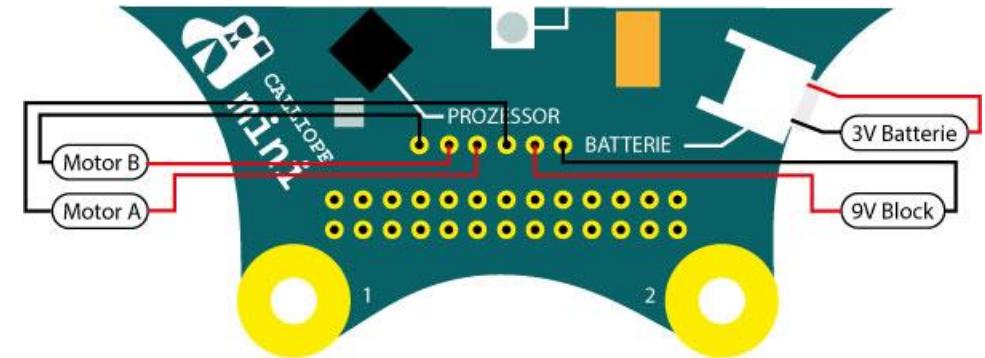
19 Aufgabe Helligkeitsabhängiger Lüfter

Mit einem Open Roberta Programm soll ein helligkeitsabhängiger Lüfter programmiert werden.

Der Motor ändert seine Geschwindigkeit in Abhängigkeit der Helligkeit in 2 Stufen:

Helligkeit < 30% < Motor-Tempo = 30%

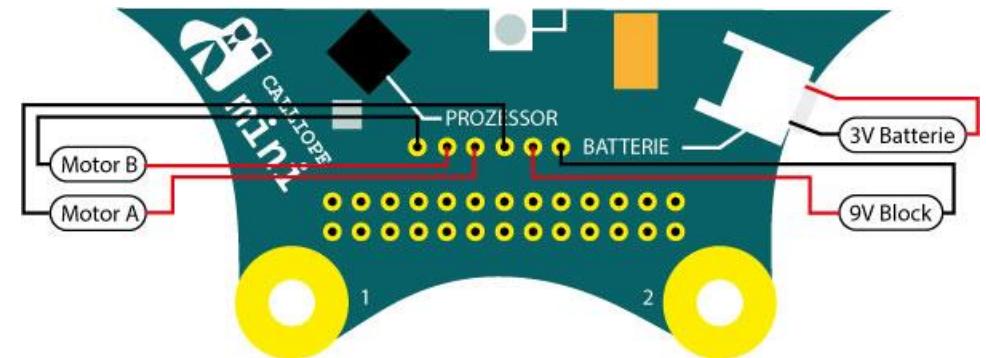
Helligkeit > 30% > Motor-Tempo = 10%



Calliope Workshop



19 Lösung Helligkeitsabhängiger Lüfter



Calliope Workshop



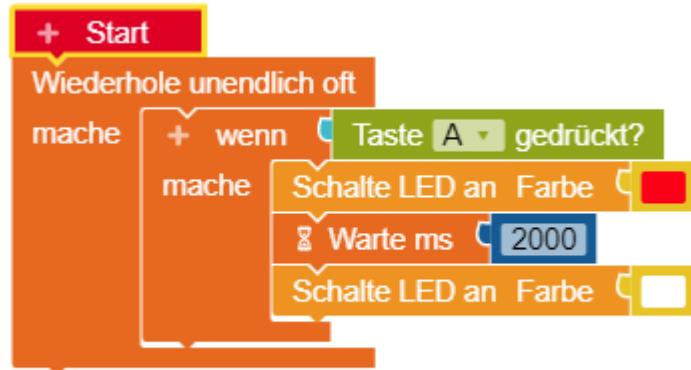
20 Aufgabe Treppenhauslicht

Mit der Taste A soll ein Treppenhauslicht realisiert werden. Nach Betätigung der Taste A schaltet die LED für 2s ein und danach wieder aus. Löse die Aufgabe mit Microsoft MakeCode.

Calliope Workshop



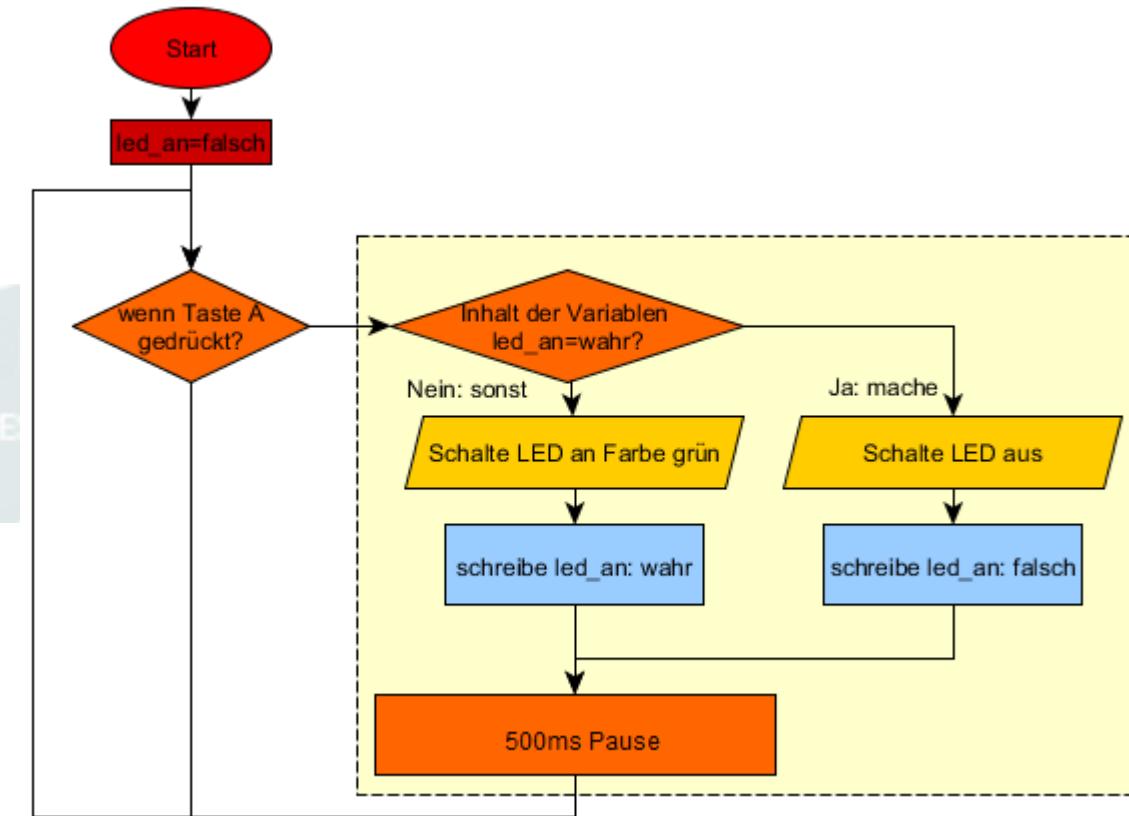
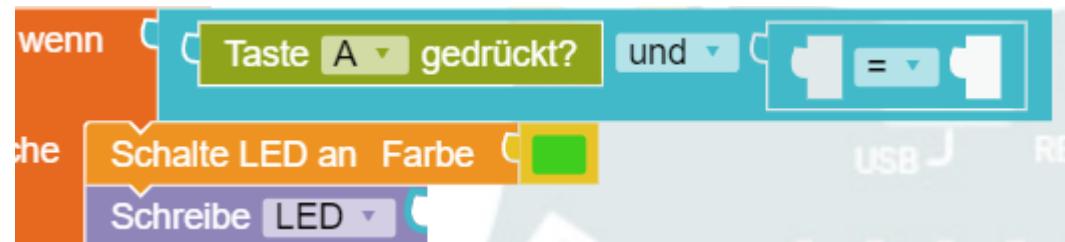
20 Lösung Treppenhauslicht



Calliope Workshop

21 Aufgabe Wechselschalter

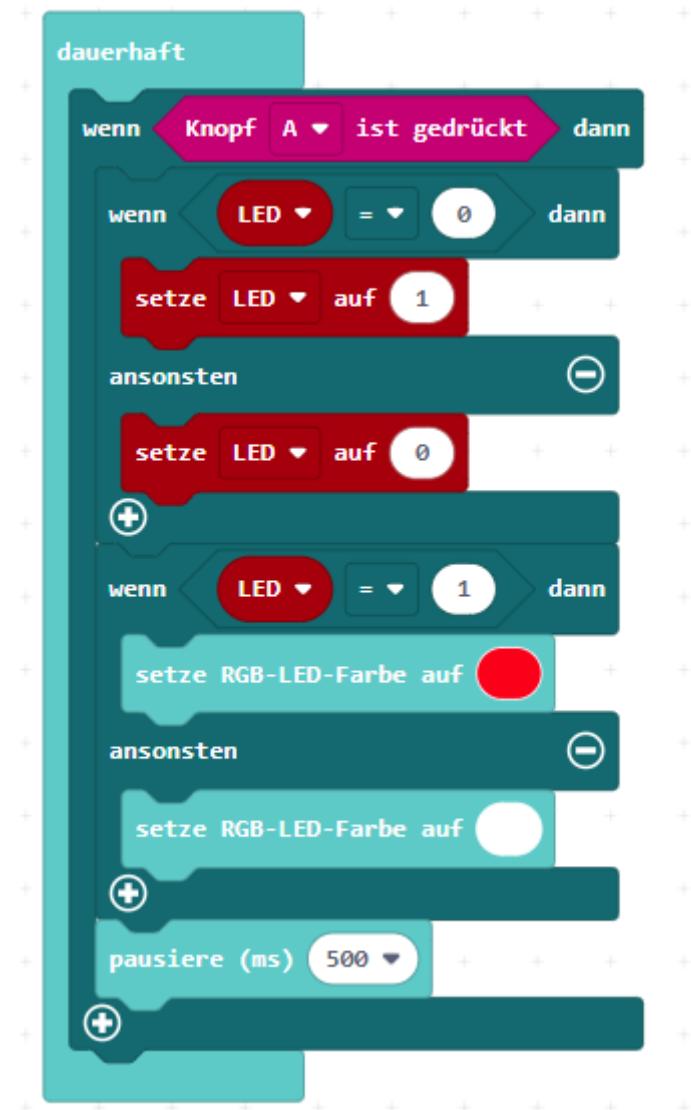
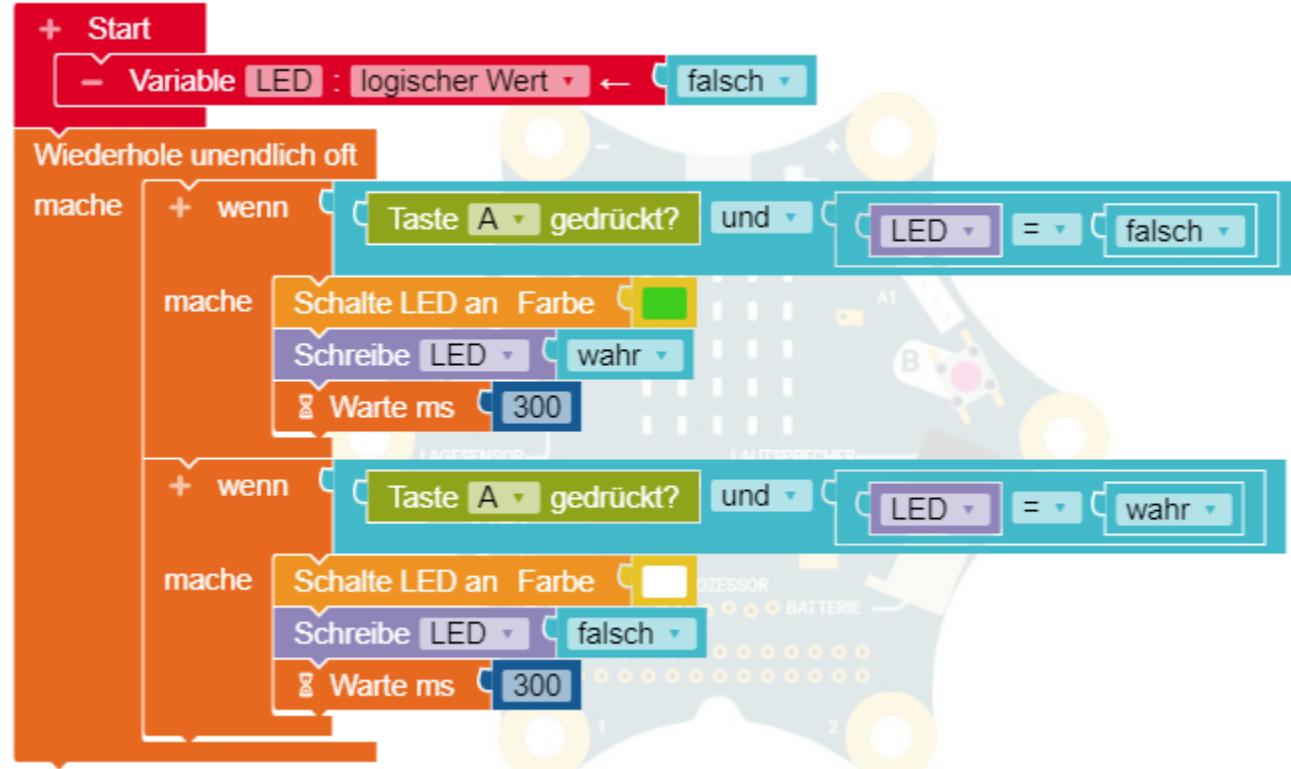
Mit der Taste A soll die RGB-LED ein- oder ausgeschaltet werden und der Zustand nach dem Loslassen der Taste erhalten bleiben (Wechsel-Schalter).



Calliope Workshop



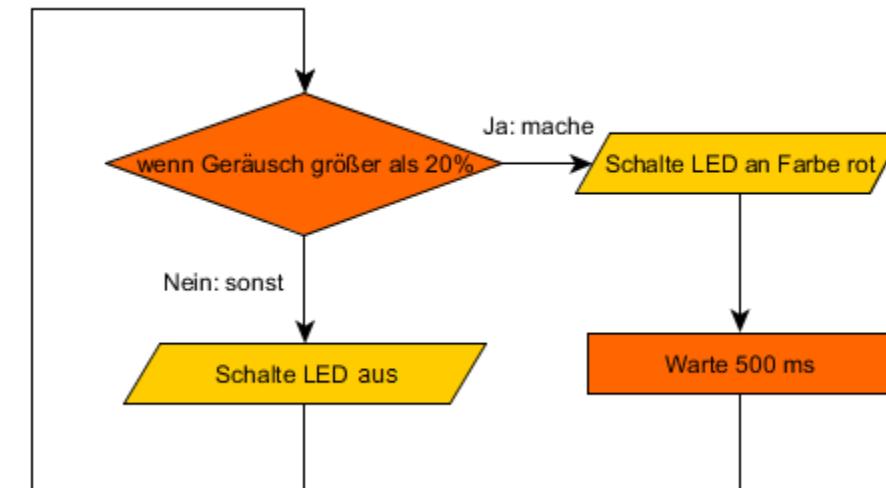
21 Lösung Wechselschalter



Calliope Workshop

22 Aufgabe Klatschschalter

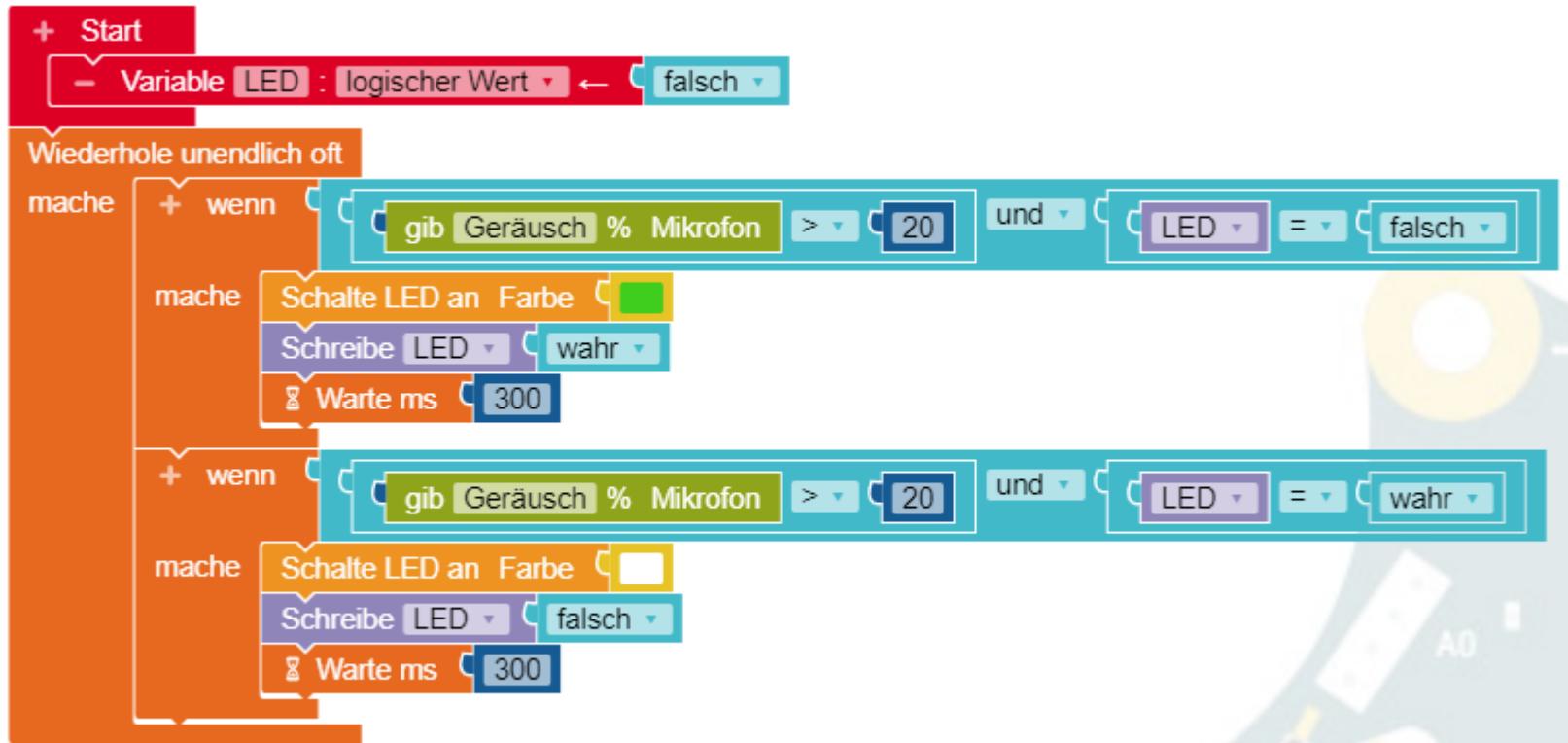
Auf der Basis der vorherigen Aufgabe soll ein Klatschschalter realisiert werden. Mit einem Geräusch (Lautstärke > 20%) wird die LED ein/ausgeschalten.



Calliope Workshop

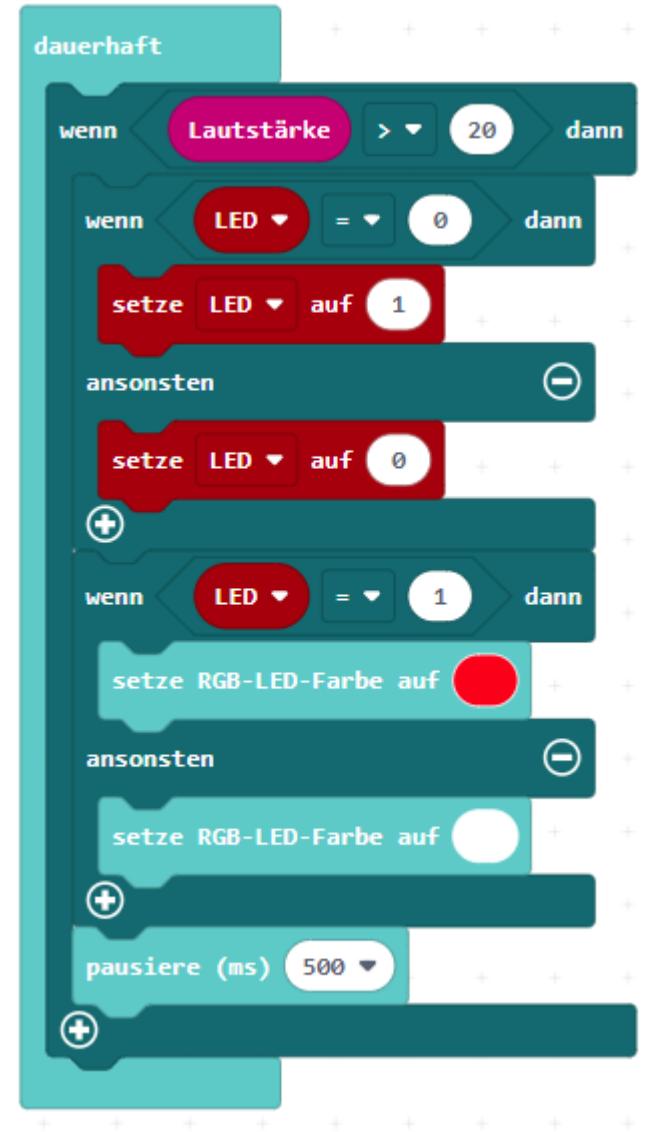


22 Lösung Klatschschalter



A Scratch script titled "Start" with a variable "LED" set to false. It loops infinitely ("Wiederhole unendlich oft"). Inside the loop, it checks if sound from the microphone is greater than 20. If true, it turns the LED on (set LED color to green) and writes "wahr" to the LED. It then waits 300ms. If false, it turns the LED off (set LED color to white) and writes "falsch" to the LED. It also waits 300ms.

```
+ Start
- Variable [LED : logischer Wert] ← [falsch]
Wiederhole unendlich oft
mache
+ wenn [gib Geräusch % Mikrofon > [20]
und [LED = [falsch]]]
mache
Schalte LED an Farbe [grün]
Schreibe [LED : wahr]
Warte ms [300]
+ wenn [gib Geräusch % Mikrofon > [20]
und [LED = [wahr]]
mache
Schalte LED an Farbe [weiß]
Schreibe [LED : falsch]
Warte ms [300]
```



A Scratch script titled "dauerhaft". It uses a "when green flag clicked" hat block. It checks if sound from the microphone is greater than 20. If true, it sets the LED to 1. If false, it sets the LED to 0. It then loops ("ansonsten") with a "set RGB-LED-Farbe auf" block (red), a "setze RGB-LED-Farbe auf" block (white), and a "pausiere (ms) [500]" block.

```
dauerhaft
when green flag clicked
wenn [Lautstärke > [20]] dann
wenn [LED = [0]] dann
setze [LED : auf [1]]
ansonsten
setze [LED : auf [0]]
+
wenn [LED = [1]] dann
setze RGB-LED-Farbe auf [rot]
setze RGB-LED-Farbe auf [weiß]
pausiere (ms) [500]
```

Calliope Workshop



23 Aufgabe Spardose

Mit einem Open Roberta Programm soll eine Spardose mit Anzeige programmiert werden, die alle eingeworfenen Münzen zählt.

P0 zählt 20 Cent-Münzen

P1 zählt 50 Cent-Münzen

P2 zählt 1 Euro-Münzen

P3 zählt 2 Euro-Münzen

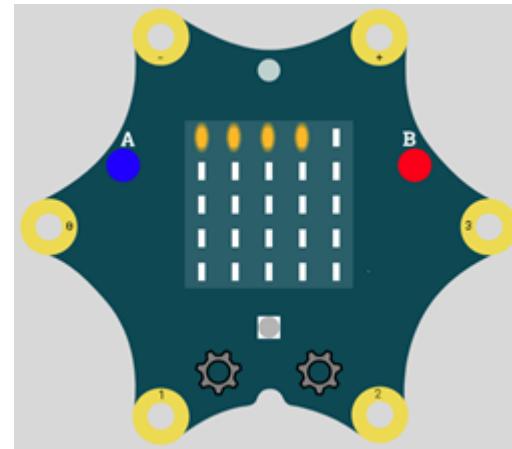
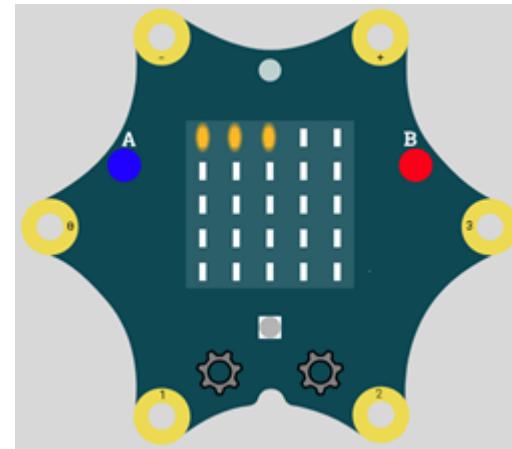
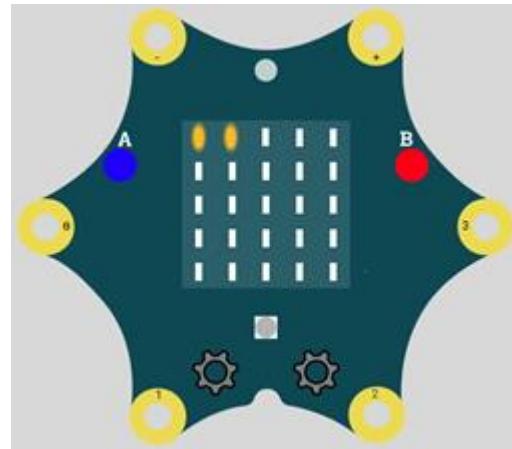
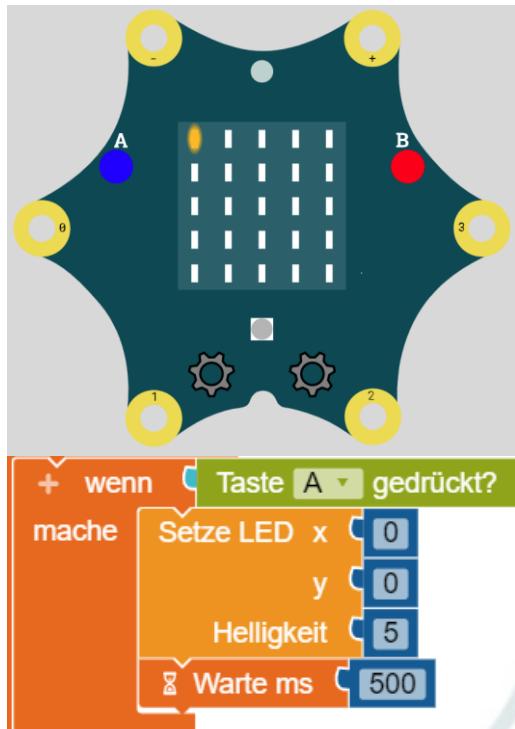
Die Taste A gibt die Summe der eingeworfenen Münzen aus. Die Taste B löscht die Summe.

Die Alarmfunktion der Spardose reagiert auf Bewegung, Licht und Geräusche.

Calliope Workshop

24 Aufgabe Zähler mit Taste A und LED-Matrix

Mit einem Open Roberta Programm wollen wir zählen, wie oft die Taste A gedrückt wurde. Der Zählerstand soll als zusätzlicher **LED-Punkt** im LED-Display des Calliope angezeigt werden.



Calliope Workshop

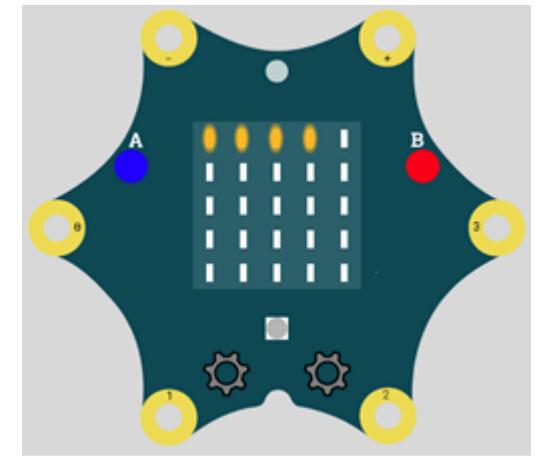
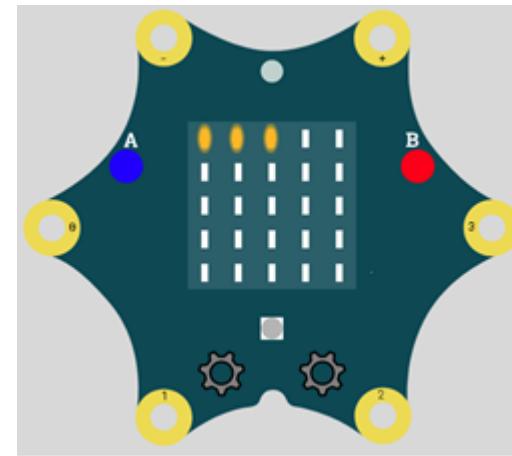
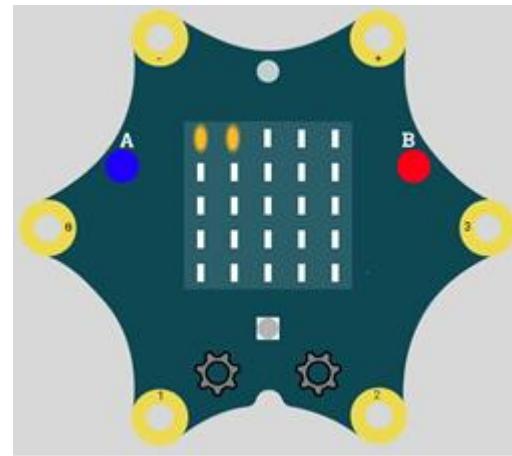
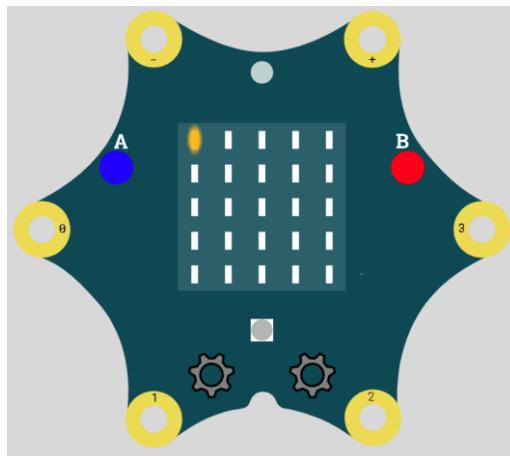
24 Lösung Zähler mit Taste A und LED-Matrix



Calliope Workshop

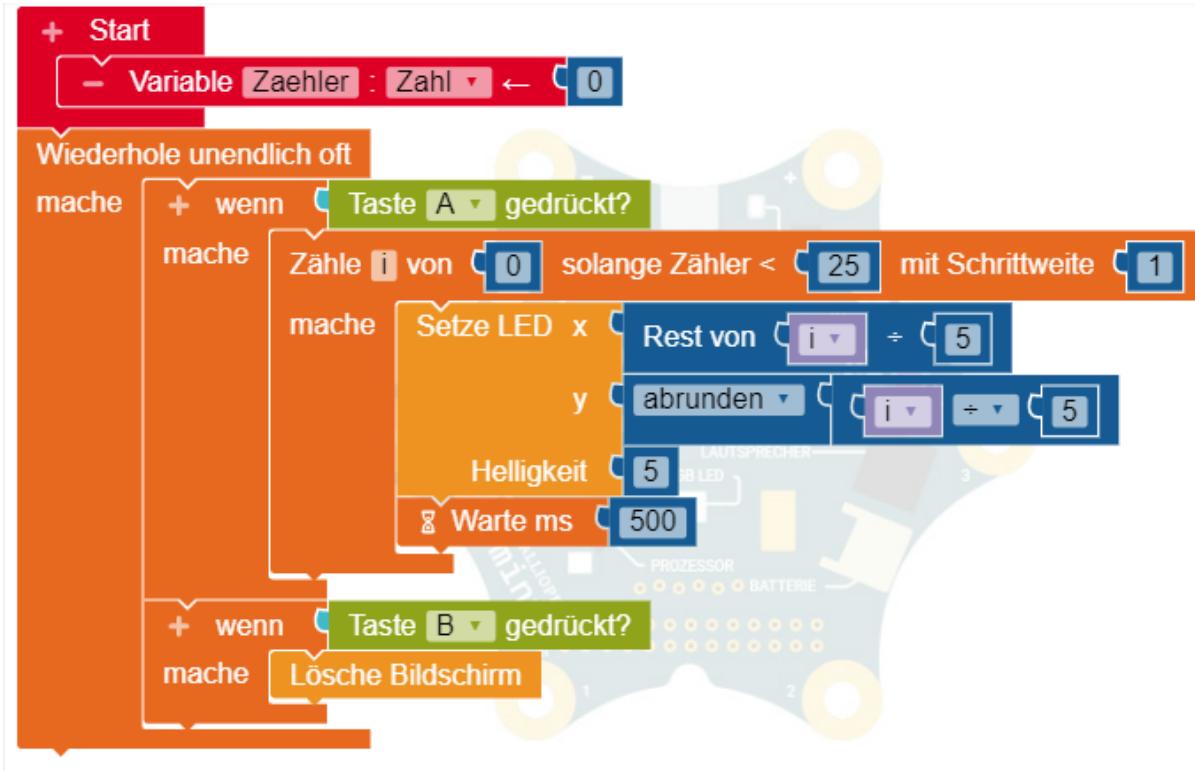
25 Aufgabe Zähler mit Taste A, B und LED-Matrix

Mit einem Open Roberta Programm soll ein **Zähler** gestartet werden, der den Zählerstand mit Taste A startet und als zusätzlichen **LED-Punkt** im LED-Display des Calliope anzeigen. Ist der Zählvorgang abgeschlossen soll die Taste B die LED-Punkte löschen.



Calliope Workshop

25 Lösung Zähler mit Taste A, B und LED-Matrix



C++

```
#include "MicroBit.h"
#include "NEPODefs.h"
#include <list>
#include <array>
#include <stdlib.h>
MicroBit _uBit;
double __Zaehler;

int main()
{
    _uBit.init();
    __Zaehler = 0;
    _uBit.display.setDisplayMode(DISPLAY_MODE_GREyscale);
    while ( true ) {
        if ( _uBit.buttonA.isPressed() ) {
            for ( int __i = 0; __i < 25; __i += 1) {
                _uBit.display.image.setPixelValue((int) __i % ((int) 5), floor(__i / ((float) 5)), (5 * _SET_BRIGHTNESS_MULTIPLIER));
                _uBit.sleep(500);
                _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
            }
        }
        if ( _uBit.buttonB.isPressed() ) {
            _uBit.display.clear();
        }
        _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
    }
    release_fiber();
}
```

Calliope Workshop



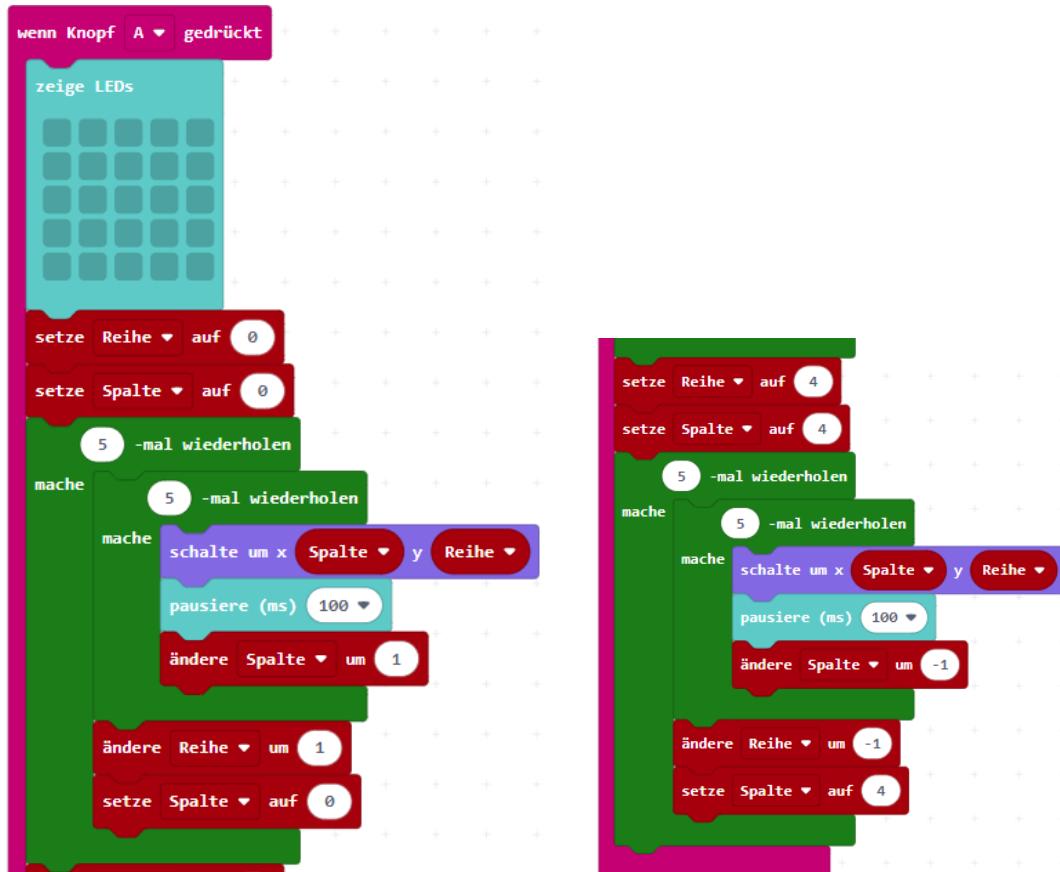
26 Aufgabe LED-Matrix Lauflicht

Mit einem Microsoft MakeCode Programm realisieren wir ein Lauflicht. Wenn die Taste A gedrückt wurden, soll sich die LED-Matrix im $\frac{1}{2}$ Sekundentakt von links oben nach rechts unten füllen und danach soll sich die LEDs der LED-Matrix von rechts unten nach links oben wieder ausschalten.



Calliope Workshop

26 Lösung LED-Matrix-Lauflicht



Python

Reihe = 0
Spalte = 0

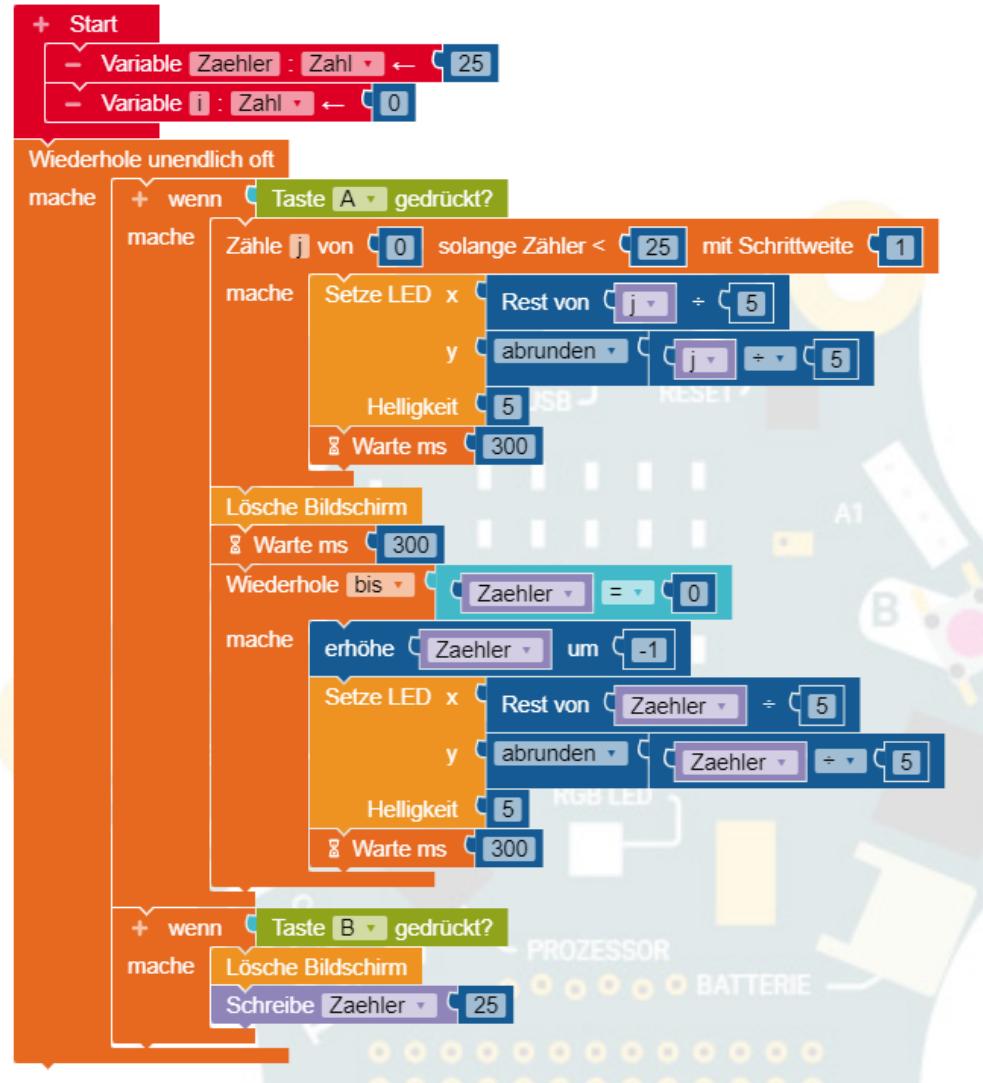
```
def on_button_pressed_a():
    global Reihe, Spalte
    basic.show_leds("""
        .....
        .....
        .....
        .....
        .....
    """)
```

```
Reihe = 0
Spalte = 0
for index in range(5):
    for index2 in range(5):
        led.toggle(Spalte, Reihe)
        basic.pause(100)
        Spalte += 1
    Reihe += 1
    Spalte = 0
Reihe = 4
Spalte = 4
for index3 in range(5):
    for index4 in range(5):
        led.toggle(Spalte, Reihe)
        basic.pause(100)
        Spalte += -1
    Reihe += -1
    Spalte = 4
```

```
input.on_button_pressed(Button.A, on_button_pressed_a)
```

Calliope Workshop

26 Lösung LED-Matrix-Lauflicht



Calliope Workshop

27 Aufgabe Sende einen Smiley

Mit einem Open Roberta Programm wollen wir zwei Calliope koppeln. Im ersten Schritt soll ein Smiley übertragen werden.

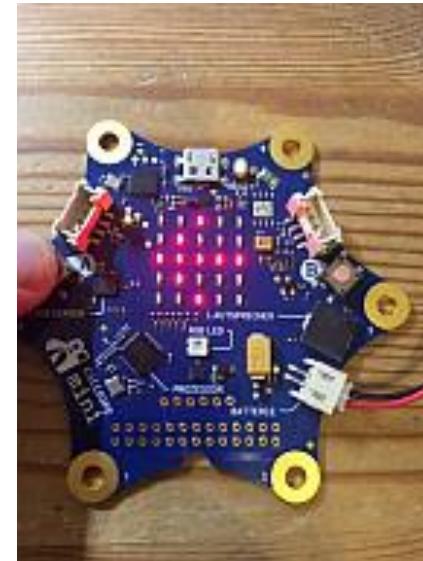
Im zweiten Schritt werden wir einen Calliope fernsteuern. Die Funktionen:

Taste A: Fahre nach links

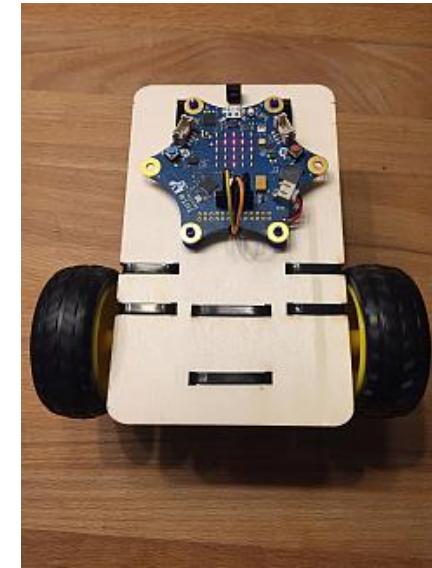
Taste B: Fahre nach rechts

Taste A+B: Fahre gerade aus

Keine Taste: STOPP



Sender



Empfänger

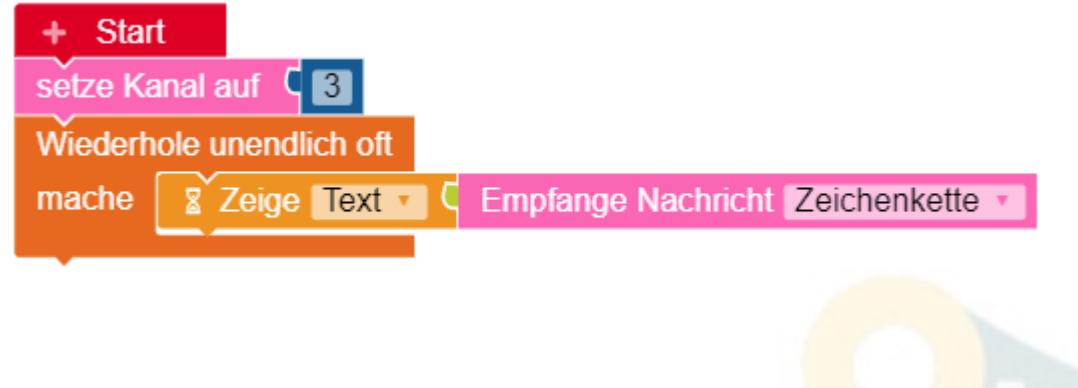


Calliope Workshop

27 Lösung Smiley senden



27 Lösung Smiley empfangen



C++

```
//Sender
#include "MicroBit.h"
#include "NEPODefs.h"
#include <list>
#include <array>
#include <stdlib.h>
MicroBit _uBit;

int main()
{
    _uBit.init();

    _uBit.radio.enable();
    _uBit.radio.setGroup(3);
    while ( true ) {
        _uBit.radio.setTransmitPower(7);
        _uBit.radio.datagram.send(ManagedString((ManagedString(":-"))));
        _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
    }
    release_fiber();
}
```

Calliope Workshop



28 Aufgabe

Programmiere jetzt den Sender für die Fahrrichtungen.

Erstelle zuerst eine Funktionstabelle für die Tasten A und B.

Taste A: Fahre nach links

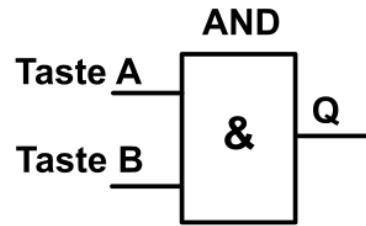
Taste B: Fahre nach rechts

Taste A+B: Fahre vorwärts

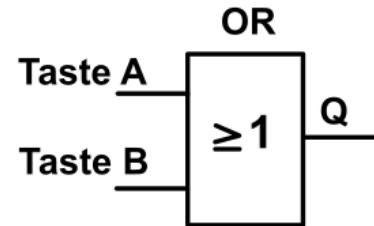
Die Fahrrichtung nach Betätigung der Tasten in der LED-Matrix mit Pfeilen angezeigt werden. Bei Stopp erscheint ein Quadrat.

Calliope Workshop

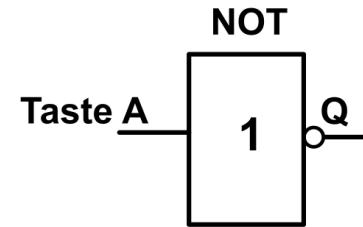
28 Logikbausteine



A	B	Q
0	0	0
1	0	0
0	1	0
1	1	1



A	B	Q
0	0	0
1	0	1
0	1	1
1	1	1

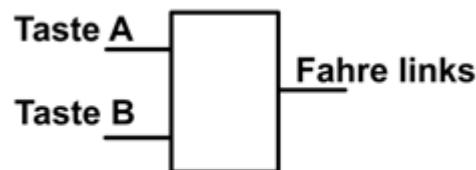


A	B	Q
0	0	1
1	0	0

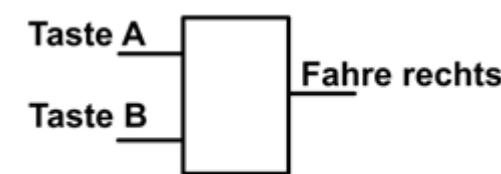
Calliope Workshop

28 Aufgabe

Ergänze den FUP und die Funktionstabelle für die Taster-Logik



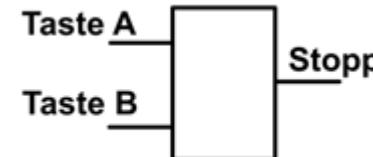
A	B	Q



A	B	Q



A	B	Q



A	B	Q

Calliope Workshop



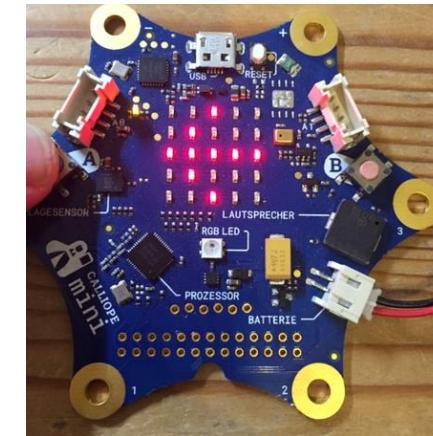
29 Aufgabe Programmiere den Sender für den Roboter nach deiner Funktionstabelle mit den Tasten A und B sowie der LED-Matrix-Anzeige für

Taste A: Fahre nach links

Taste B: Fahre nach rechts

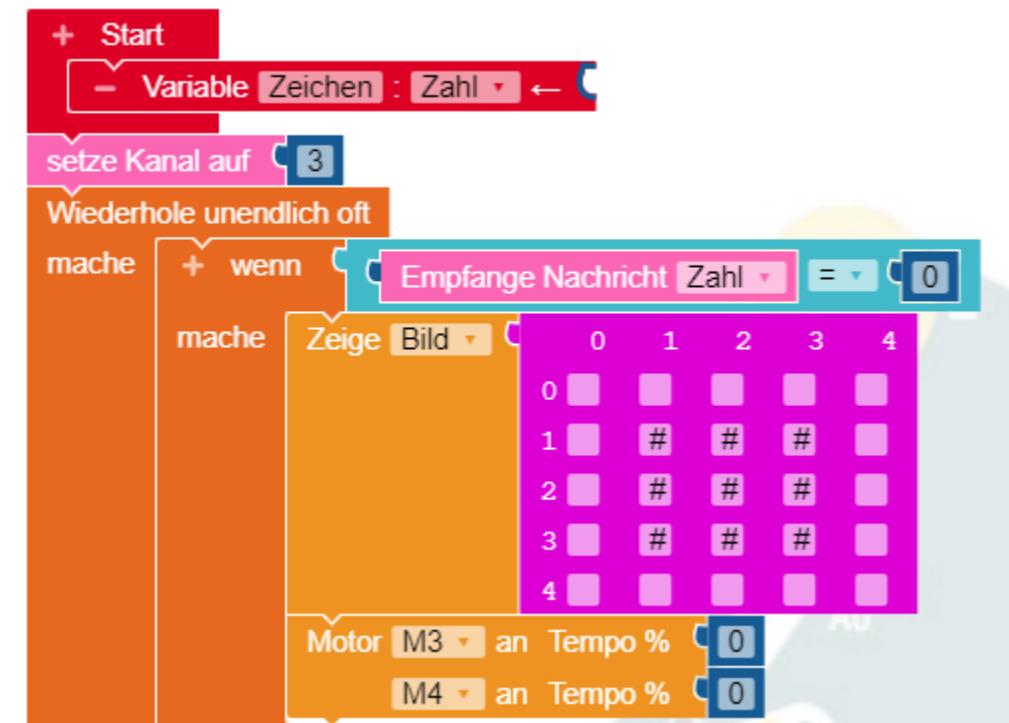
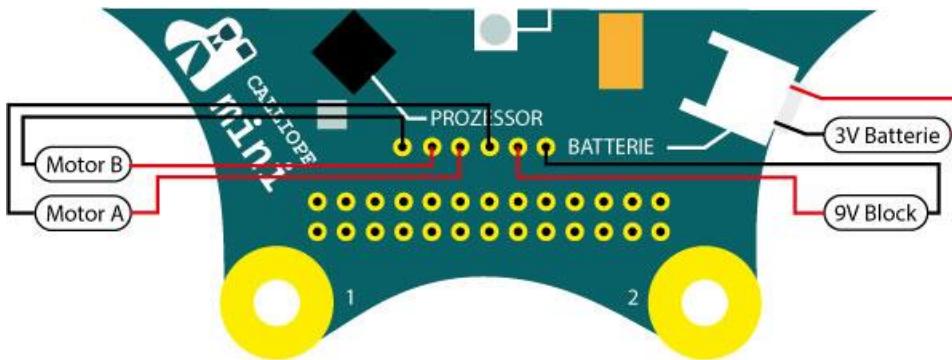
Taste A+B: Fahre vorwärts

Die Fahrrichtung nach Betätigung der Tasten in der LED-Matrix mit Pfeilen angezeigt werden. Bei Stopp erscheint ein Quadrat.



Calliope Workshop

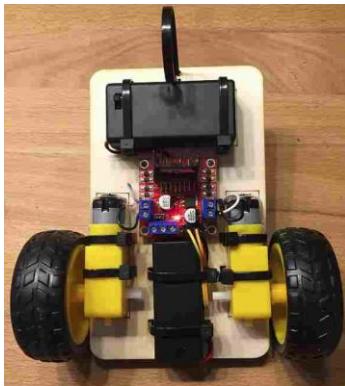
30 Aufgabe Programmiere den Empfänger für den Roboter. Die Motoren A und B dienen als Antrieb.



Calliope Workshop



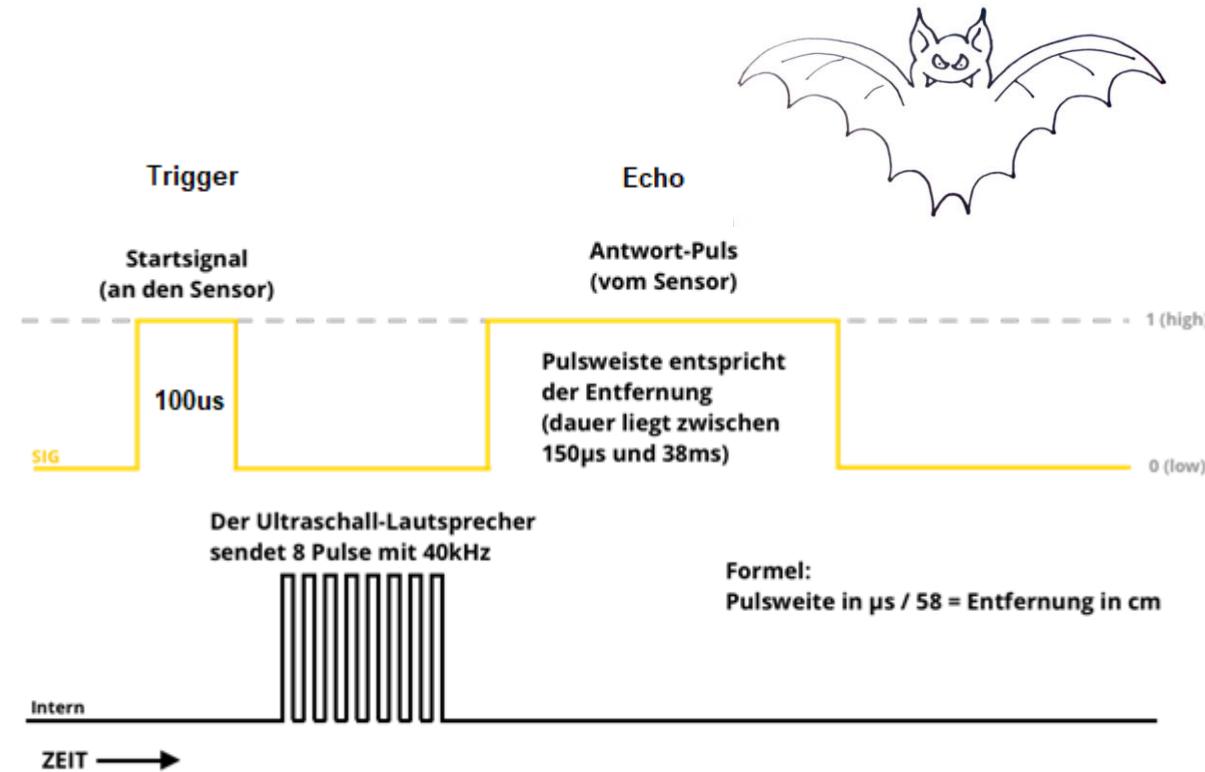
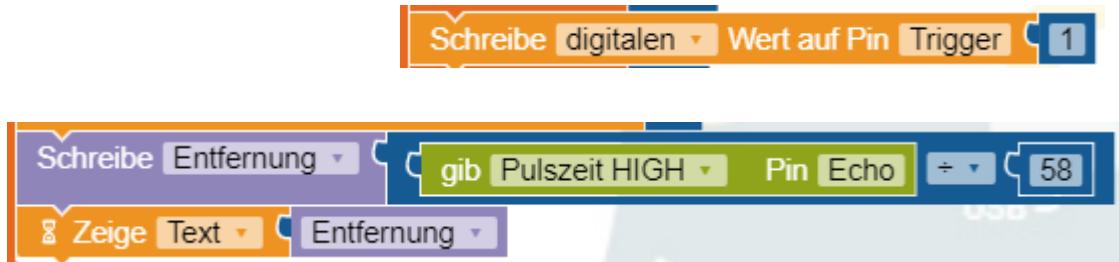
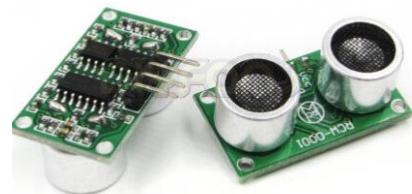
Wenn zwei Roboter fertig sind, könnt ihr einen Parkour abstecken und ein Rennen fahren!



Calliope Workshop



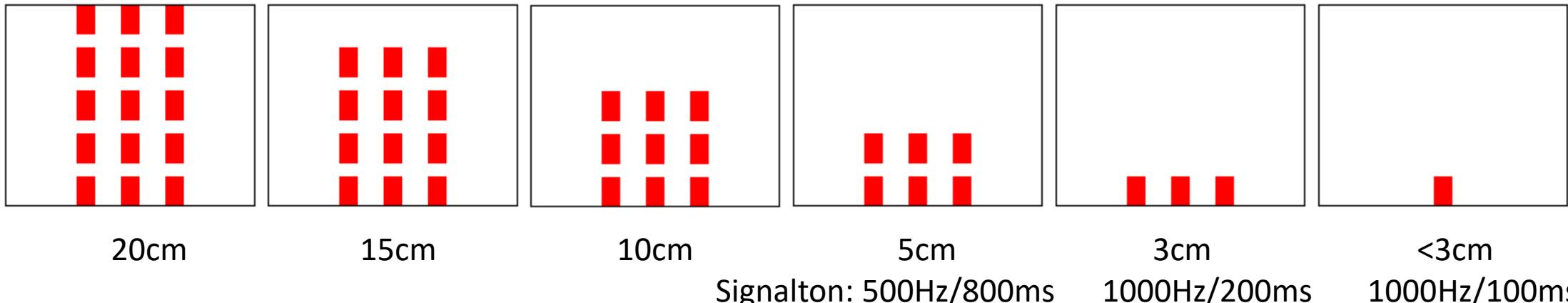
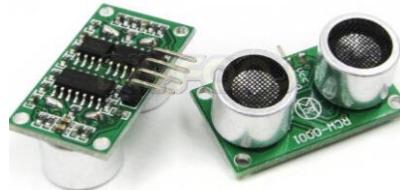
33 Aufgabe Programmiere eine Hinderniserkennung für deinen Roboter mit einem Ultraschallsensor.



Calliope Workshop



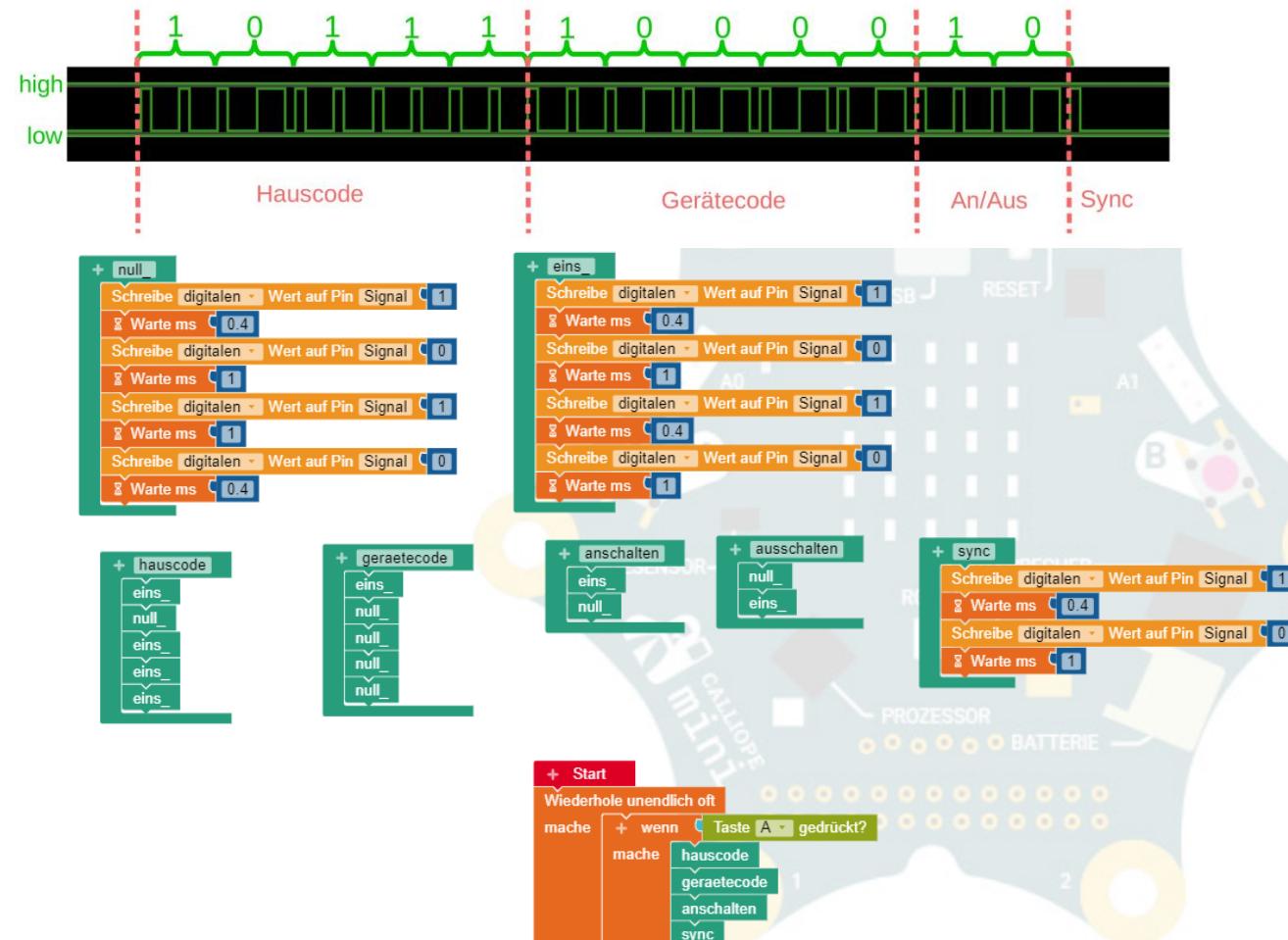
34 Aufgabe Programmiere eine Hinderniserkennung für deinen Roboter mit einer Balkenanzeige und einem Signalton.



Calliope Workshop



35 Aufgabe Programme mit Funktionsblöcken (Unterprogramme)



Calliope Workshop

36 Aufgabe Wir programmieren uns eine Wetterstation

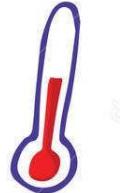
Wetterdaten:

Sonnenschein

Regen

Wind

Temperatur



Bestimme die Sensoren für die jeweiligen Wetterdaten und die dazugehörigen Eingänge.

Lege fest, wie die Wetterdaten dargestellt werden sollen.

Erstelle das Programm und teste es.

Calliope Workshop



Wie geht's weiter?

Projekte und Anregungen:

<https://makecode.calliope.cc/projects>

[Open Roberta Lab \(open-roberta.org\)](https://open-roberta.org) - Galerie

The screenshot shows the "Dokumentation" section of the Calliope website. At the top, there is a navigation bar with icons for search, user profile, and navigation. Below it, a large "Projekte" heading is followed by three cards: "Erste Schritte" (showing a hand plugging a cable into a laptop), "Tutorials" (showing a grid of colored squares), and "Calliope Links" (showing a stack of Calliope modules). A "Siehe auch" section at the bottom lists links to "Erste Schritte", "Tutorials", "Calliope Links", "Bearbeiten dieser Seite auf GitHub", and "Vorlage dieser Seite auf GitHub bearbeiten".

The screenshot shows the "PROGRAMM Schalte_LED_e_a ROBOTERKONFIGURATION" section of the Open Roberta Lab project gallery. It displays a grid of six projects for the "Calliope mini": "Farben" (purple card, created by fridolin_friedrich on 26.11.2020, 12:05, 20 likes), "GeiloSpiel" (pink card, created by elias.eppert on 18.11.2020, 09:51, 267 likes), "SpacelInvaders" (teal card, created by hewinx on 13.11.2020, 21:30, 214 likes), "THAlarm2" (green card, created by Roboterfreund), "THAlarm" (light blue card, created by Roboterfreund), and "Alarmanlage_1_2" (dark blue card, created by Roboterfreund). Each project card includes a "gefällt mir" (likes) button.

Calliope Workshop



Wie geht's weiter?

Programmierung in C++

Calliope Workshop



Wie geht's weiter?

Programmierung in Java und Python

Online Tipps und Hilfen: <https://makecode.calliope.cc/about>

The screenshot shows the Microsoft MakeCode editor interface for the Calliope mini. On the left is a graphical workspace with a Calliope mini board icon. To its right is a sidebar with categories like Grundlagen, Eingabe, Musik, LED, Funk, Schleifen, Logik, Variablen, Mathematik, and Motoren. Below these are sections for Explorer, Fortgeschritten, and a list of recent projects. The main area contains a text-based code editor with the following JavaScript code:

```
1 let Reihe = 0
2 let Spalte = 0
3 input.onButtonPressed(Button.A, function () {
4     basic.showLeds(`
5     .....
6     .....
7     .....
8     .....
9     .....
10    `)
11    Reihe = 0
12    Spalte = 0
13    for (let index = 0; index < 5; index++) {
14        for (let index = 0; index < 5; index++) {
15            led.toggle(Spalte, Reihe)
16            basic.pause(100)
17            Spalte += 1
18        }
19        Reihe += 1
20        Spalte = 0
21    }
22    Reihe = 4
23    Spalte = 4
24    for (let index = 0; index < 5; index++) {
25        for (let index = 0; index < 5; index++) {
26            led.toggle(Spalte, Reihe)
27            basic.pause(100)
28            Spalte += -1
29        }
30        Reihe += -1
31        Spalte = 4
32    })
33 })
34 }
```

The screenshot shows the Microsoft MakeCode documentation page for Python. At the top is a navigation bar with "Documentation" and links for "Über", "Häufige Fragen / FAQ", "Projekte", "GitHub", "Blöcke", and "Python". The "Python" section is expanded, showing sub-topics: Aufrufen, Sequenzierung, Variablen, Operatoren, Anweisungen, Funktionen, Klassen, and Entwickler. Below this is a list of topics under "Python": Calling, Sequencing, Variables, Operators, Statements, Functions, and Classes. At the bottom are links to edit the page on GitHub.

Docs > Python

Python

The Microsoft MakeCode programming environment uses Python als

These topics give a brief introduction to Python with MakeCode:

- **Calling** - How to use a function
- **Sequencing** - Ordering statements in code
- **Variables** - Remember data and save values
- **Operators** - Operations to change and compare values
- **Statements** - The elements of code that take action
- **Functions** - Portions of code to use again and again
- **Classes** - Contain related data and operations together

[Edit this page on GitHub](#)
[Edit template of this page on GitHub](#)

Calliope Workshop



Wie geht's weiter?

Programmierung in Python

Python Editor: <https://codewith.mu/en/download>

A screenshot of the Mu Python editor interface. The title bar says "Mu 1.0.3 - Taster-LED.py". The toolbar contains icons for Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Hilfe, and Beenden. The main window shows three tabs: "Hello Mu.py" (selected), "Unbenannt *", and "Taster-LED.py". The code in "Taster-LED.py" is:

```
1 """
2 Taster->LED
3
4 =====
5 import digitalio
6 import board
7
8 led = digitalio.DigitalInOut(board.D13)
9 led.switch_to_output()
10 button = digitalio.DigitalInOut(board.D2)
11 button.switch_to_input()
12
13 while True:
14     btn_state = button.value
15     led.value = not btn_state
```

The status bar at the bottom right shows "Microbit" and a gear icon.

Calliope Workshop



Wie geht's weiter?

Programmierung in Python

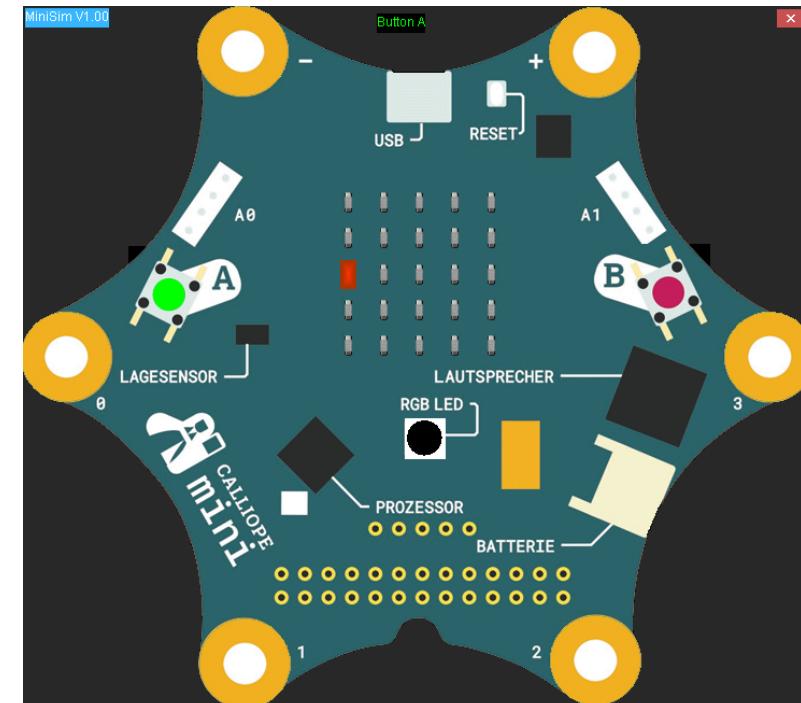
Python Editor: <https://www.tigerjython4kids.ch/>

TigerJython

Datei Bearbeiten Ausführen Tools Hilfe

TB_blink.py x RGB_LED.py x untitled 1* x Cp5b.py x

```
1 from calliope_mini import *
2
3 def blink(x, y):
4     display.set_pixel(x, y, 9)
5     sleep(500)
6     display.set_pixel(x, y, 0)
7     sleep(500)
8
9 while True:
10    if button_a.is_pressed():
11        blink(0, 2)
12    if button_b.is_pressed():
13        blink(4, 2)
14
```



Calliope Workshop



Wie geht's weiter?

C++ Programme schreiben und simulieren mit [Tinkercad.com](https://www.tinkercad.com)

A screenshot of the Tinkercad website. On the left, there's a sidebar with a robot icon, the name "fräsky", a search bar, and buttons for "3D-Entwürfe" (selected), "Circuits", "Codeblöcke", and "Lektionen". Below that is a section for "Deine Klassen" with a "Projekte" tab showing "Project 1" and a "+ Projekt erstellen" button. Further down are "Tweets" and a "Tinkercad Retweeted" section with a post from Amanda Alderman (@Alderman_WHS) about a lunch crew using Tinkercad to 3D print. The main content area features a "Tinkercad Lesson Plans" pop-up window with a photo of two children at a computer. Below it, a heading "Meine zuletzt verwendeten Entwürfe" leads to a grid of 10 3D models: "Cool Bombul-Elzing", "R2D2", "3D-Workshop Ferienprogramm", "Heliostat", "Bad", "Ampel", "Kindl", "Eber", and "Mammut". Each model has a small preview image, its name, a timestamp, and a "Privat" status indicator. At the top right of the main area are "Katalog" and "Blog" links.

Calliope Workshop



Wie geht's weiter?

C++ Programme schreiben und simulieren mit [Tinkercad.com](https://www.tinkercad.com)

A screenshot of the Tinkercad interface. On the left, a breadboard circuit is shown with an Arduino Uno at its center. A 4.30 V power source is connected to the Arduino's 5V pin and ground. Two potentiometers are connected to analog pins A0 and A1. The Arduino's digital pins 7, 8, 9, 10, 11, and 12 are connected to the breadboard via green wires. A 16x2 LCD screen displays "Analogwert: 71.00". On the right, the Arduino Uno R3 board is selected in the simulation environment. The code editor shows a C++ program for reading analog values and displaying them on an LCD. A red circle highlights the "Blöcke und Text" (Blocks and Text) button in the code editor's toolbar.

```
int tempPin = 5; // AREF RS E D4 D5 D6 D7
LiquidCrystal lcd(12, 11, 8, 7, 6, 5);
void setup()
{
    lcd.begin(16, 2);
}
void loop()
{
    int tempReading = analogRead(tempPin);
    double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
    tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * t
    float tempC = tempK - 274.15; // Kelvin zu Celcius
    //float tempF = (tempC * 9.0) / 5.0 + 32.0; // Celcius zu Fahr
    // Display Temperatur in C
    lcd.setCursor(2, 0);
    lcd.print("Analogwert: ");
    // Display Temperatur in F
    //lcd.print("Temp F ");
    lcd.setCursor(5, 1);
    // Display Temperatur in C
    lcd.print(tempC);
    // Display Temperatur in F
    //lcd.print(tempF);
    delay(500);
}
```

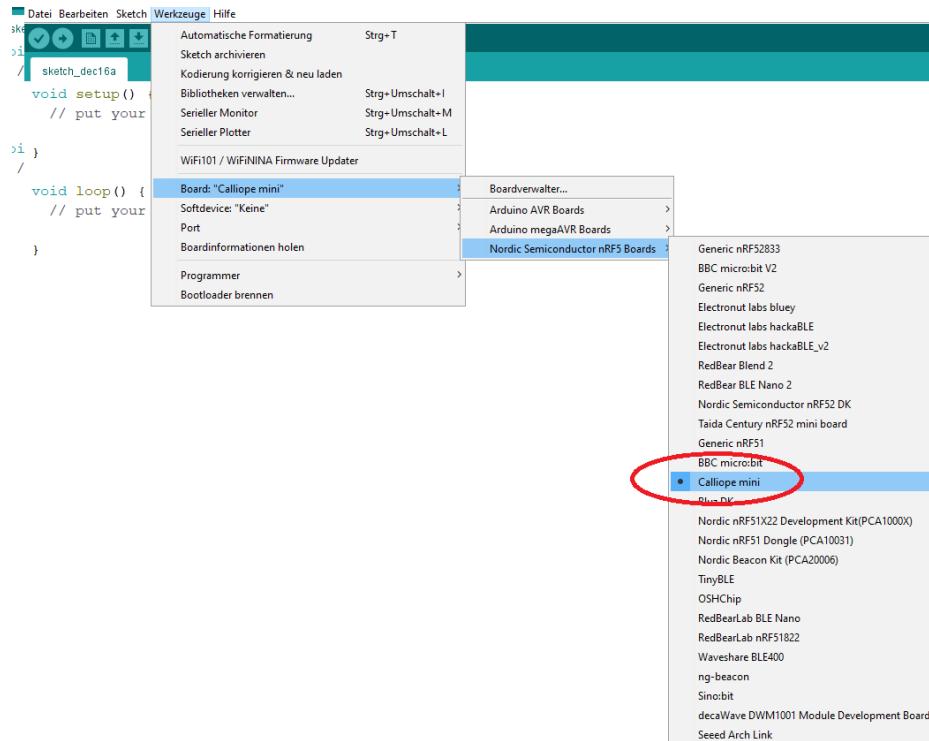
Calliope Workshop



Wie geht's weiter?

C++ Programme schreiben und simulieren mit Arduino IDE

<https://learn.adafruit.com/use-micro-bit-with-arduino/install-board-and-blink>



Calliope Workshop



Wir wünschen dir viel Spaß beim Programmieren!

