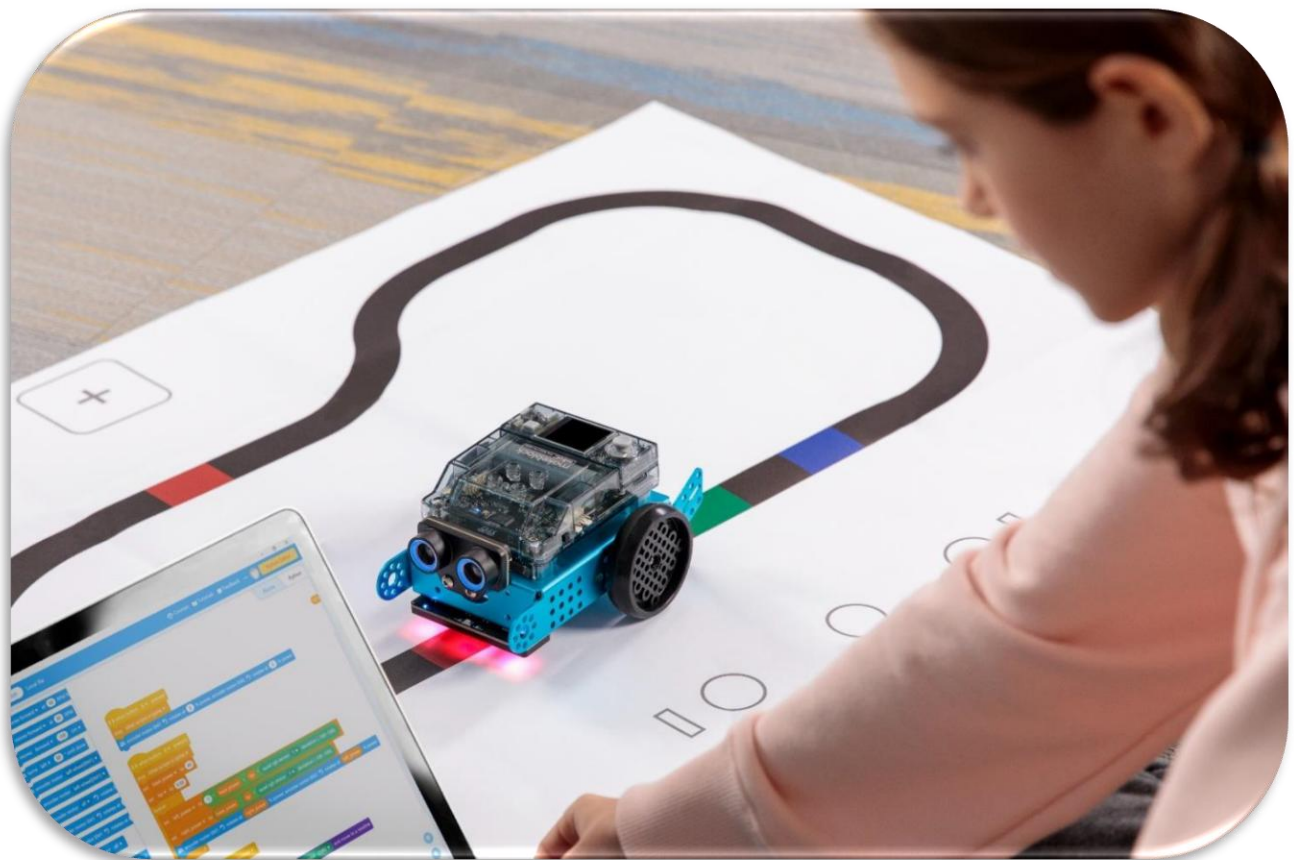


# mBot2

# TUTORIAL



# Inhalt

1	Notwendige Software.....	4
1.1	Die Programmoberfläche.....	7
1.2	Basisgerät und Erweiterungen.....	8
2	Verbindung herstellen .....	10
2.1	Verbindung über USB-Kabel: .....	10
2.2	Verbindung über Bluetooth:.....	11
3	Interaktive Verbindung („Spielmodus“): .....	14
4	Einführende Aufgaben:.....	18
4.1	Grundlegende Aufgaben:.....	18
4.1.1	Geradlinige Bewegungen:.....	18
4.1.2	Drehungen: .....	19
4.1.3	Signale:.....	20
4.2	Einfache Bewegungsaufgaben:.....	20
4.3	Verwendung der Sensoren des mBot2: .....	21
4.3.1	Lichtsensoren:.....	22
4.3.2	Schallsensoren: .....	23
4.3.3	Gyroskop/Beschleunigungssensoren:.....	23
4.3.4	Ultraschallsensoren: .....	24
4.3.5	4-fach RGB Sensoren:.....	25
5	Komplexere Aufgaben: .....	27
5.1	Einsatz von Variablen:.....	27
5.2	Quadrat: .....	27
5.3	Regelmäßige Vielecke: .....	28
5.4	Roboter folgt (Haustier):.....	28
5.5	Line-Follower: .....	29
5.6	Chamäleon: .....	32
5.7	Der Roboter tanzt: .....	33
6	Musterlösungen der Karten: .....	35

---

6.1	Aufgabe 1 .....	35
6.2	Aufgabe 2 .....	35
6.3	Aufgabe 3 .....	35
6.4	Aufgabe 4 .....	36
6.5	Aufgabe 5 .....	36
6.6	Aufgabe 6 .....	37
6.7	Aufgabe 7 .....	37
6.8	Aufgabe 8 .....	38
6.9	Aufgabe 9 .....	38
6.10	Aufgabe 10 .....	39
6.11	Aufgabe 11 .....	40
6.12	Aufgabe 12 .....	41
7	Abbildungsverzeichnis: .....	42

# 1 NOTWENDIGE SOFTWARE

Der Roboter kann über die Software mBlock programmiert werden:

The screenshot shows the mBlock software download page with three main sections:

- mBlock web version:** Includes a panda logo with gears, the text "mBlock web version", "Chrome browser recommended >>", and "Support Windows/Mac/Linux/Chromebook". It features two buttons: "Code with blocks" and "Code with Python".
- mBlock PC version:** Includes the same panda logo, the text "mBlock PC version", "Version: V5.4.0", "Released: 2021.11.30", and "Released log >> Previous version >>". It features two buttons: "Download for Windows" (with subtext "Win7 or Win10 (64-bit recommended)") and "Download for Mac" (with subtext "macOS 10.12+").
- mBlock mobile app:** Includes a mobile app icon with the panda logo, the text "mBlock mobile app", and "Learn coding in phones and tablets". It features two QR codes: one for "Android" (with subtext "Android 6.0 + (ARM-based devices only, X86 Android not supported)") and one for "iOS" (with subtext "iOS 10.0 +").

Abbildung 1: Programmversionen

Hier existiert eine Programmversion für Windows und Mac. Alternativ kann auch direkt aus dem Browser (Chrome wird empfohlen) die Oberfläche gestartet werden. Für mobile Devices (Handy, Tablets) gibt es eine APP-Version für Android und für IOS.

Der mBot2 kommt mit folgenden Basiskomponenten:

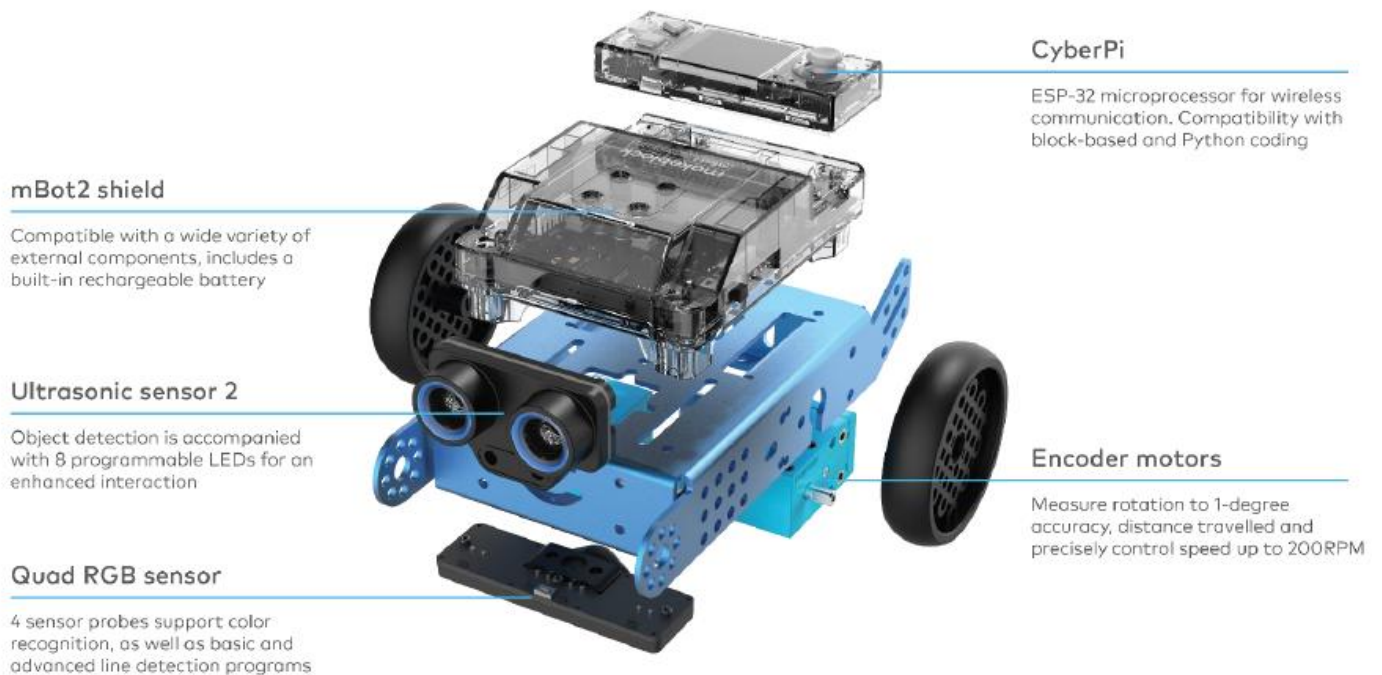


Abbildung 2: mBot2 - Grundstruktur

Das Herzstück ist der CyperPi Mikrocontroller über den auch die Programmierung läuft. Zusätzliche Komponenten der Grundausstattung sind:

- der mBot2-Rahmen (zur Befestigung weiterer Komponenten)
- der Ultraschallsensor 2 (mit 2 blauen LEDs)



Abbildung 3: Ultraschallsensor

- der 4fach RGB Sensor (zur Farb- und Linienerkennung, Weiterentwicklung des 2fach Sensors bei mBot1)

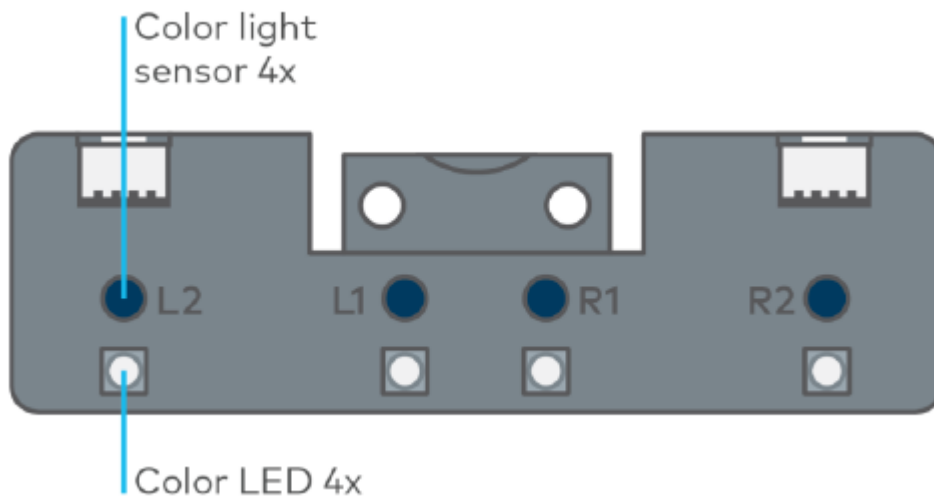


Abbildung 4: RGB Sensoren

- Encoder-Motoren (präzisere Steuerung als beim Vorgängermodell)

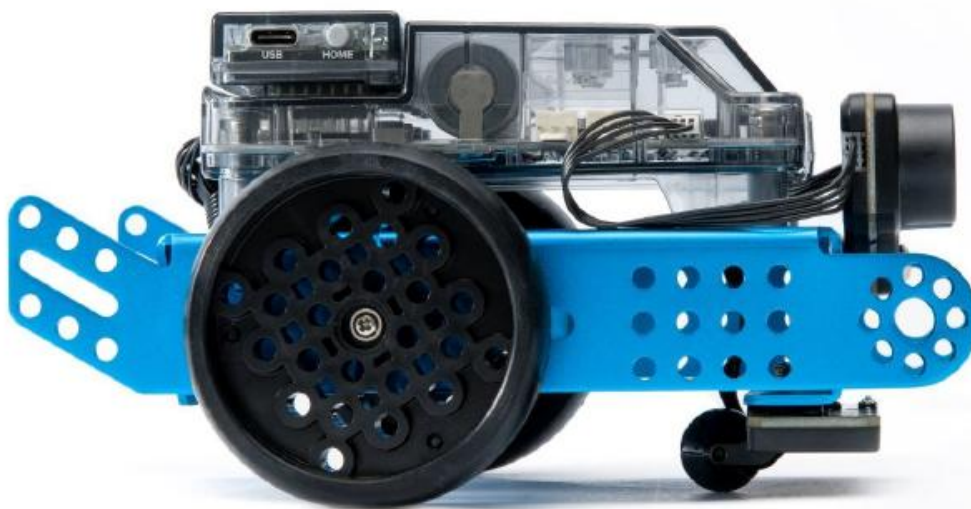


Abbildung 5: Encoder-Motoren

## 1.1 DIE PROGRAMMOBERFLÄCHE

Die Programmoberfläche mBlock zeigt zunächst eine Scratch-Oberfläche mit den entsprechenden Komponenten (nach der Installation der notwendigen Erweiterungen)

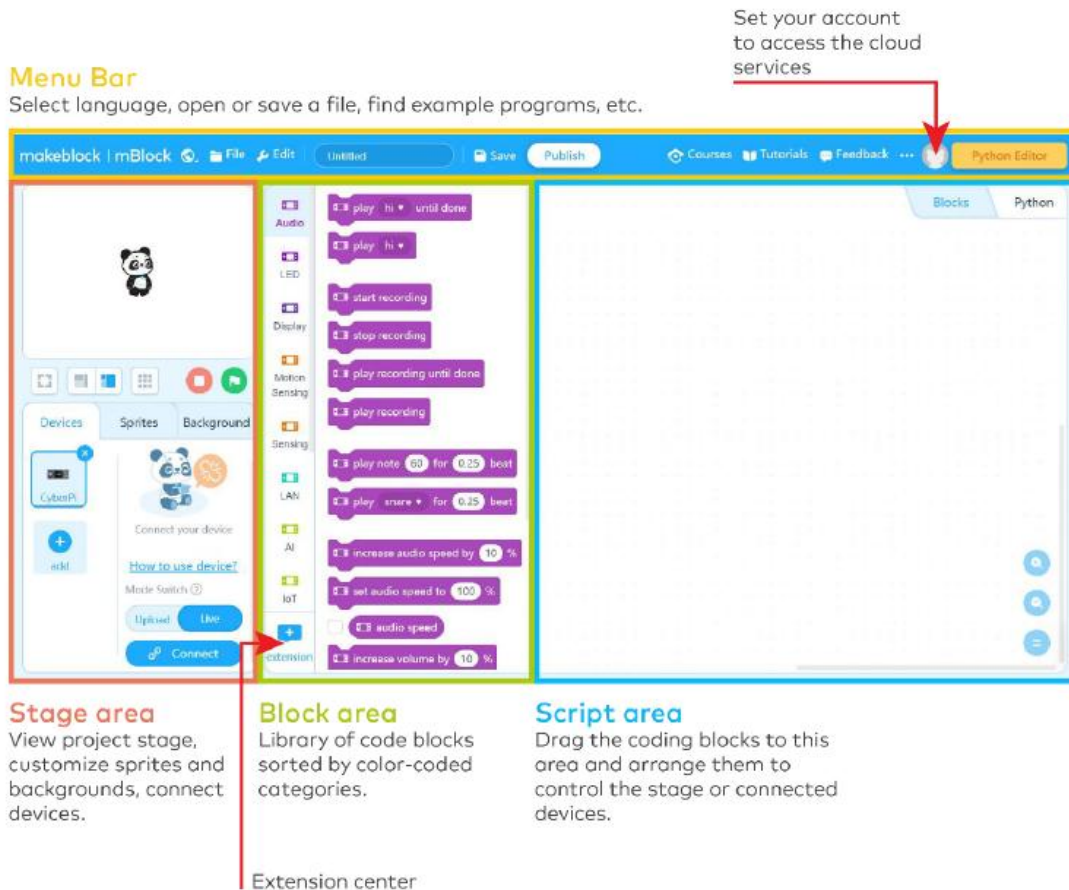


Abbildung 6: Programmoberfläche

Neben der blockorientierten (Scratch-)Oberfläche kann der Roboter auch direkt mit Python programmiert werden:

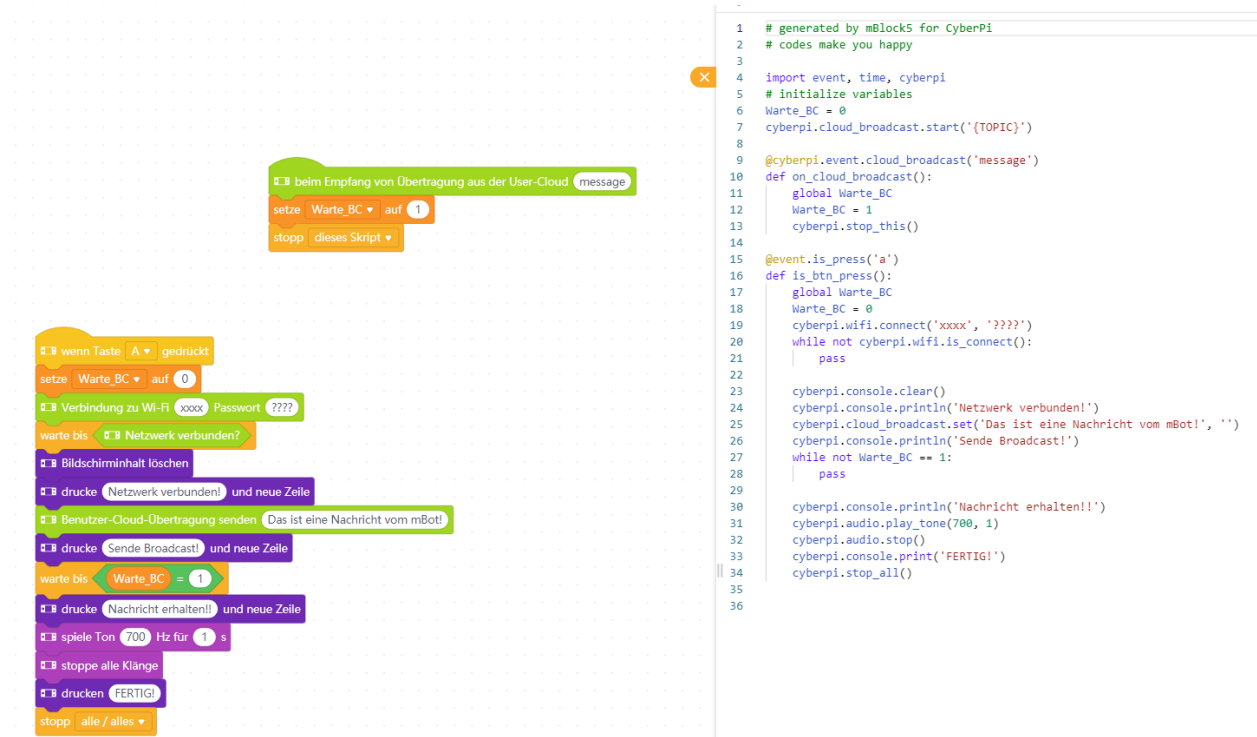


Abbildung 7: Block- oder Pythonoberfläche?

## 1.2 BASISGERÄT UND ERWEITERUNGEN

Um nun den mBot2 (und dessen Basiskomponenten) aus der Programmieroberfläche entsprechend ansprechen zu können wird nun zunächst der Controller (CyberPi) als Gerät hinzugefügt:

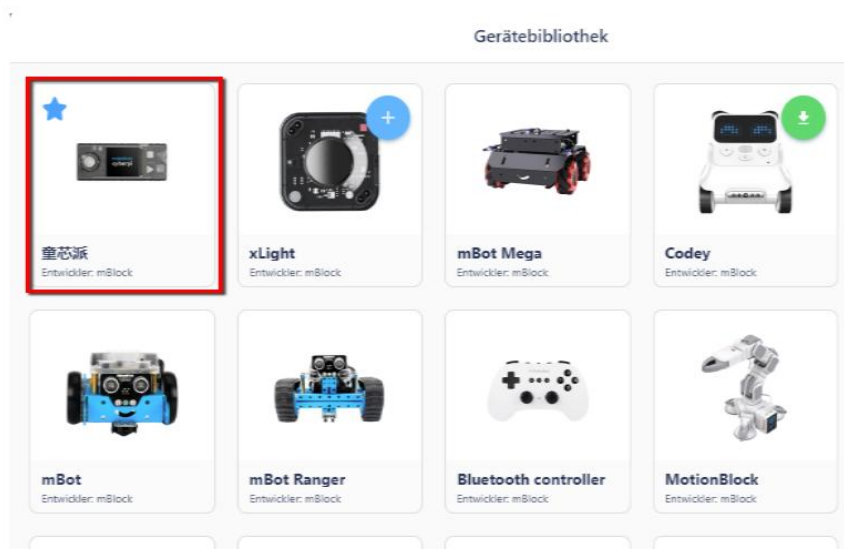


Abbildung 8: CyberPi hinzufügen

Danach müssen noch die vorhandenen Sensoren/Motoren als Erweiterungen hinzugefügt werden:

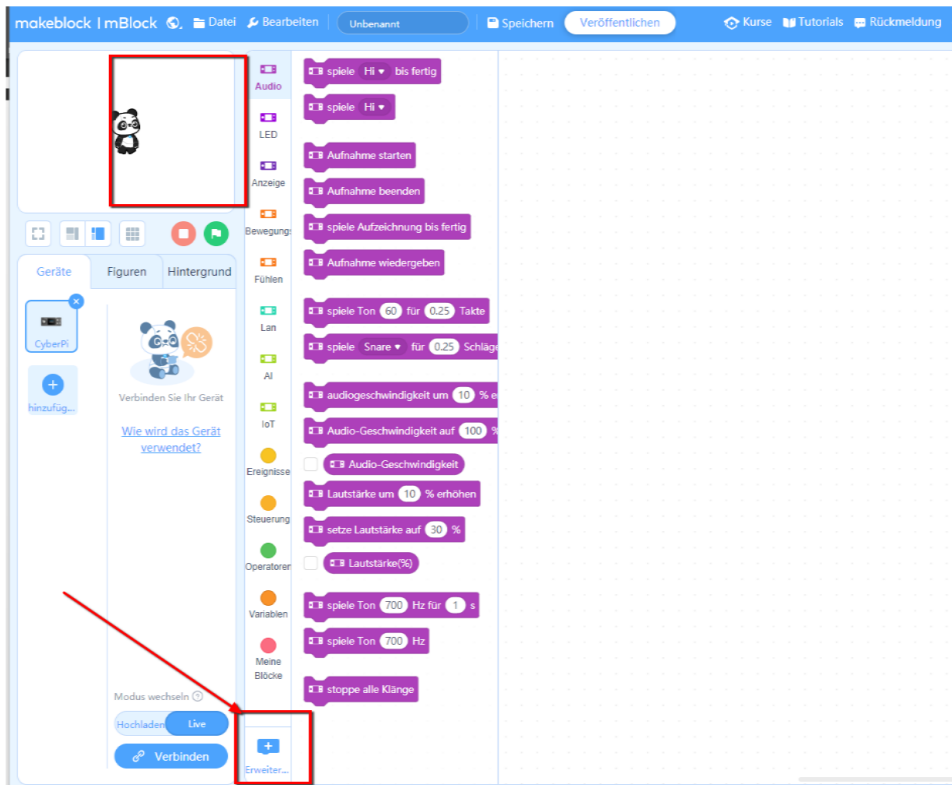
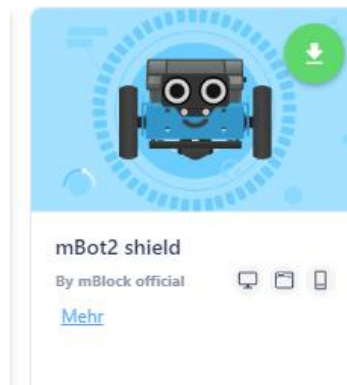


Abbildung 9: Erweiterungen hinzufügen

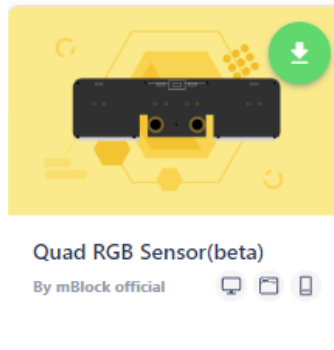
Aus den angebotenen Erweiterungen werden nun folgende installiert:

- mBot2-Chassis (Motorensteuerung):



- Ultraschallsensor 2:





- QuadRGB-Sensor:

## 2 VERBINDUNG HERSTELLEN

Die Verbindung zwischen der Programmieroberfläche (PC, Notebook, Tablett, ...) kann auf 3 Arten hergestellt werden: (Dabei wird noch grundsätzlich unterschieden, ob das Gerät im Livemodus oder

### 2.1 VERBINDUNG ÜBER USB-KABEL:



Abbildung 10: CyperPi über USB verbinden

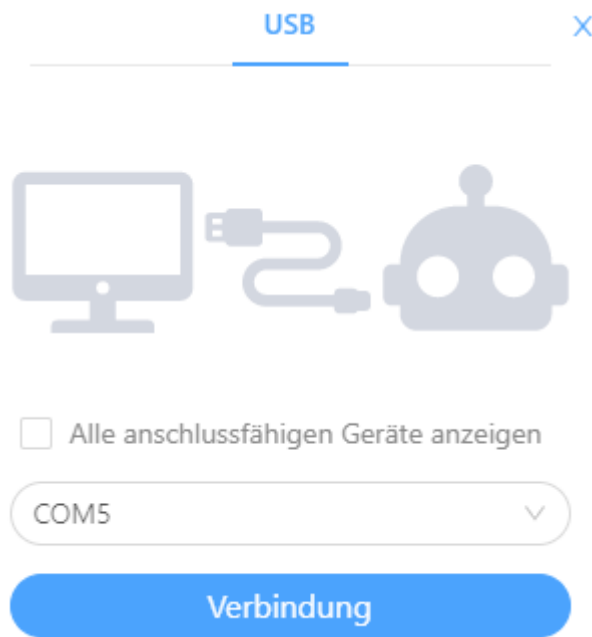


Abbildung 12: Schnittstelle auswählen

Das Gerät ist nun verbunden. Über die Einstellungen kann nun der Hostname (wichtig für Bluetooth und WLAN Verbindungen) und die WLAN Einstellung vorgenommen werden:

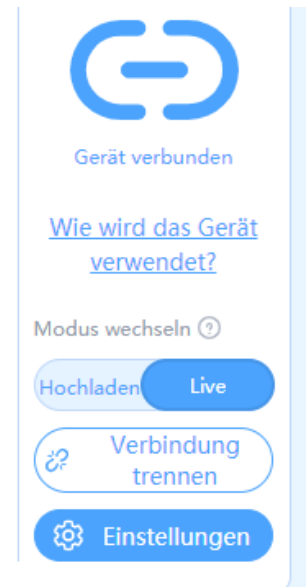


Abbildung 11: USB Verbindung funktioniert

## 2.2 VERBINDUNG ÜBER BLUETOOTH:



Funktioniert für Mobilgeräte (oder am PC über speziellen Bluetooth-Adapter). Auf den Mobilgeräten (Tablets, iPads) sollte die App „makeblock / mBlock“ installiert sein.

Verbindung

Bluetooth einschalten:

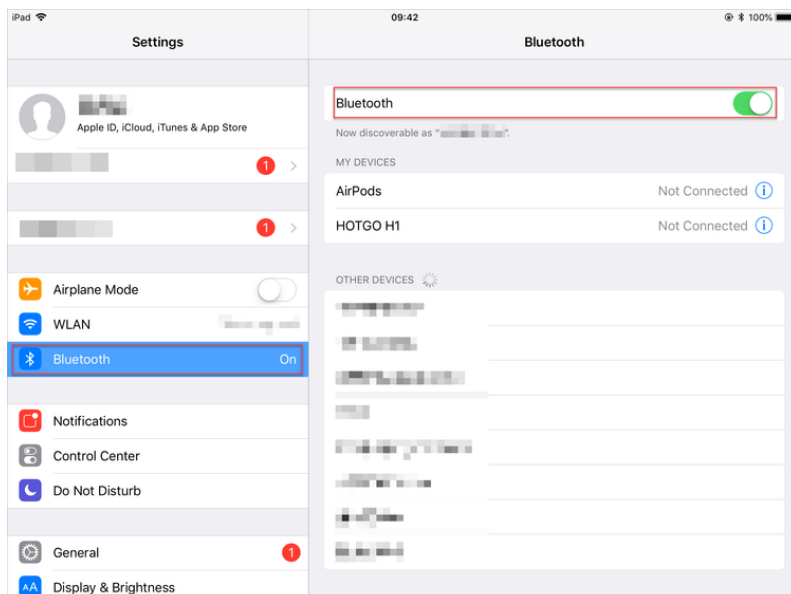


Abbildung 13: Bluetooth am iPad aktivieren

In der App mBlock den Roboter (cyperPi) wählen:

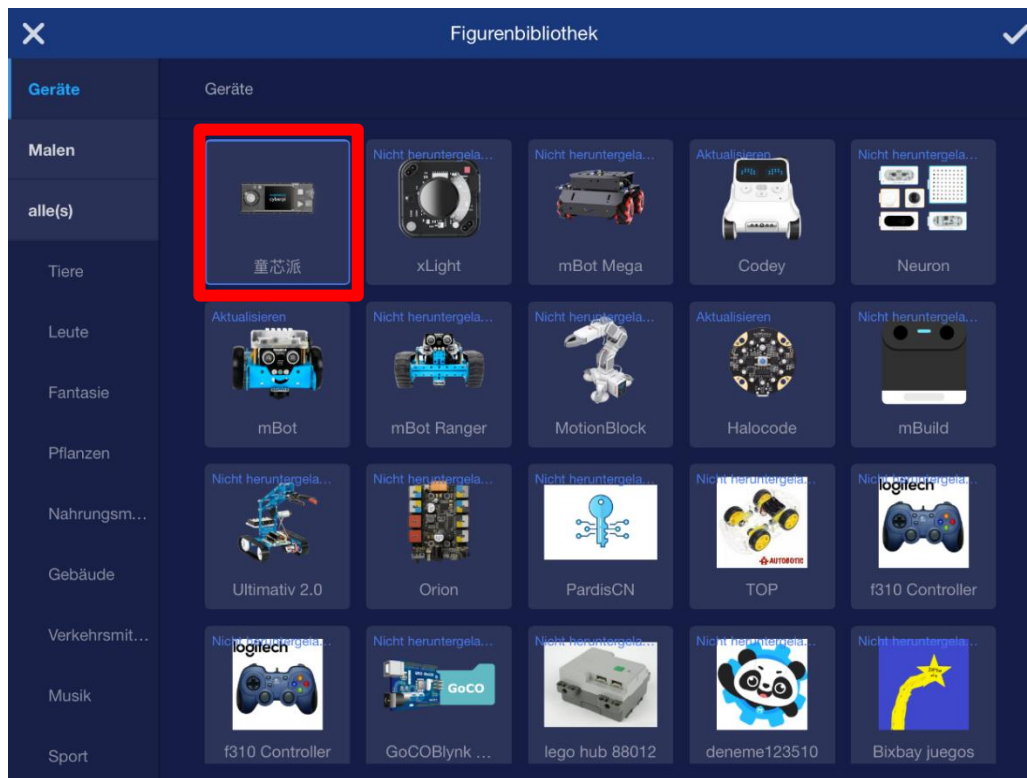


Abbildung 14: CyperPi als Gerät auswählen

Erweiterungen (mBot2, Ultraschallsensor, QuadRGB Sensor) installieren:

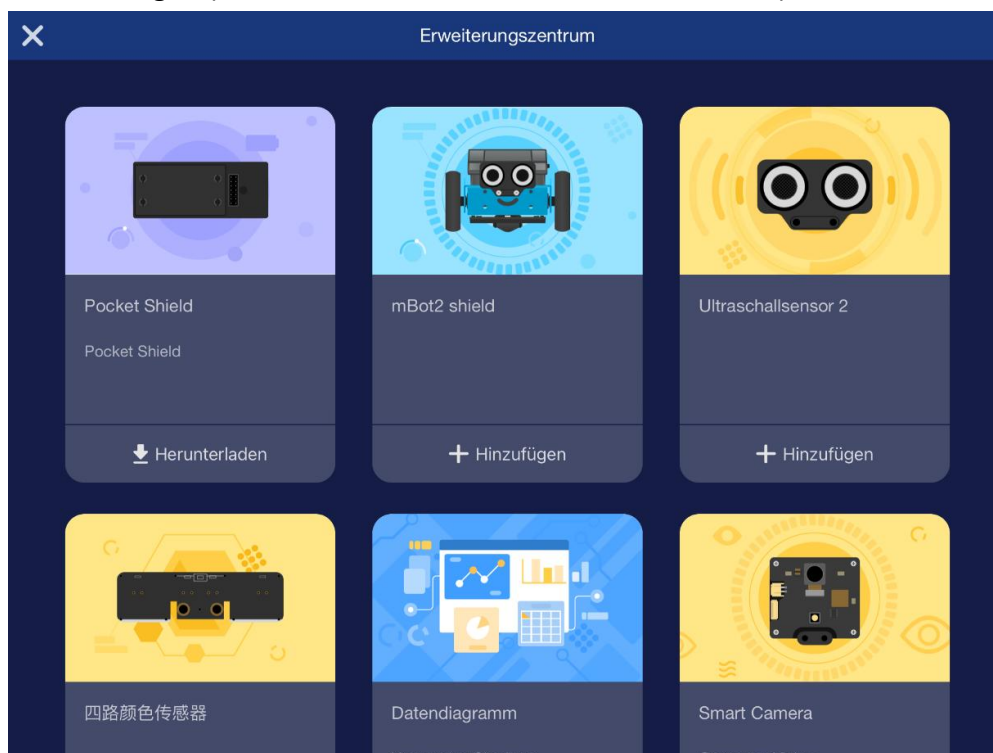


Abbildung 15: Erweiterungen hinzufügen

### CyperPi via Bluetooth verbinden:

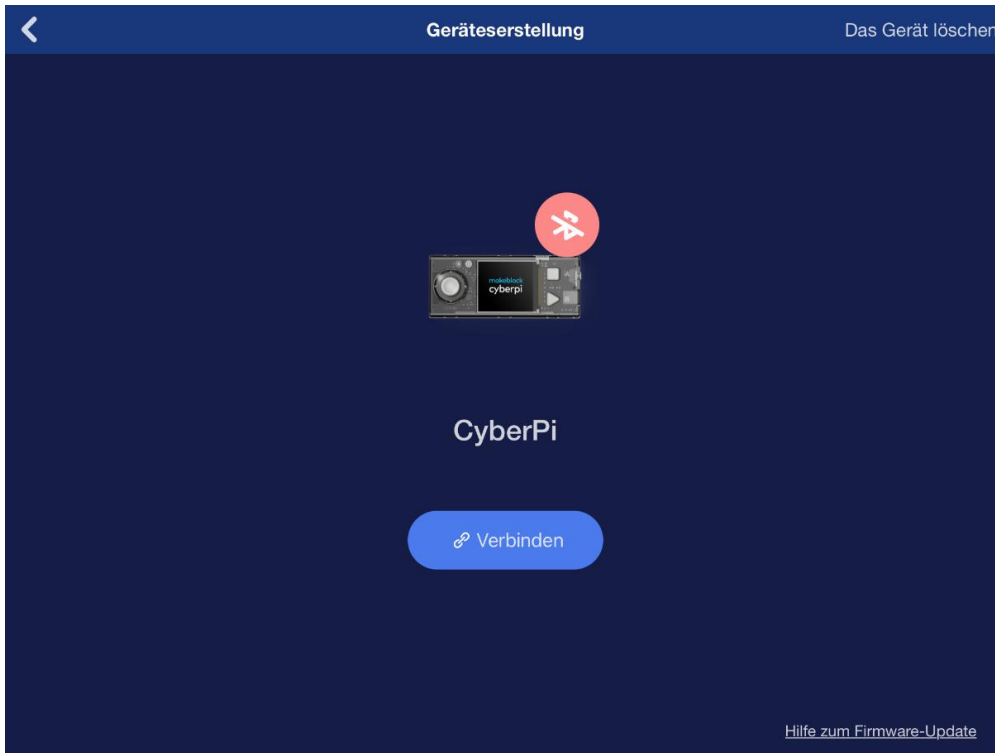


Abbildung 16: mit Bluetooth verbinden

### Verbindung herstellen:



Abbildung 17: Pairing

Verbindung hergestellt:

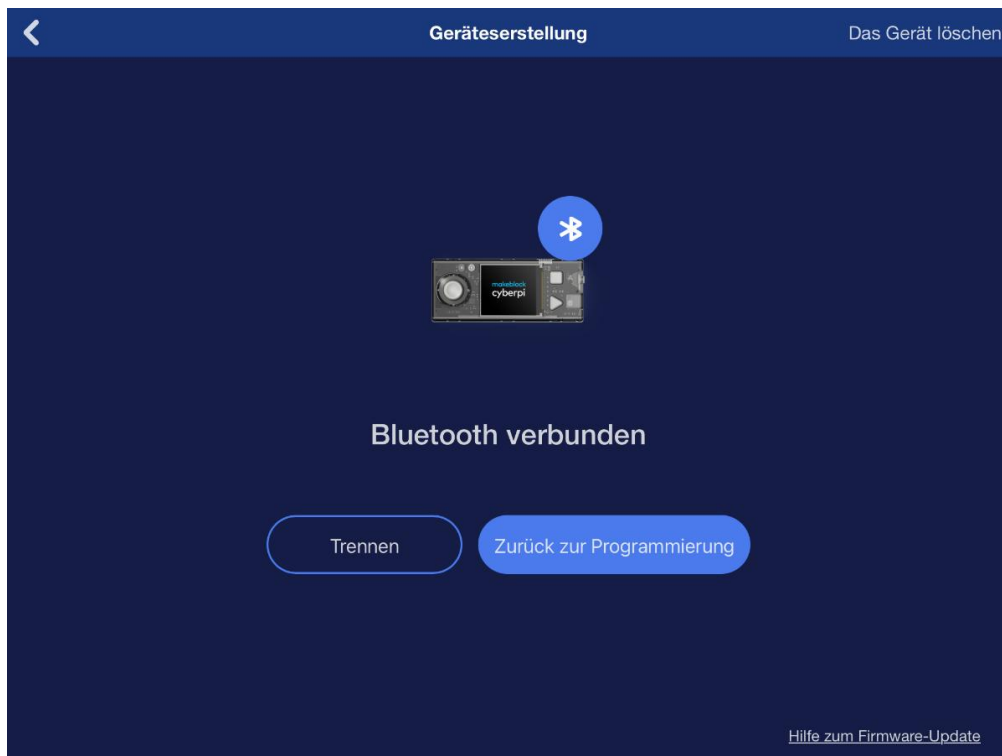


Abbildung 18: Bluetooth erfolgreich

### 3 INTERAKTIVE VERBINDUNG („SPIELEMODUS“):

Alternativ zum Programmiermodus (mBlock-APP) kann über das Tablet der Roboter auch im „Spielmodus“ (über die App makeblock) verbunden werden. Dabei wird das Tablet als Fernsteuerung für den Roboter verwendet:

Nach dem Starten der App kann über das Symbol  das zu verbindende Gerät ausgewählt werden.

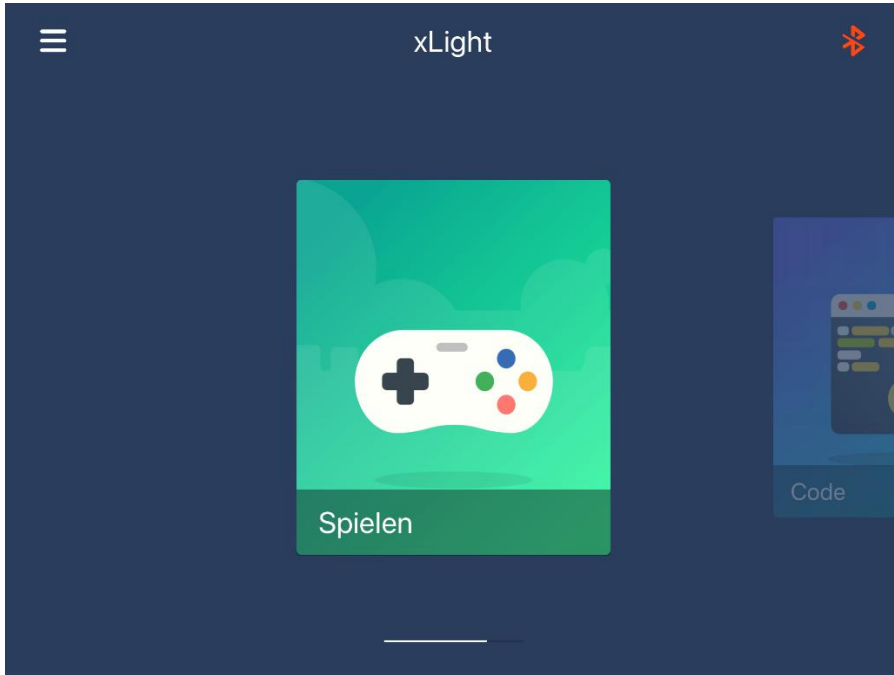


Abbildung 19: Spielemodus (Livemodus)

In unserem Fall wird hier der Roboter mBot2 ausgewählt.

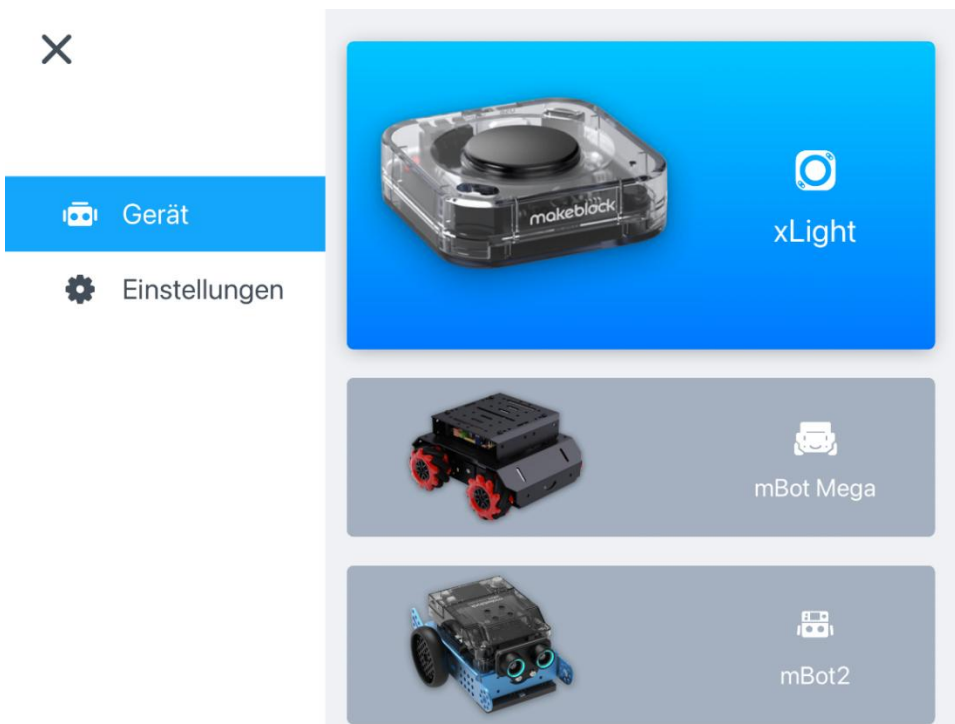



Abbildung 20: mBot2 als Gerät

Wenn dieser nun (statt xLight) aufscheint, kann durch einen Klick auf das rote Bluetoothsymbol (  ) die Verbindungsaufnahme gestartet werden:

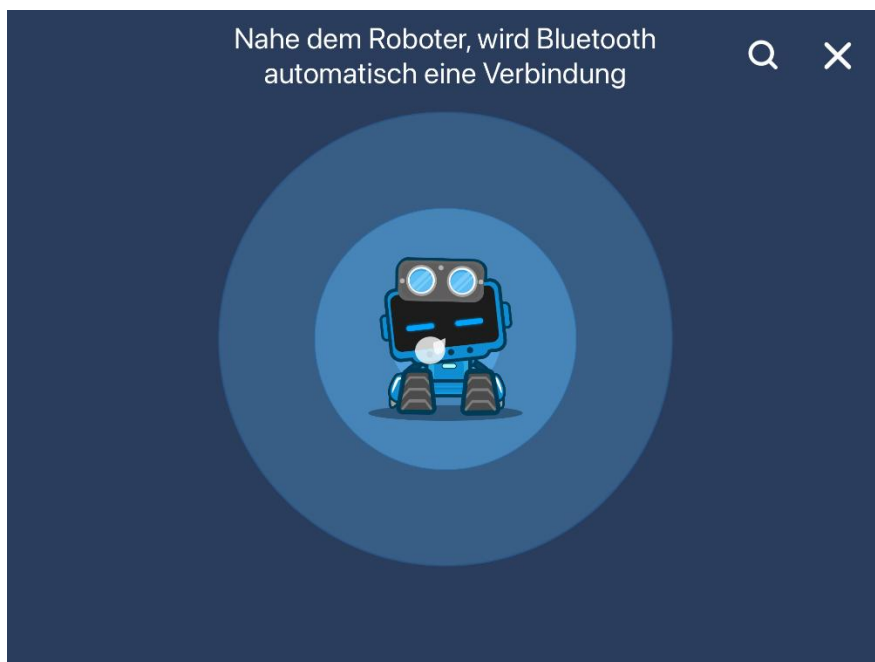


Abbildung 21: via Bluetooth verbinden

Das nun weiße Bluetoothsymbol zeigt den verbundenen Roboter.

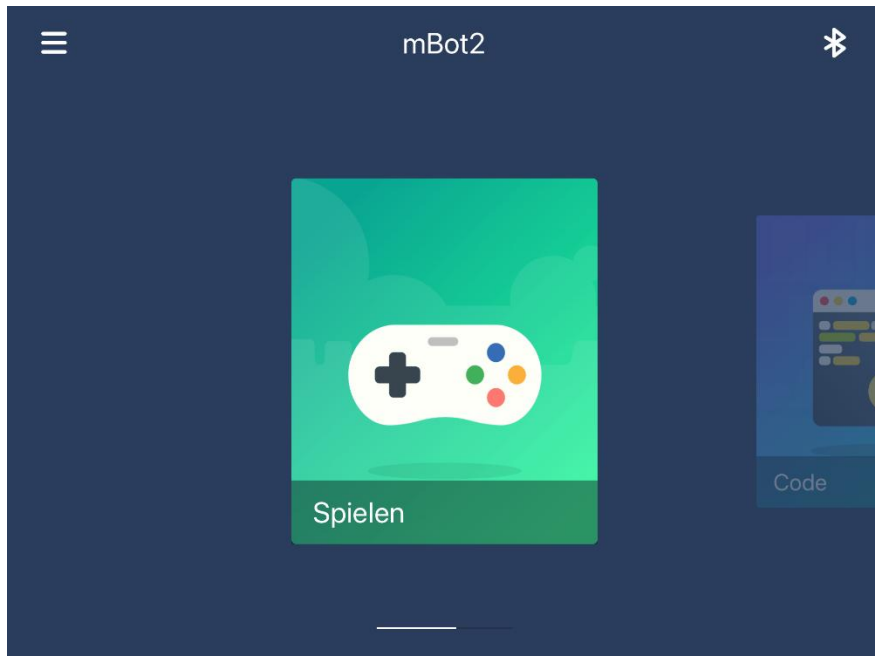


Abbildung 22: aktive Verbindung

Durch Auswahl von „Spielen“ => „Fahren“ wird das Tablet nun zur Fernsteuerung:



Abbildung 23: Spielemodus - Fahren



Abbildung 24: iPad als Fernbedienung

## 4 EINFÜHRENDE AUFGABEN:

### 4.1 GRUNDLEGENDE AUFGABEN:

Start: Das Programm soll jeweils durch drücken auf den Button „A“ (siehe Abbildung 25) gestartet werden.



Abbildung 25: Tasten am CyberPi

#### 4.1.1 GERADLINIGE BEWEGUNGEN:

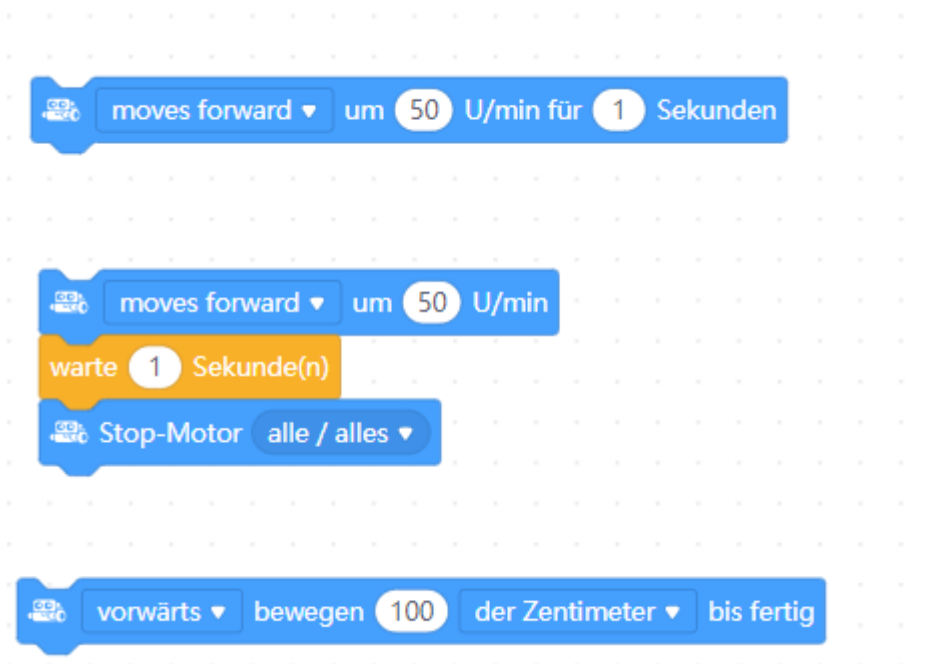


Abbildung 26: Befehle für geradlinige Bewegungen

**Aufgabe 1:** Versuche mit mehreren Befehlen den Roboter genau 25cm fahren zu lassen. Notiere dir die dazu passenden Daten (wie viele U/min – welche Zeit) – Teste die Genauigkeit mit längeren Strecken (1m, 2m)



Abbildung 27: Beispiellösung Aufgabe 1

**Forschungsaufgaben:** Liefern die Kombinationen 7U/min – 10s, 10U/min – 7s und 70U/min – 1s gleiche Ergebnisse? – Finde weitere Lösungen – Wie sieht es mit den Tests auf längere Distanzen aus? – Wie genau ist die eingebaute Funktion?

#### 4.1.2 DREHUNGEN:

Drehungen können durchgeführt werden, indem z.B. nur ein Motor angesteuert wird. Versuche auf unterschiedliche Arten eine Drehung um 90° (rechts oder links) zu erzeugen. Was ist der Unterschied zur Bewegung beim Befehl „Nach links drehen 90° bis fertig“?

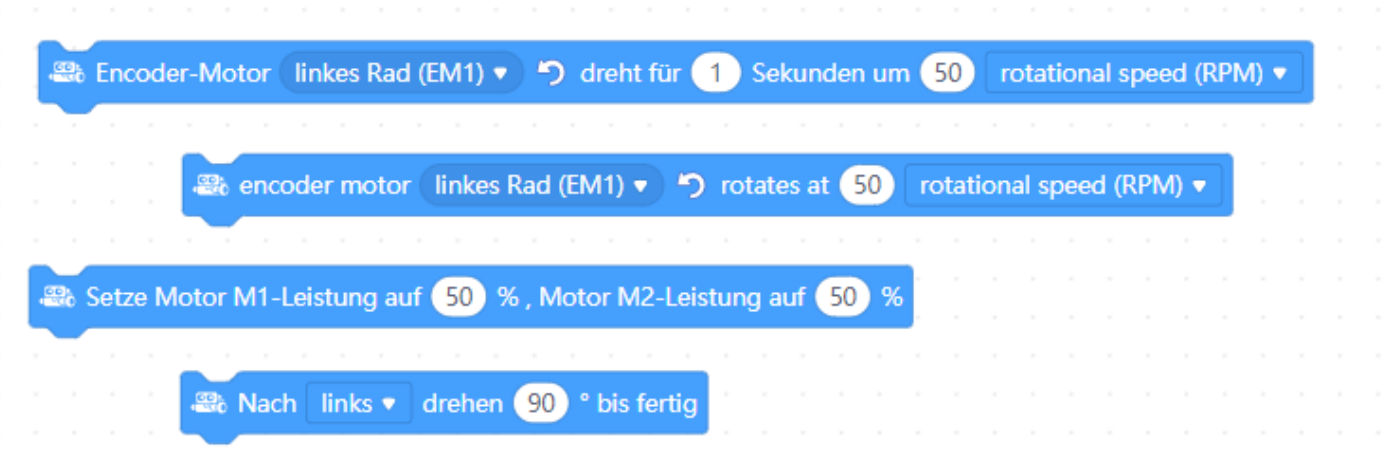


Abbildung 28: Basisbefehle zur Erzeugung von Drehungen

#### Aufgabe 2:

- Erzeuge eine Linksdrehung um 90° nur durch Ansteuerung des rechten Motors. Welche Daten zur Motoransteuerung funktionieren bei dir? (Findest du ev. mehrere Möglichkeiten?)
- Erzeuge eine Rechtsdrehung um 90° nur durch Ansteuerung des linken Motors. Welche Daten zur Motoransteuerung funktionieren bei dir? (Findest du ev. mehrere Möglichkeiten?)
- Vergleiche die Bewegungen aus den ersten beiden Aufgaben mit der Bewegung bei „Nach links (bzw. rechts) drehen 90° bis fertig“ – Wie könntest du diese Bewegung nachbauen?



Abbildung 29: Lösung Teilaufgabe von 2

### 4.1.3 SIGNALE:

Der Roboter kann Töne aussenden und auch Informationen am Display ausgeben. Die Basisbefehle siehst du in nachfolgender Abbildung. Probiere sie einmal aus:



Abbildung 30: Audio- und Bildschirmausgabe

## 4.2 EINFACHE BEWEGUNGSAUFGABEN:

Programmiere den Roboter so, dass er einem einfachen Bewegungsmuster (vgl. Abbildung 31) folgen kann. Notiere dir deine jeweilige Lösung der Aufgabenstellung.

Zusatzaufgabe: Schaffst du es eventuelle, dass der Roboter beim Start und beim Ankommen im Ziel jeweils ein passendes Signal (Audio) aussendet und/oder auch etwas Passendes am CyperPi anzeigt?



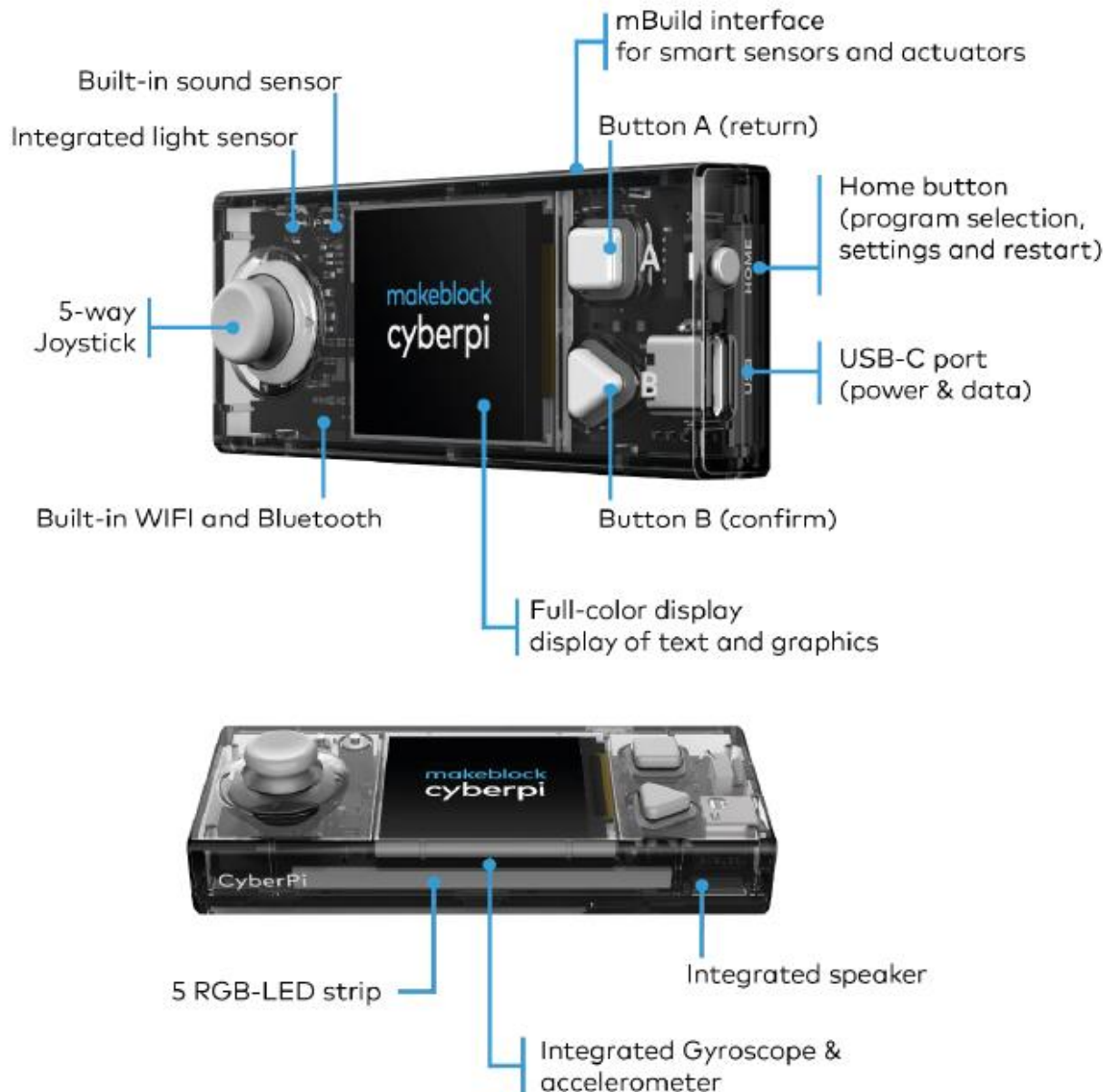


Abbildung 33: CyberPi – Funktionen

Hier findet man einen integrierten Lichtsensor, einen Schallsensor und einen Beschleunigungssensor, die entsprechend in Programmen auch angesprochen werden können. Die nachfolgenden Einführungsaufgaben sollen dir einige Grundfunktionen demonstrieren:

#### 4.3.1 LICHTSENSOR:

Im Beispiel wird die Umgebungshelligkeit im Display angezeigt:

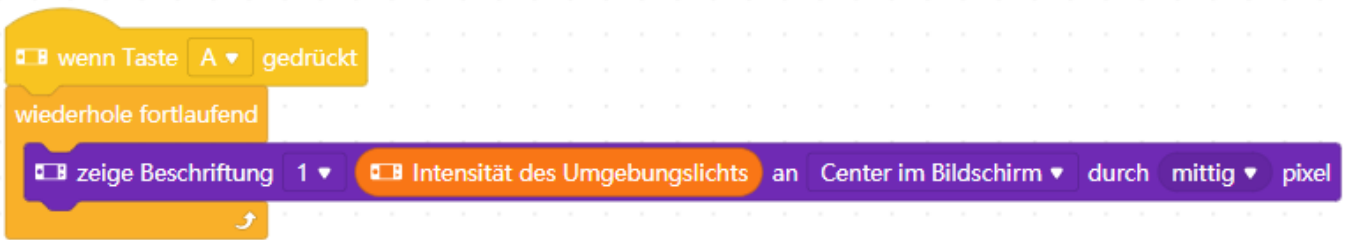


Abbildung 34: Verwendung des Lichtsensors

#### 4.3.2 SCHALLSENSOR:

Die Umgebungslautstärke soll am Display angezeigt werden:

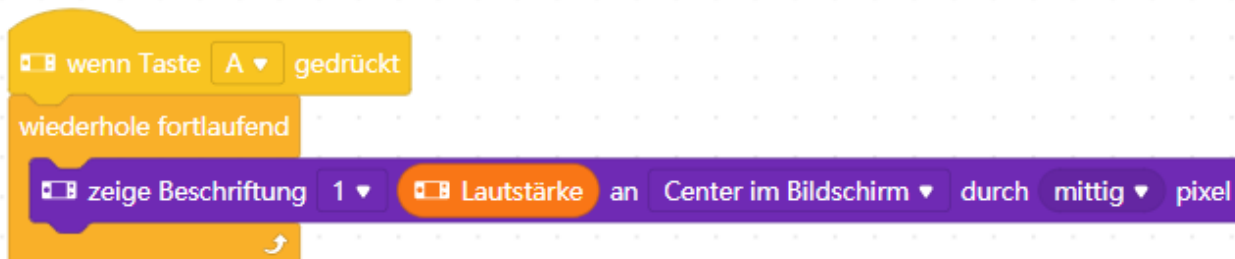


Abbildung 35: Verwendung des Schallsensors

#### 4.3.3 GYROSKOP/BESCHLEUNIGUNGSSENSOR:

Dient zum Anzeigen von Beschleunigungen bzw. der Lage (Neigung) des Roboters. Wenn der Roboter nach hinten kippt (oder bergauf fährt) wird am Display „UP!“ angezeigt.

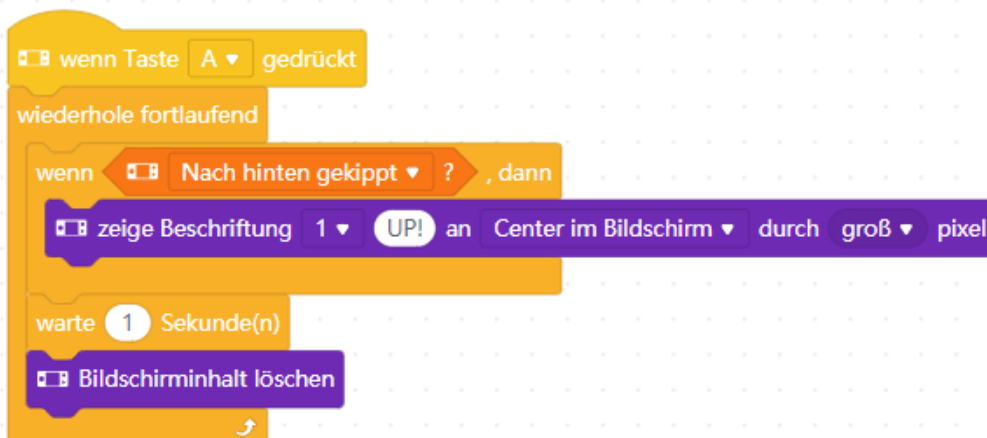


Abbildung 36: Gyroskop

Kannst du das Programm so anpassen, dass auch ein Kippen nach unten, rechts oder links angezeigt wird?

```

wenn Taste A gedrückt
wiederhole fortlaufend
  wenn Nach hinten gekippt ? , dann
    zeige Beschriftung 1 UP! an Center im Bildschirm durch groß pixel
  wenn Nach vorne gekippt ? , dann
    zeige Beschriftung 1 DOWN! an Center im Bildschirm durch groß pixel
  wenn Nach links gekippt ? , dann
    zeige Beschriftung 1 LEFT! an Center im Bildschirm durch groß pixel
  wenn Nach rechts gekippt ? , dann
    zeige Beschriftung 1 RIGHT! an Center im Bildschirm durch groß pixel
  warte 1 Sekunde(n)
  Bildschirminhalt löschen

```

Abbildung 37: Lösung Zusatzaufgabe Gyroskop

#### 4.3.4 ULTRASCHALLENSOR:

Das nachfolgende Beispiel zeigt die Verwendung des Ultraschallsensors. Damit kann der Roboter (nach vorne) Hindernisse erkennen. Das Programm wird wieder mit der Taste „A“ gestartet und läuft so lange, bis jemand auf die Taste „B“ drückt. Falls der Sensor einen Gegenstand ortet, der näher als 10cm ist, dreht sich der Roboter um 90° nach links und fährt weiter.

Sobald die Taste „B“ gedrückt wird, wird das Programm beendet.

- Was passiert, wenn du den Block „Encoder-Motor anhalten“ entfernst?
- Kannst du in der Anzeige immer die aktuelle Entfernung zum nächsten Hindernis anzeigen lassen?

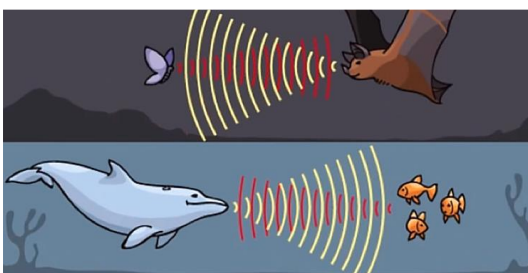


Abbildung 38: Ultraschall in der Tierwelt

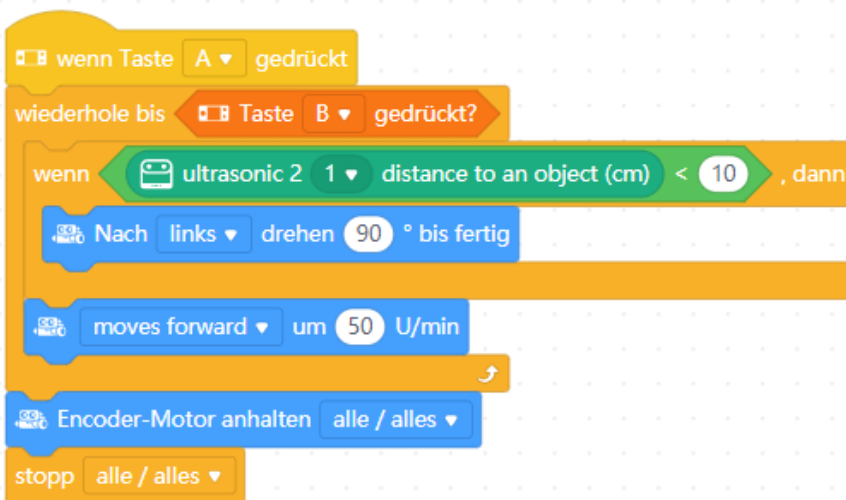


Abbildung 39: Ultraschallsensor (Entfernungsmessung)

#### 4.3.5 4-FACH RGB SENSOR:

Er besteht aus 4 Sensoren die mit R2, R1, L1, L2 (Rechts oder Links) beschriftet sind und auch unter diesen Bezeichnungen angesprochen werden können:

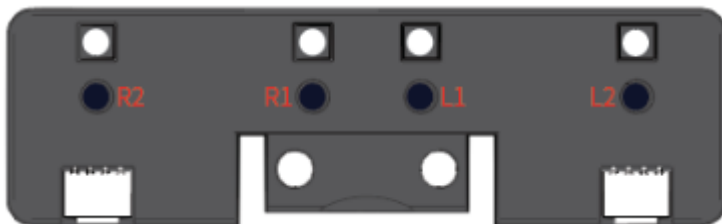


Abbildung 40: RGB-Sensoren

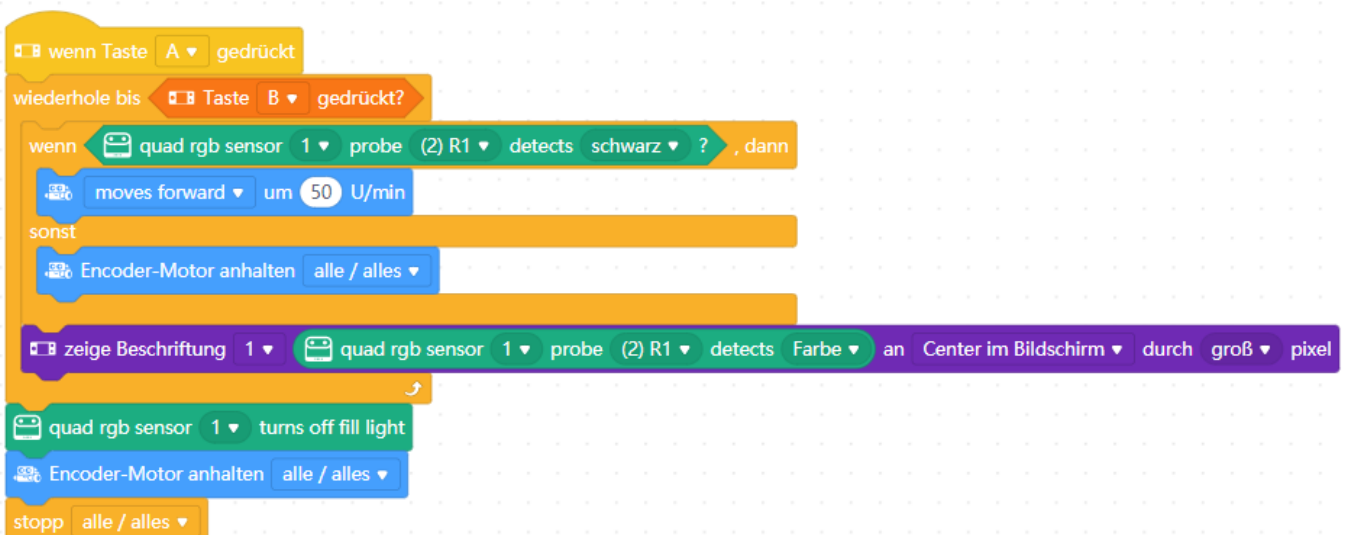


Abbildung 41: Beispiel mit RGB Sensoren

Das Beispiel (siehe Abbildung 41) zeigt eine Anwendung des RGB Sensors. Der Roboter fährt, solange der R1-Sensor die Farbe Schwarz sieht. Zusätzlich wird die aktuell festgestellte Farbe am Display angezeigt.



*Abbildung 42: Testmuster für RGB-Aufgabe*

## 5 KOMPLEXERE AUFGABEN:

### 5.1 EINSATZ VON VARIABLEN:

Durch einen Klick auf „Neue Variable“ (im Block Variablen) kann ein Platzhalter erstellt werden. Im folgenden Beispiel wird eine Variable „Seitenlaenge“ erzeugt und im Programm entsprechend verwendet. Die Bewegung des Roboters wird vom aktuellen Wert dieser Variablen abhängig gemacht. Was macht das nachfolgende Programm eigentlich?



Abbildung 43: Verwendung von Variablen

Kannst du das Programm so abändern, dass die aktuelle Seitenlänge am Display angezeigt wird?

### 5.2 QUADRAT:

Hinweis: In der Programmierung können auch Variable verwendet werden.

Programmiere den Roboter, dass er ein Quadrat mit 1m Seitenlänge abfährt. Achte auf entsprechende Präzision!

- Kannst du das Programm so abändern, dass die Seitenlänge des Quadrats 2m bzw. 50cm beträgt?
- Kannst du das Programm auch so ändern, dass eine Startseitenlänge von 100cm vorgegeben wird, diese aber durch den Joystick noch verändert werden kann (Anzeige zur Kontrolle) bevor der Roboter seine Bewegung startet?

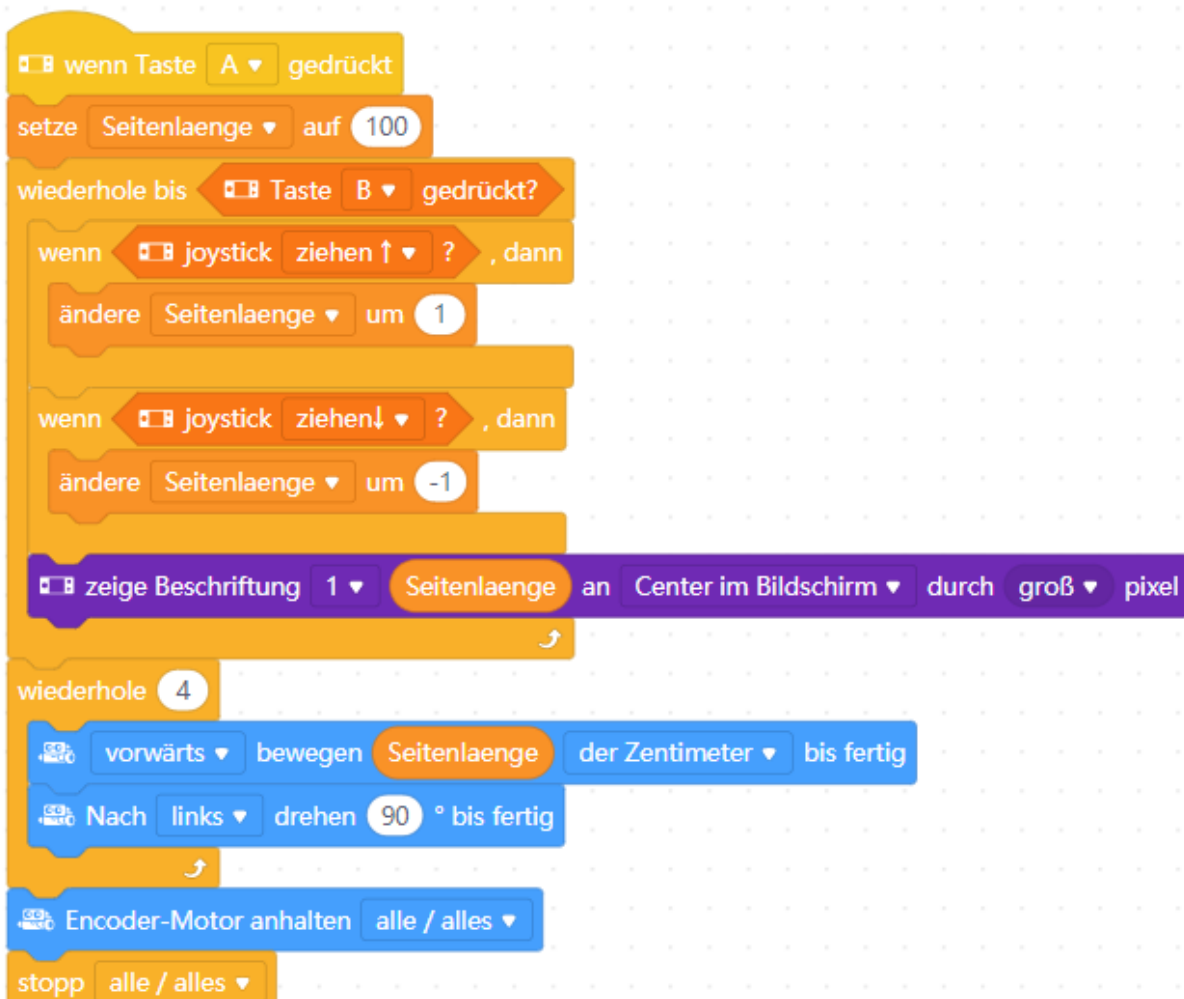


Abbildung 44: Lösung der Quadrataufgabe mit variabler Seitenlänge

### 5.3 REGELMÄßIGE VIELECKE:

Kannst du das Programm so abändern, dass statt des Quadrats ein gleichseitiges Dreieck, ein regelmäßiges 5-Eck bzw. ein regelmäßiges 6-Eck abgefahren wird?

Wie bestimmst du jeweils den Winkel um den hier gedreht werden muss?

Kannst du das für beliebige regelmäßige Vielecke verallgemeinern?

### 5.4 ROBOTER FOLGT (HAUSTIER):

Der Roboter wird so programmiert, dass er einem Gegenstand folgt (geradlinig). Er versucht den Abstand jeweils zu halten:

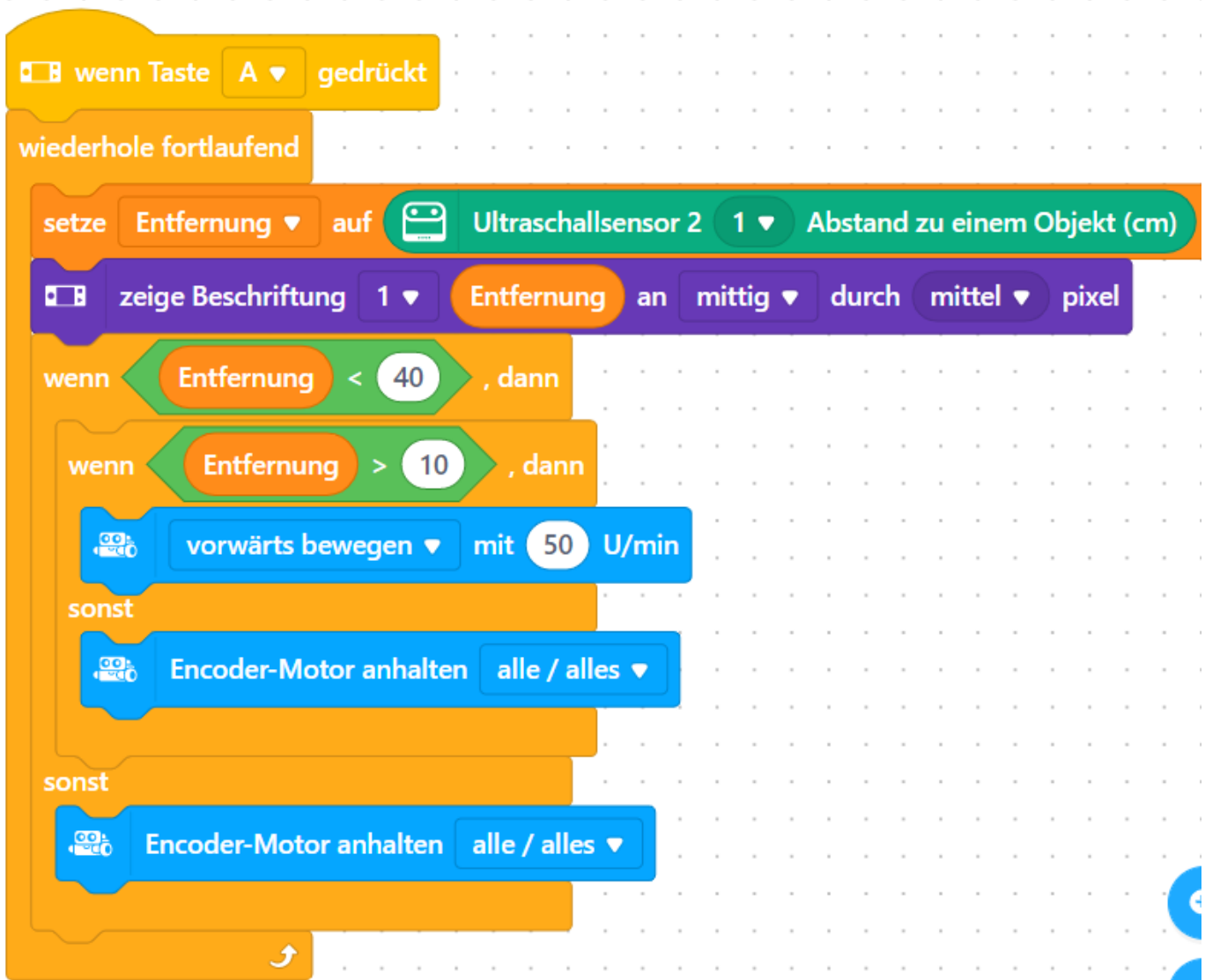


Abbildung 45: Roboter folgt einem Objekt

## 5.5 LINE-FOLLOWER:

Eine wichtige Aufgabe bei Transportrobotern in Betrieben ist es vorgegebenen Wegen zu folgen. Versuche anhand einfacher vorgegebener Wegstrecken, den Roboter so zu programmieren, dass er der schwarzen Linie folgt und, wenn er diese verliert, durch passende Drehungen wieder auf den Weg zurückfindet. Da unser Roboter sogar über 4 Sensoren verfügt, kann man daraus erkennen, in welche Richtung sich der Roboter drehen muss.

- Bei einer Abzweigung nach rechts wird der Sensor R2 die Richtungsänderung erkennen → Drehung nach rechts!
- Bei einer Abzweigung nach links wird der Sensor L2 die Richtungsänderung erkennen → Drehung nach links!



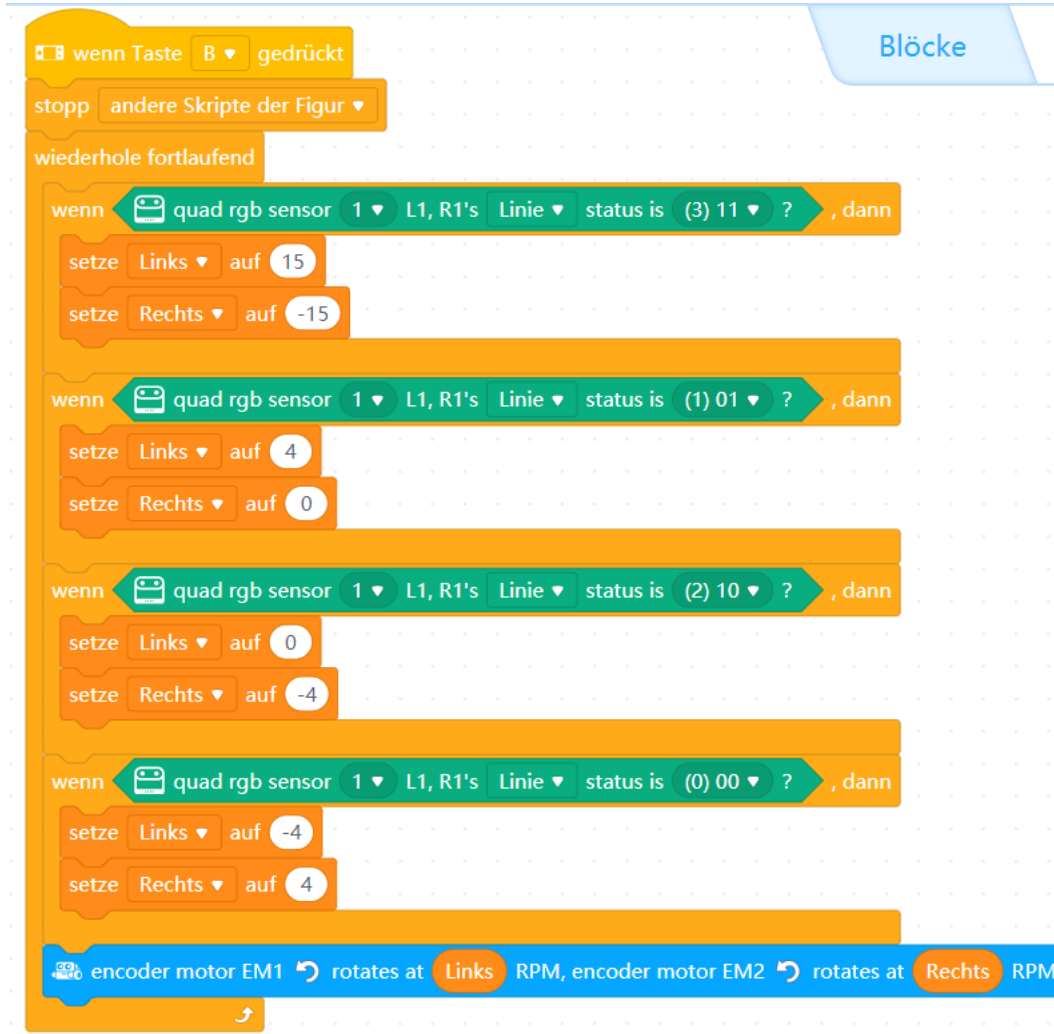


Abbildung 46: LineFollower-Hauptprogramm

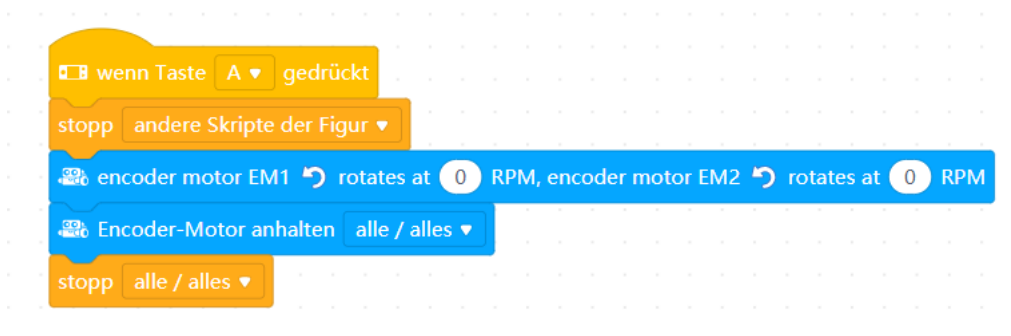
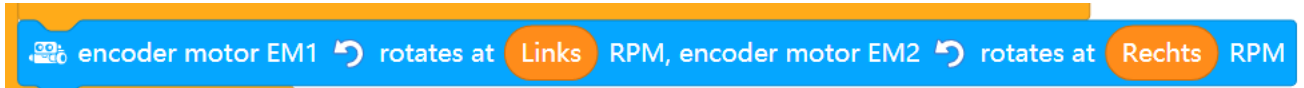


Abbildung 47: LineFollower: Nebenprogramm für Ende


Hinführung zur LineFollower-Lösung:

- Erstelle 2 Variablen: Links und Rechts – sie beschreiben die „Antriebskraft“ für die beiden Motoren.




- Der Roboter soll fahren, solange er „auf der Linie“ ist: (Achte auf die Antriebsrichtung der Motoren)

```

wenn  quad rgb sensor 1 ▾ L1, R1's Linie ▾ status is (3) 11 ▾ ? , dann
  setze Links ▾ auf 15
  setze Rechts ▾ auf -15
  
```

- Bei einer „Rechtskurve“ sieht der rechte Sensor noch die Linie (=1), während die linke Seite nichts mehr sieht (=0). Daher muss der linke Motor sich drehen, während der rechte steht:

```

wenn  quad rgb sensor 1 ▾ L1, R1's Linie ▾ status is (1) 01 ▾ ? , dann
  setze Links ▾ auf 4
  setze Rechts ▾ auf 0
  
```

- Versuche nun dein Programm in Richtung des Line-Followers weiterzuentwickeln!

Beispiellösung:

```















 wenn Taste B ▾ gedrückt
  stopp andere Skripte der Figur ▾
  wiederhole fortlaufend
    wenn  QuadRGB-Sensor 1 ▾ L1, R1's linie ▾ im Status (3) 11 ▾ ? , dann
      setze Links ▾ auf 15
      setze Rechts ▾ auf -15
    wenn  QuadRGB-Sensor 1 ▾ L1, R1's linie ▾ im Status (1) 01 ▾ ? , dann
      setze Links ▾ auf 4
      setze Rechts ▾ auf 0
    wenn  QuadRGB-Sensor 1 ▾ L1, R1's linie ▾ im Status (2) 10 ▾ ? , dann
      setze Links ▾ auf 0
      setze Rechts ▾ auf -4
    wenn  QuadRGB-Sensor 1 ▾ (0) 00 ▾ ? , dann
      setze Links ▾ auf -4
      setze Rechts ▾ auf 4
   encoder motor EM1  rotates at Links RPM, encoder motor EM2  rotates at Rechts RPM
  
 wenn Taste A ▾ gedrückt
  stopp andere Skripte der Figur ▾
   encoder motor EM1  rotates at 0 RPM, encoder motor EM2  rotates at 0 RPM
   Encoder-Motor anhalten alle / alles ▾
  stopp alle / alles ▾
  
```

Abbildung 48: LineFollower: Beispiellösung

## 5.6 CHAMÄLEON:

Hintergrundinformation: Jede Farbe kann durch Mischung der Grundfarben (Rot, Grün, Blau) erzeugt werden. Der RGB-Sensor des mBOT2 kann die einzelnen Farbanteile messen, die LEDs können durch Angabe des Farbanteils gefärbt werden.

Als Vorübung kann die LED mittels RGB-Anteiles in unterschiedlichen Farben gefärbt werden.

Aufgabenstellung: Mit dem RGB-Sensor soll die Untergrundfarbe bestimmt werden und die LEDs dann in dieser Farbe gefärbt werden (vgl: Chamäleon)



Abbildung 49: Chamäleon in der Natur

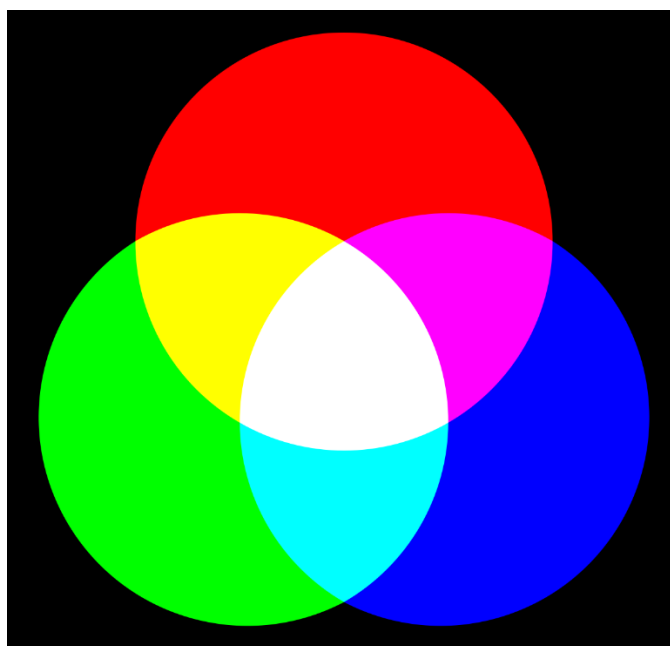


Abbildung 50: Prinzip der additiven Farbmischung

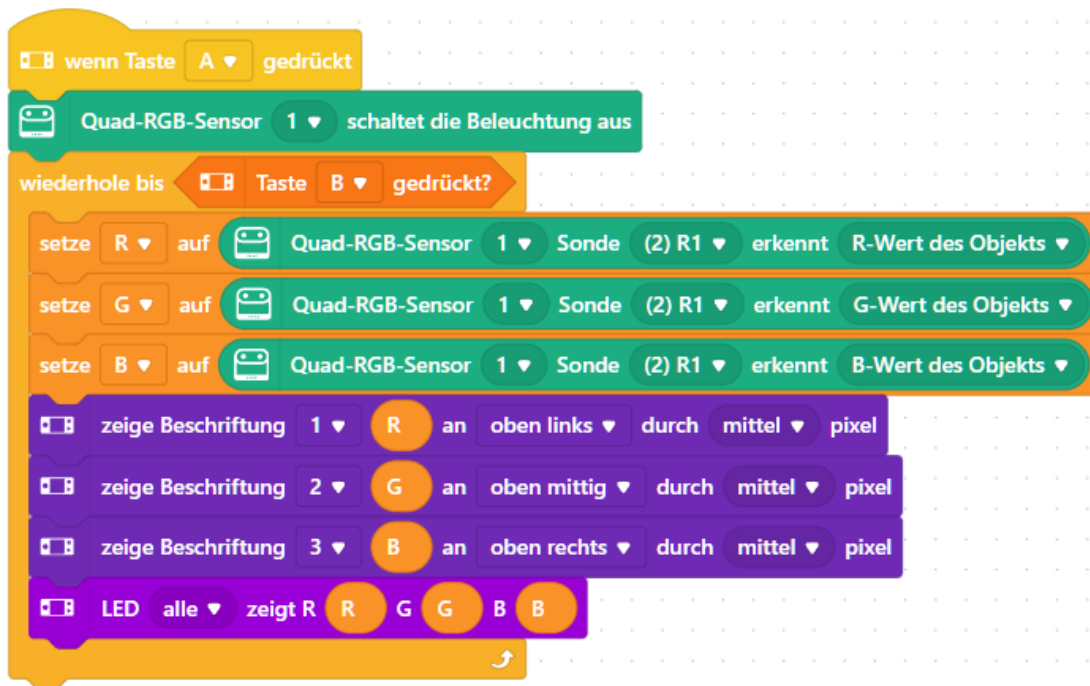


Abbildung 51: Chamäleon - Beispiellösung

## 5.7 DER ROBOTER TANZT:

Über den Akustikteil kann der Roboter Musik erzeugen, über die LEDs Lichteffekte gestalten und über seine Motoren Bewegungen durchführen. Verwende diese Eigenschaften um einen kreativen Tanz zu gestalten. Dazu können Blöcke erzeugt werden, die einzelne Teile dieses Tanzes erzeugen.

Um übergreifend an einem Projekt arbeiten zu können, kann auf den iPADS folgendes Login (in App makeblock) verwendet werden:

Email: [XXXXXX](#)

PW: xxxxxx

Wenn zwei (oder mehr) Aktivitäten gleichzeitig durchgeführt werden sollen, müssen diese Blöcke durch das gleiche Ereignis gestartet werden (siehe Musterlösung):

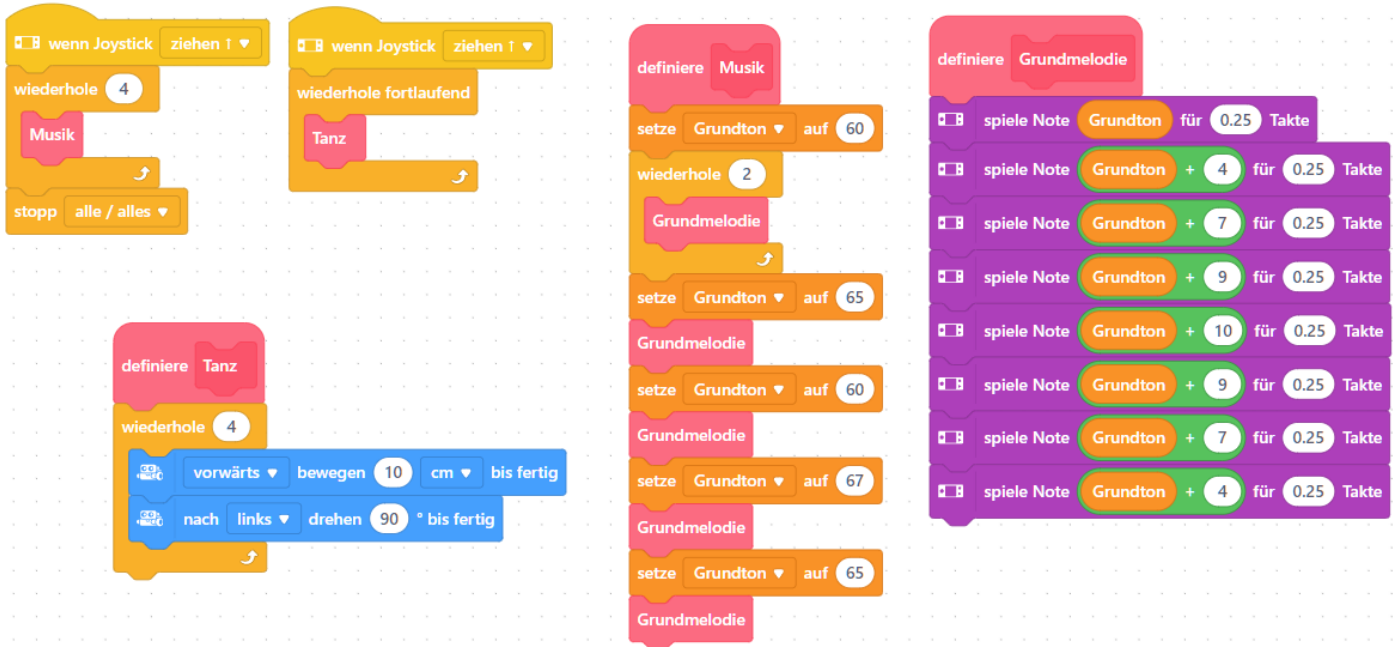


Abbildung 52: Mustertanz

## 6 MUSTERLÖSUNGEN DER KARTEN:

### 6.1 AUFGABE 1



Abbildung 53: Musterlösung Aufgabe 1

### 6.2 AUFGABE 2



Abbildung 54: Musterlösung Aufgabe 2

### 6.3 AUFGABE 3



Abbildung 55: Musterlösung Aufgabe 3

#### 6.4 AUFGABE 4



Abbildung 56: Musterlösung Aufgabe 4

#### 6.5 AUFGABE 5

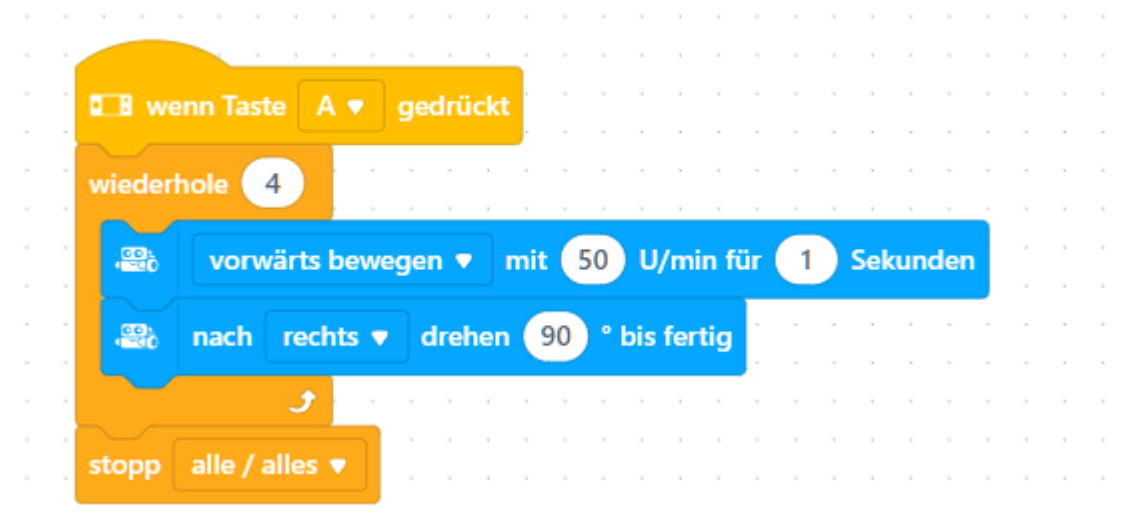


Abbildung 57: Musterlösung Aufgabe 5

## 6.6 AUFGABE 6



Abbildung 58: Musterlösung Aufgabe 6

## 6.7 AUFGABE 7

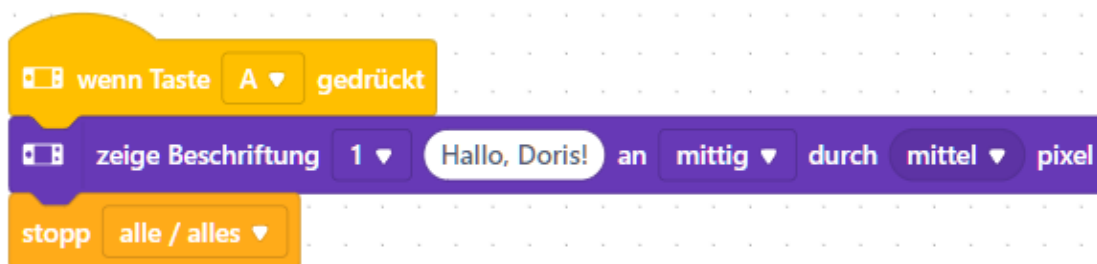


Abbildung 59: Musterlösung Aufgabe 7

## 6.8 AUFGABE 8



Abbildung 60: Musterlösung Aufgabe 8

## 6.9 AUFGABE 9

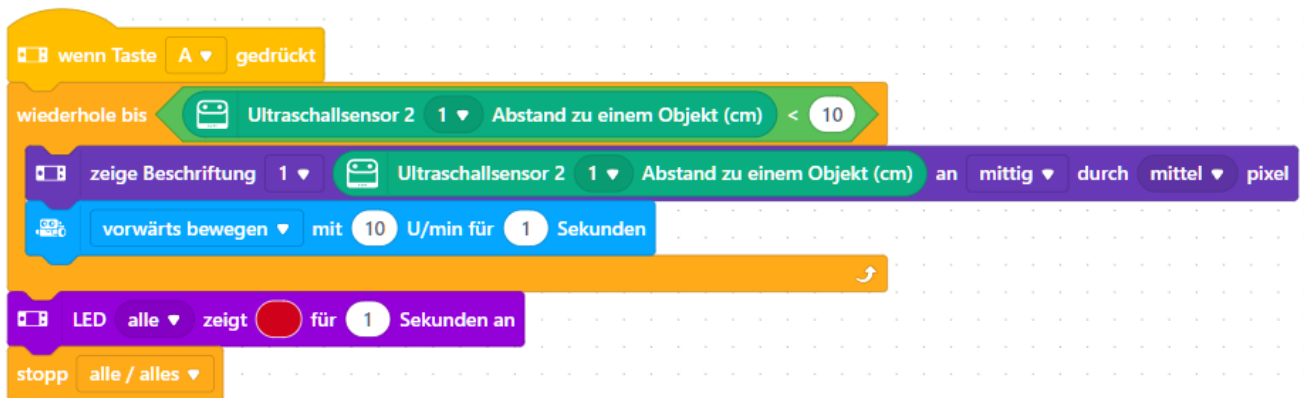


Abbildung 61: Musterlösung Aufgabe 9

## 6.10 AUFGABE 10



Abbildung 62: Musterlösung Aufgabe 10

## 6.11 AUFGABE 11

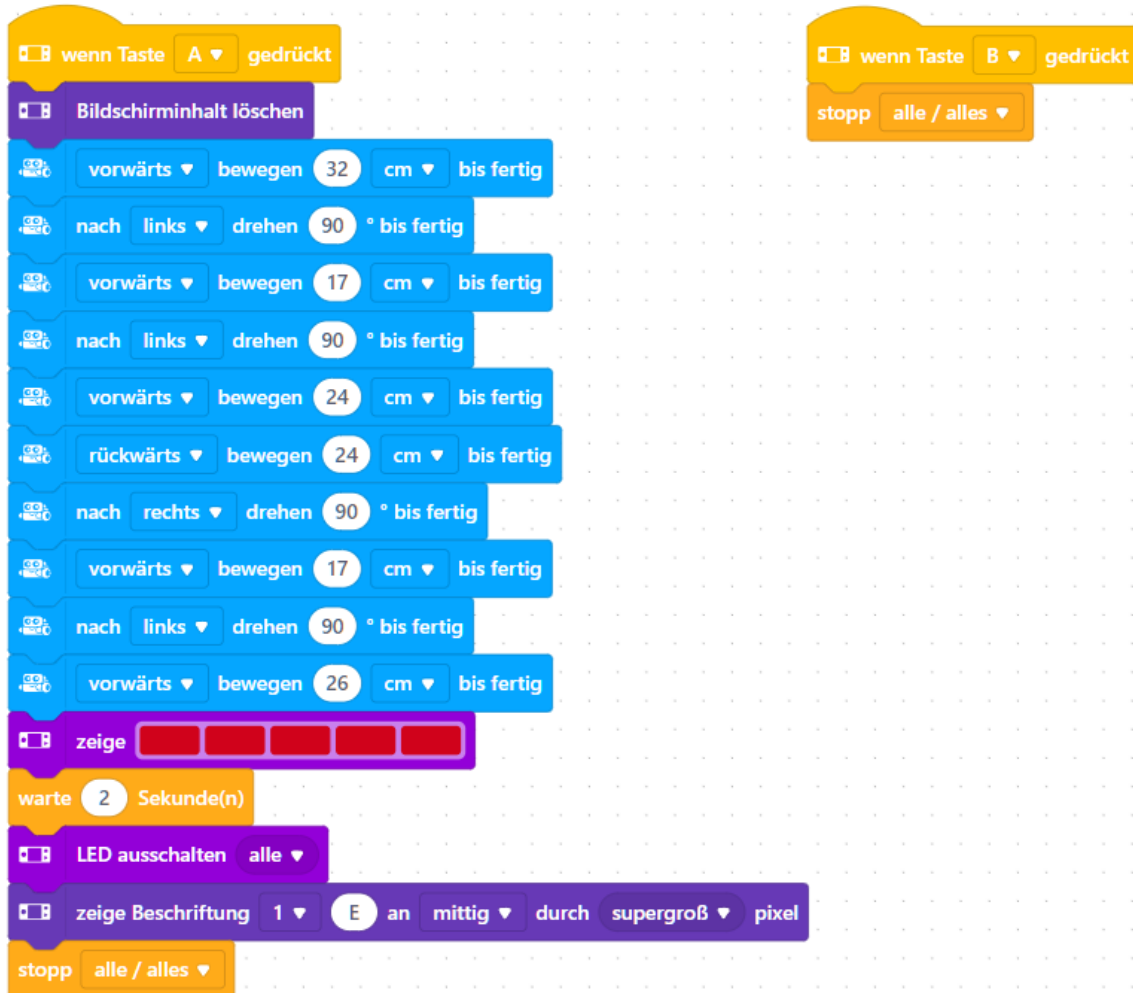


Abbildung 63: Musterlösung Aufgabe 11

## 6.12 AUFGABE 12

(mit Zusatzteil: Seitenlänge kann variabel gewählt werden)

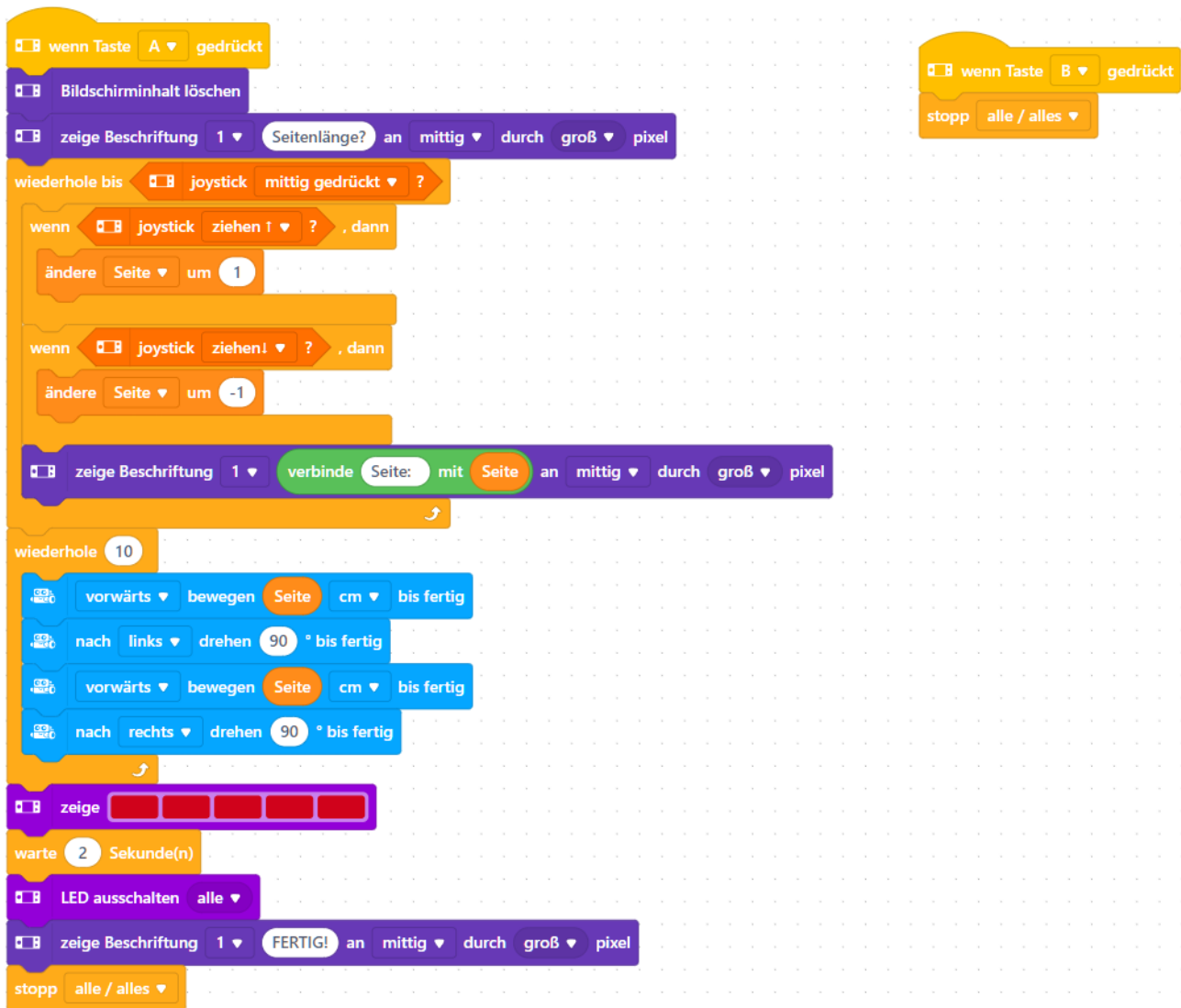


Abbildung 64: Musterlösung Aufgabe 12

## 7 ABBILDUNGSVERZEICHNIS:

Abbildung 1: Programmversionen .....	4
Abbildung 2: mBot2 - Grundstruktur .....	5
Abbildung 3: Ultraschallsensor .....	5
Abbildung 4: RGB Sensoren .....	6
Abbildung 5: Encoder-Motoren .....	6
Abbildung 6: Programmoberfläche .....	7
Abbildung 7: Block- oder Pythonoberfläche? .....	8
Abbildung 8: CyperPi hinzufügen.....	8
Abbildung 9:Erweiterungen hinzufügen .....	9
Abbildung 10: CyperPi über USB verbinden .....	10
Abbildung 11: USB Verbindung funktioniert .....	11
Abbildung 12: Schnittstelle auswählen.....	11
Abbildung 13: Bluetooth am iPad aktivieren .....	11
Abbildung 14: CyperPi als Gerät auswöhlen.....	12
Abbildung 15: Erweiterungen hinzufügen .....	12
Abbildung 16: mit Bluetooth verbinden .....	13
Abbildung 17: Pairing.....	13
Abbildung 18: Bluetooth erfolgreich .....	14
Abbildung 19: Spielemodus (Livemodus) .....	15
Abbildung 20: mBot2 als Gerät.....	15
Abbildung 21: via Bluetooth verbinden .....	16
Abbildung 22: aktive Verbindung .....	16
Abbildung 23: Spielemodus - Fahren.....	17
Abbildung 24: iPad als Fernbedienung .....	17
Abbildung 25: Tasten am CyperPi.....	18
Abbildung 26: Befehle für geradlinige Bewegungen .....	18
Abbildung 27: Beispiellösung Aufgabe 1 .....	19
Abbildung 28: Basisbefehle zur Erzeugung von Drehungen.....	19
Abbildung 29: Lösung Teilaufgabe von 2 .....	20
Abbildung 30: Audio- und Bildschirmausgabe.....	20
Abbildung 31: Beispiel für eine Bewegungsaufgabe .....	21
Abbildung 32: Vorschläge für weitere Aufgaben.....	21
Abbildung 33: CyperPI – Funktionen .....	22
Abbildung 34: Verwendung des Lichtsensors.....	23
Abbildung 35: Verwendung des Schallsensors .....	23
Abbildung 36: Gyroskop.....	23
Abbildung 37: Lösung Zusatzaufgabe Gyroskop .....	24

---

Abbildung 38: Ultraschall in der Tierwelt .....	24
Abbildung 39: Ultraschallsensor (Entfernungsmessung).....	25
Abbildung 40: RGB-Sensoren.....	25
Abbildung 41: Beispiel mit RGB Sensoren .....	25
Abbildung 42: Testmuster für RGB-Aufgabe .....	26
Abbildung 43: Verwendung von Variablen .....	27
Abbildung 44: Lösung der Quadrataufgabe mit variabler Seitenlänge .....	28
Abbildung 45: Roboter folgt einem Objekt.....	29
Abbildung 46: LineFollower-Hauptprogramm.....	30
Abbildung 47: LineFollower: Nebenprogramm für Ende.....	30
Abbildung 48: LineFollower: Beispiellösung .....	31
Abbildung 49: Chamäleon in der Natur .....	32
Abbildung 50: Prinzip der additiven Farbmischung.....	32
Abbildung 51: Chamäleon - Beispiellösung .....	33
Abbildung 52: Mustertanz .....	34
Abbildung 53: Musterlösung Aufgabe 1 .....	35
Abbildung 54: Musterlösung Aufgabe 2 .....	35
Abbildung 55: Musterlösung Aufgabe 3 .....	36
Abbildung 56: Musterlösung Aufgabe 4 .....	36
Abbildung 57: Musterlösung Aufgabe 5 .....	36
Abbildung 58: Musterlösung Aufgabe 6 .....	37
Abbildung 59: Musterlösung Aufgabe 7 .....	37
Abbildung 60: Musterlösung Aufgabe 8 .....	38
Abbildung 61: Musterlösung Aufgabe 9 .....	38
Abbildung 62: Musterlösung Aufgabe 10 .....	39
Abbildung 63: Musterlösung Aufgabe 11 .....	40
Abbildung 64: Musterlösung Aufgabe 12 .....	41