

Calli:bot beim meet and code 2019 v1.0

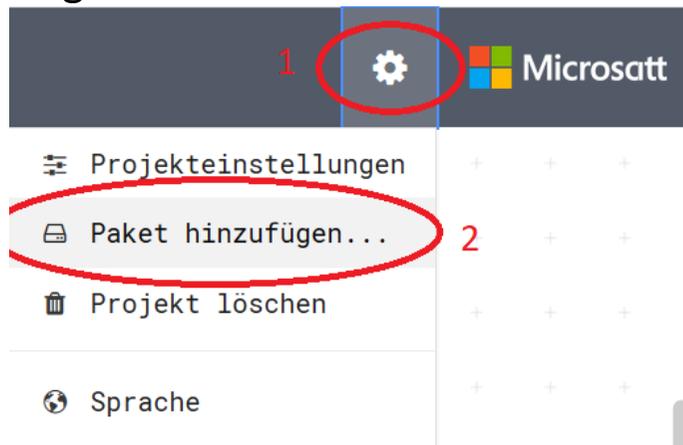


Um Deinen *Calli:bot* programmieren zu können gibst Du in der Adresszeile deines Browsers die Adresse der Programmierumgebung MakeCode ein:

makecode.calliope.cc

Nun musst Du zuerst die passenden Programmierblöcke laden. Das geht so:

a) Auf das Zahnradsymbol gehen und Paket hinzufügen... wählen



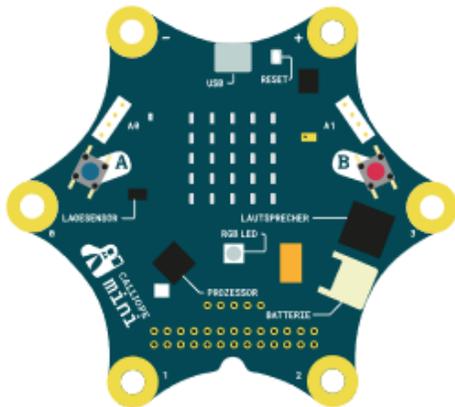
b) Paketnamen eingeben:

<https://github.com/knotechgmbh/Callibot>



Suche die passenden Programmierblöcke aus dem Menü und baue Dein erstes Programm zusammen. Die Farben der Blöcke geben an, in welcher Kategorie Du suchen musst.

Was macht Dein *Calli:bot*?



Suche...

- Grundlagen
- Eingabe
- Musik
- LED
- Schleifen
- Logik
- Variablen
- Mathematik
- Calli:bot

beim Start

pausiere (ms) 500

Schalte Motor links vorwärts mit 0 %

Stoppe Motor beide auslaufend

beim Start

Schalte Motor beide vorwärts mit 100 %

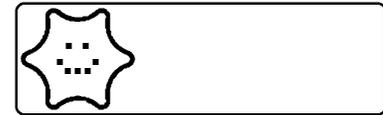
pausiere (ms) 500

Schalte Motor links rückwärts mit 50 %

Schalte Motor rechts vorwärts mit 50 %

pausiere (ms) 500

Stoppe Motor beide bremsend



Calli:bot Herausforderung RGB-Beleuchtung

Der *Calli:bot* besitzt 4 RGB-LEDs (links vorne, rechts vorne, Links hinten und rechts hinten) die einzeln angesteuert werden können. Lasse sie nacheinander in verschiedenen Farben blinken! Schalte sie dazu der Reihe nach an. Wenn Du eine weitere LED angeschaltet hast, musst Du die vorherige LED ausschalten. Verwende dazu diese 4 Blöcke

Volle Helligkeit 8 = LED anschalten

```
Schalte Beleuchtung links vorne Farbe rot Helligkeit 8 (0..8)
Schalte Beleuchtung links vorne Farbe rot Helligkeit 0 (0..8)
```

```
pausiere (ms) 500
```

```
dauerhaft
```

Werde Kreativ!

Helligkeit 0 = LED ausschalten

Du kannst auch alle LEDs auf einmal einschalten oder mit einer Variablen in einer Schleife die LEDs dimmen (pausieren nicht vergessen).

```
für Index von 0 bis 8
  mache
    Schalte Beleuchtung rechts vorne Farbe violett Helligkeit 8 - Index (0..8)
    Schalte Beleuchtung links hinten Farbe grün Helligkeit Index (0..8)
```

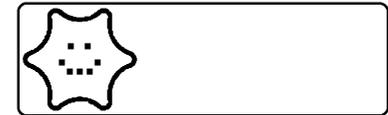
abdunkeln

Heller werden

Calli:bot Herausforderung RGB-Beleuchtung (Mögliche Lösung)

```
dauerhaft
  Schalte Beleuchtung links vorne Farbe rot Helligkeit 8 (0..8)
  pausiere (ms) 100
  Schalte Beleuchtung rechts vorne Farbe blau Helligkeit 8 (0..8)
  Schalte Beleuchtung links vorne Farbe grün Helligkeit 0 (0..8)
  pausiere (ms) 100
  Schalte Beleuchtung links hinten Farbe violett Helligkeit 8 (0..8)
  Schalte Beleuchtung rechts vorne Farbe grün Helligkeit 0 (0..8)
  pausiere (ms) 100
  Schalte Beleuchtung rechts hinten Farbe violett Helligkeit 8 (0..8)
  Schalte Beleuchtung links hinten Farbe grün Helligkeit 0 (0..8)
  pausiere (ms) 100
  Schalte Beleuchtung links hinten Farbe grün Helligkeit 0 (0..8)
```

Calli:bot Herausforderung Lärmsensor (Mikrofon)



Sicher hast Du gemerkt, dass manche Deiner MitschülerInnen schreien, wenn der *Calli:bot* kurz davor ist vom Tisch zu fallen. Mit dem Mikrofon des Calliope Mini kannst Du den Lärmpegel messen. Ein Wert am Mikrofonpin $>$ als 700 sollte den Mini stoppen zu lassen. Wir benötigen zusätzliche Logikbausteine und den Baustein zum analogen Lesen des Mikrofonpins aus dem

Menü ▲ Fortgeschritten 📍 Pins

```
Scratch code blocks:  
- Pin: MIC  
- Action: Stoppe Motor (beide, auslaufend)  
- Action: Schalte Motor (links, vorwärts, mit 0%)  
- Control: dauerhaft  
- Logic: wenn (wahr) dann (pausiere 500 ms) ansonsten
```

Werde Kreativ!

- Speichere in einer Variablen ob der *Calli:bot* gerade fährt oder steht. Wenn er steht, soll er durch erneutes Rufen wieder losfahren!
- Oder messe die Lichtstärke! Wenn es hell ist soll der *Calli:bot* fahren, wenn es dunkler wird (Lichtstärke $<$ 60) „schlafen“.

Verwende dazu 📍 Lichtstärke oder ⚙️ Warte bis Helligkeit < 60

Calli:bot Herausforderung Lärmsensor (Lösung)

```

dauerhaft
  wenn (analoge Werte von Pin MIC < 700)
  dann
    Schalte Motor beide vorwärts mit 100 %
  ansonsten
    Stoppe Motor beide auslaufend
    pausiere (ms) 3000

```

Calli:bot Stop and Go

```

beim Start
  ändere Motoran auf wahr

dauerhaft
  wenn (analoge Werte von Pin MIC < 700)
  dann
    wenn (Motoran)
    dann
      Schalte Motor beide vorwärts mit 100 %
  ansonsten
    ändere Motoran auf nicht (Motoran)
    Stoppe Motor beide auslaufend

```



Calli:bot Herausforderung Lichtstärke (Lösungsmöglichkeiten)

```

dauerhaft
  Schalte Motor beide vorwärts mit 80 %
  Warte bis Helligkeit < 60
  Stoppe Motor beide auslaufend

```

```

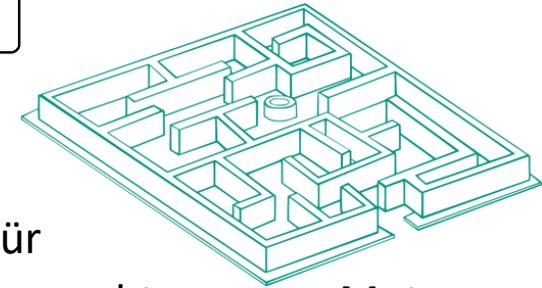
dauerhaft
  wenn Lichtstärke > 60
  dann Schalte Motor beide vorwärts mit 100 %
  ansonsten Stoppe Motor beide auslaufend

```

Calli:bot Herausforderung Labyrinth



PUBLIC DOMAIN



- (1)** Damit man den *Calli:bot* z.B. durch ein Labyrinth steuern kann muss er so programmiert werden, dass er sich genau um 90° nach rechts oder links drehen kann. Dazu musst Du für eine kurze Zeit den linken Motor vorwärts und gleichzeitig den rechten Motor rückwärtslaufen lassen. Für eine umgekehrte Drehung natürlich genau andersherum. Denke daran einen Prozentsatz für die Geschwindigkeit anzugeben!



- (2)** Natürlich musst Du dem *Calli:bot* auch beibringen wie er eine festgelegte Streckenlänge geradeaus fahren kann. Programmiere eine Strecke von 50 cm!



- (3)** Kombiniere die Programmteile **(1)** und **(2)** so, dass der *Calli:bot* ein Quadrat mit 50 cm Seitenlänge fährt. Wenn gleiche Programmteile mehrfach vorkommen kannst Du auch eine Schleife verwenden.



Calli:bot Herausforderung Labyrinth 90° (mögliche Lösung)

beim Start

Schalte Motor links vorwärts mit 100 %

Schalte Motor rechts rückwärts mit 100 %

pausiere (ms) 1370

Stoppe Motor beide bremsend

Die Länge der
Pausen variiert
je nach
Entladung der
Batterien!

Calli:bot Herausforderung Labyrinth 50 cm (mögliche Lösung)

beim Start

Schalte Motor beide vorwärts mit 100 %

pausiere (ms) 3200

Stoppe Motor beide bremsend

Calli:bot Herausforderung Labyrinth Quadrat fahren (mögliche Lösung)

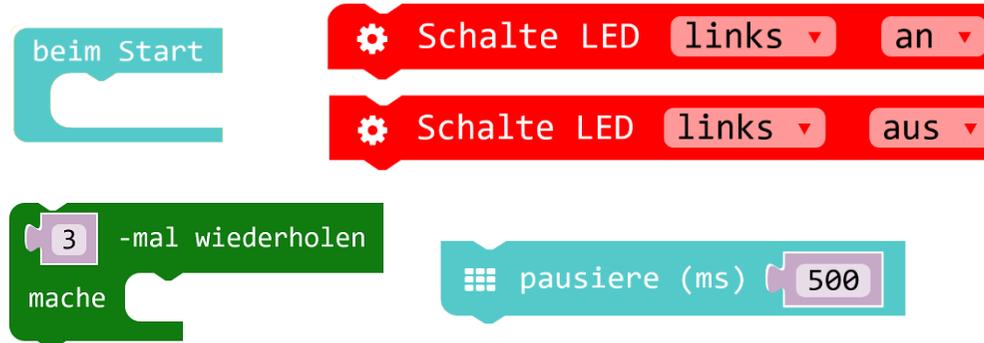
```
beim Start
  4 -mal wiederholen
  mache
    Schalte Motor beide vorwärts mit 100 %
    pausiere (ms) 3200
    Stoppe Motor beide bremsend
    Schalte Motor links vorwärts mit 100 %
    Schalte Motor rechts rückwärts mit 100 %
    pausiere (ms) 1370
```

Calli:bot Herausforderung Blinken



Der *Calli:bot* besitzt zwei rote Blinker.

- 1) Schalte jeden Blinker dreimal eine halbe Sekunde (500 ms) lang an und aus
- 2) Jetzt sollen beide gleichzeitig eine halbe Sekunde lang blinken

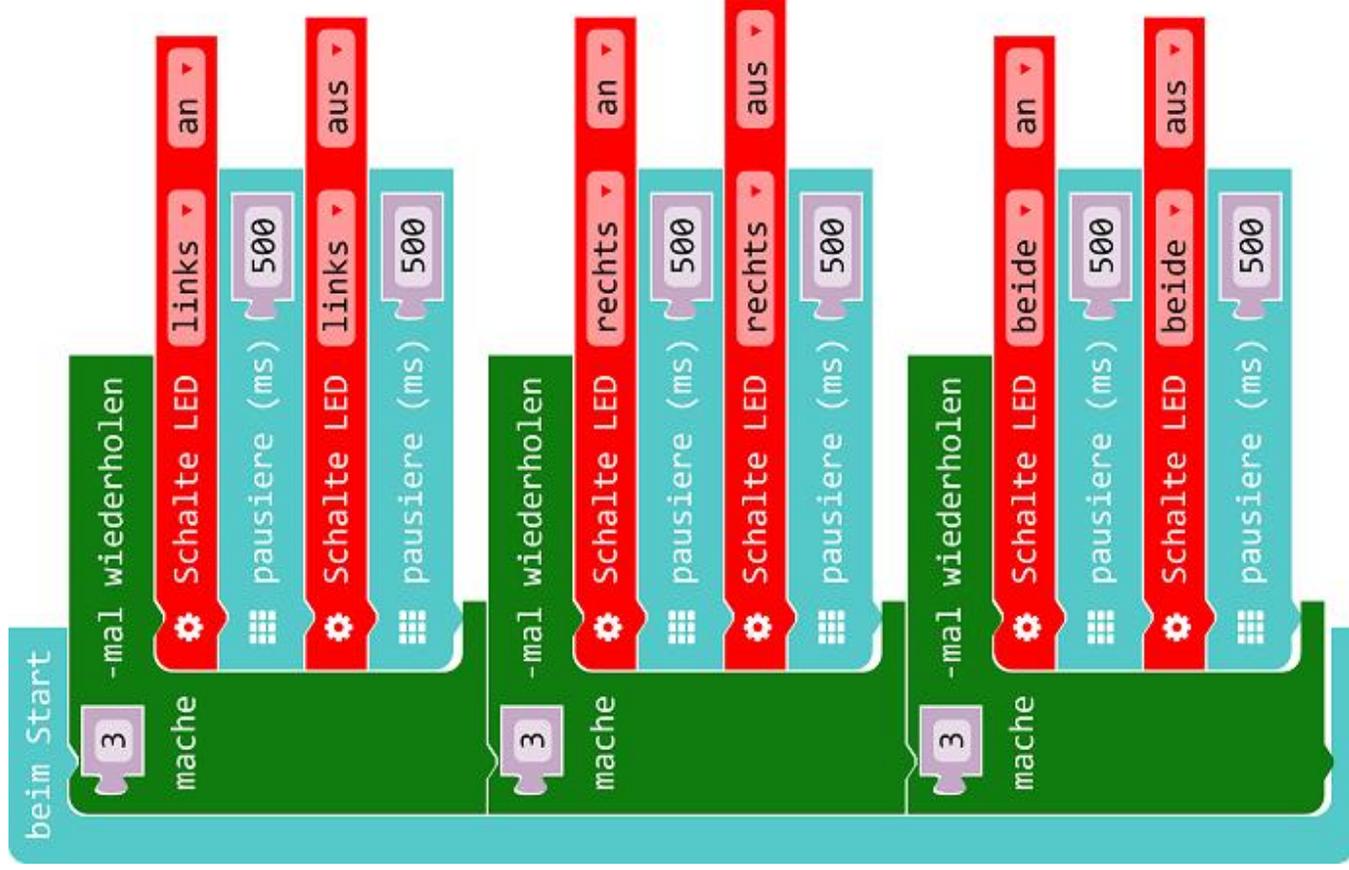


Werde Kreativ!

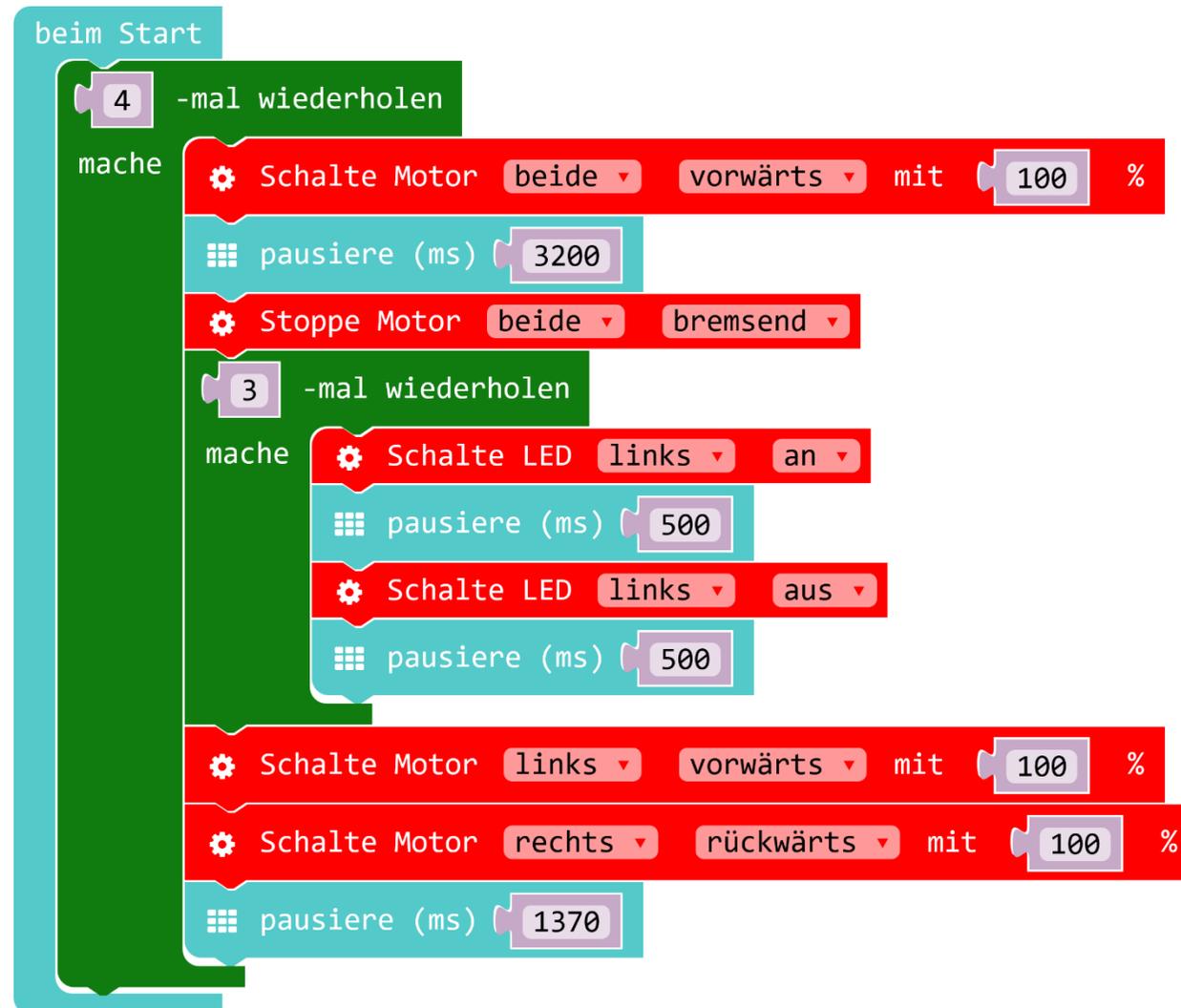
- Hast Du schon die Herausforderung „Labyrinth“ gelöst? Dann kombiniere diese doch mit dem Blinken. Vor jeder Kurve soll der *Calli:bot* 3 mal blinken!
- Denke Dir selbst aus wann die Lichter leuchten oder blinken sollen.

Calli:bot Herausforderung Blinken

(mögliche Lösung)

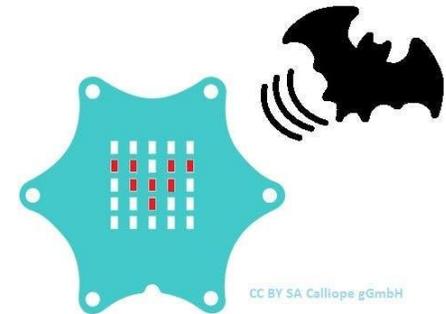
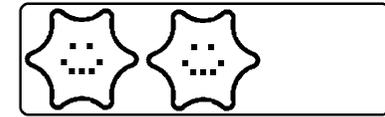


Calli:bot Herausforderung Blinken **Kreativ** (mögliche Lösung)



Calli:bot Herausforderung Ultraschallsensor

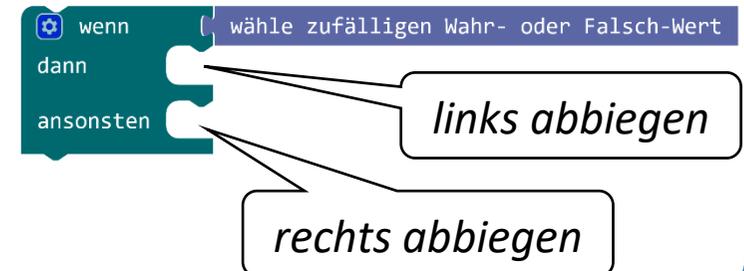
Wusstest Du, dass Fledermäuse mit Ultraschall Insekten orten und Entfernungen messen können? Der Calliope Mini kann das auch mit dem Ultraschallsensor auf dem *Calli:bot*.



- (1) Programmiere den *Calli:bot* mit dem Entfernungsblock und einer wenn-dann-Abfrage so, dass der *Calli:bot* stehen bleibt, wenn ein Hindernis z.B. 15 cm entfernt ist. 
- (2) Intelligenter wäre es, wenn der *Calli:bot* dem Hindernis z.B. nach links ausweicht, wenn er sich einem Hindernis nähert. Hilf deinem *Calli:bot* dabei!

Werde Kreativ!

- Erzeuge einen zufälligen Wahr- oder Falsch-Wert (aus dem Mathematikmenü) und weiche dann dem Hindernis entweder links oder rechts aus.
- Vergiss nicht vorher zu blinken!
- Du kannst auch gerne die RGB-LEDs dabei leuchten lassen.



Calli:bot Herausforderung Ultraschallsensor (mögliche Lösung)

```

dauerhaft
  wenn (Entfernung cm > 15)
  dann
    Schalte Motor beide vorwärts mit 100 %
  ansonsten
    Stoppe Motor beide auslaufend

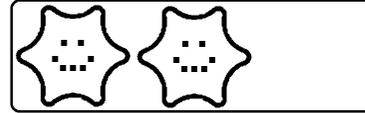
```

```

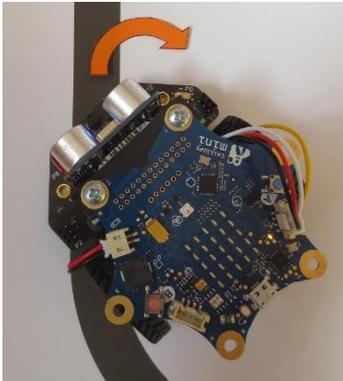
dauerhaft
  wenn (Entfernung cm > 15)
  dann
    Schalte Motor beide vorwärts mit 80 %
  ansonsten
    Stoppe Motor beide auslaufend
    Schalte Motor links vorwärts mit 100 %
    Schalte Motor rechts rückwärts mit 100 %
  pausiere (ms) 300

```

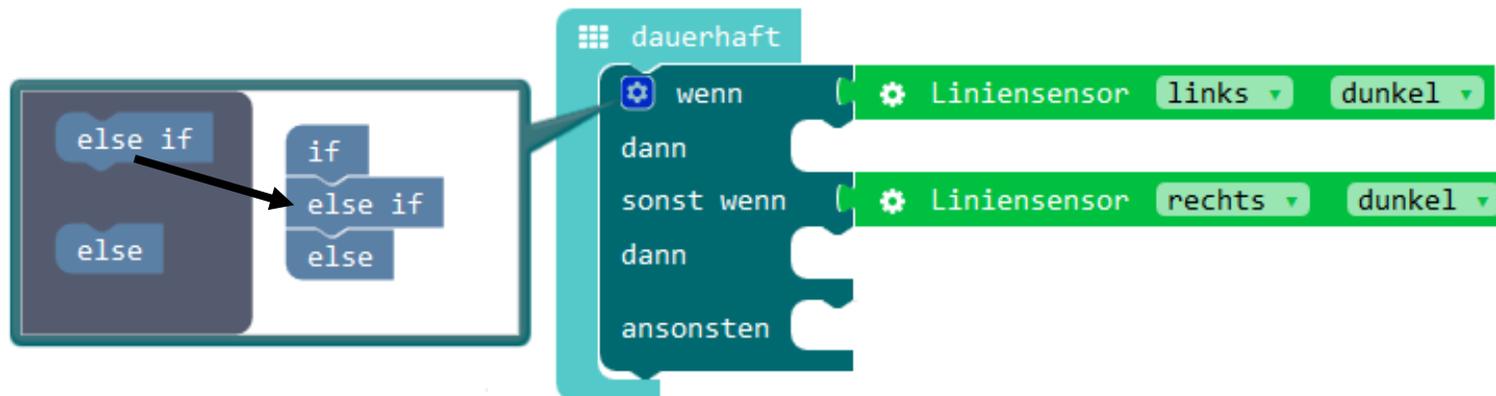
Calli:bot Herausforderung Arena



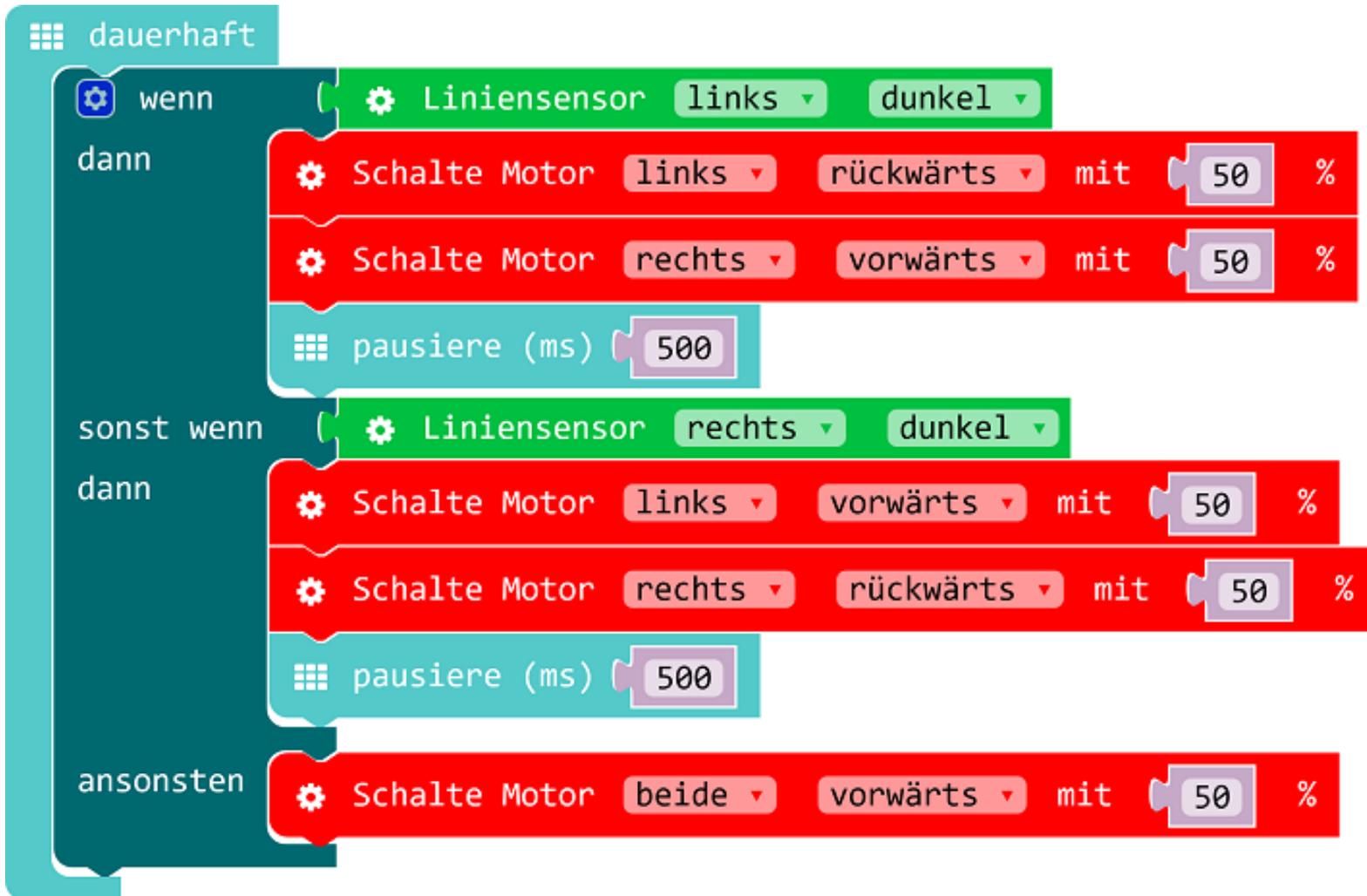
Der *Calli:bot* besitzt auf der Unterseite 2 Sensoren die erkennen können, ob der Untergrund hell oder dunkel ist. Diese wollen wir nutzen um den *Calli:bot* in der „Arena“ herumfahren zu lassen ohne dass er die schwarze Umrandungslinie überfährt. Wenn der linke Liniensensor die dunkle Linie erkennt, soll sich der *Calli:bot* nach rechts drehen, wenn der rechte Sensor den dunklen Untergrund erkennt, soll sich der *Calli:bot* nach links drehen. Ansonsten fährt der *Calli:bot* gradeaus.



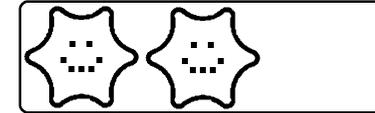
Für diese Herausforderung musst Du die „wenn-dann“-Abfrage um ein „sonst wenn“ erweitern. Klicke dazu auf das Zahnrad und ziehe ein „else if“ zwischen „if“ und „else“



Calli:bot Herausforderung Arena (mögliche Lösung)



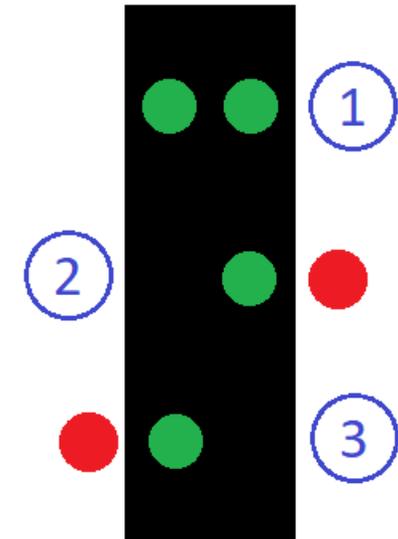
Calli:bot Herausforderung Linienfolger



Nun soll der *Calli:bot* immer der schwarzen Linie folgen **1** und wenn einer der beiden Sensoren auf den hellen Untergrund kommt nach links **2** oder nach rechts **3** auf die Linie zurücklenken.

Tipps:

- Erweitere die „wenn-dann“-Abfrage wie in der Herausforderung „Arena“
- Verknüpfe die beiden Zustände der Liniensensoren mit 
- Drehe den *Calli:bot* nur mit 50% Geschwindigkeit



Werde Kreativ!

- Überprüfe mit dem Ultraschallsensor ob ein Hindernis im Weg ist. Überlege, was dein *Calli:bot* dann tun soll!
- Lasse den *Calli:bot* mit den LEDs die Richtung anzeigen in die er fährt
- Lasse die RGB-LEDs leuchten. Du kannst dazu eine zweite dauerhaft-Klammer verwenden!

Calli:bot Herausforderung Linienfolger (mögliche Lösung)

```

dauerhaft
wenn
  Liniensensor links dunkel und Liniensensor rechts dunkel
dann
  Schalte Motor beide vorwärts mit 100 %
sonst wenn
  Liniensensor links dunkel und Liniensensor rechts hell
dann
  Stoppe Motor links auslaufend
  Schalte Motor rechts vorwärts mit 50 %
ansonsten
  Schalte Motor links vorwärts mit 50 %
  Stoppe Motor rechts auslaufend

```

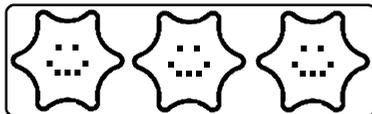
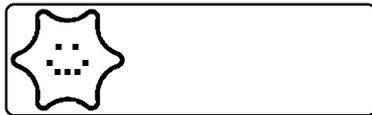
Weitere Ideen:

Herausforderung im Abstand hintereinanderfahren

Herausforderung Fernsteuerung einfach / schwer

Herausforderung synchron tanzen

Schwierigkeitsgrade:



Diese Arbeitskarten entstanden zu einem „meet and code“ Event meiner Klasse und sind so gedacht, dass keine Vorkenntnisse der Schüler nötig sind. Zuerst sollten allerdings die einfachen Codingcards bearbeiten da bei den schwereren wenig Hilfen gegeben sind.

Viel Spaß Michael Klein <https://twitter.com/kleinswelt>

