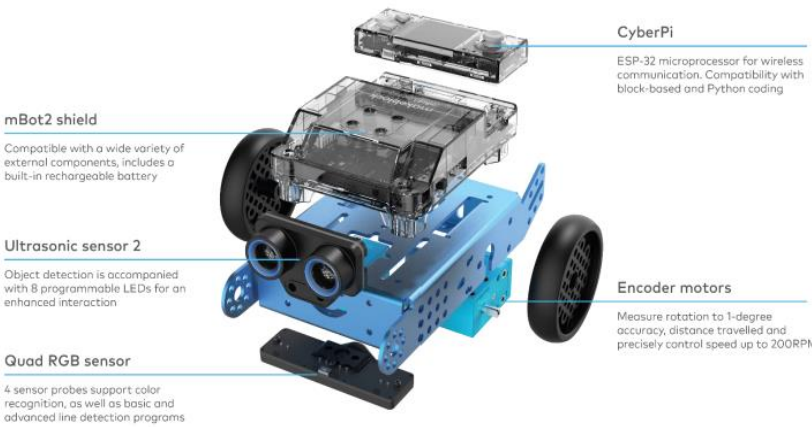
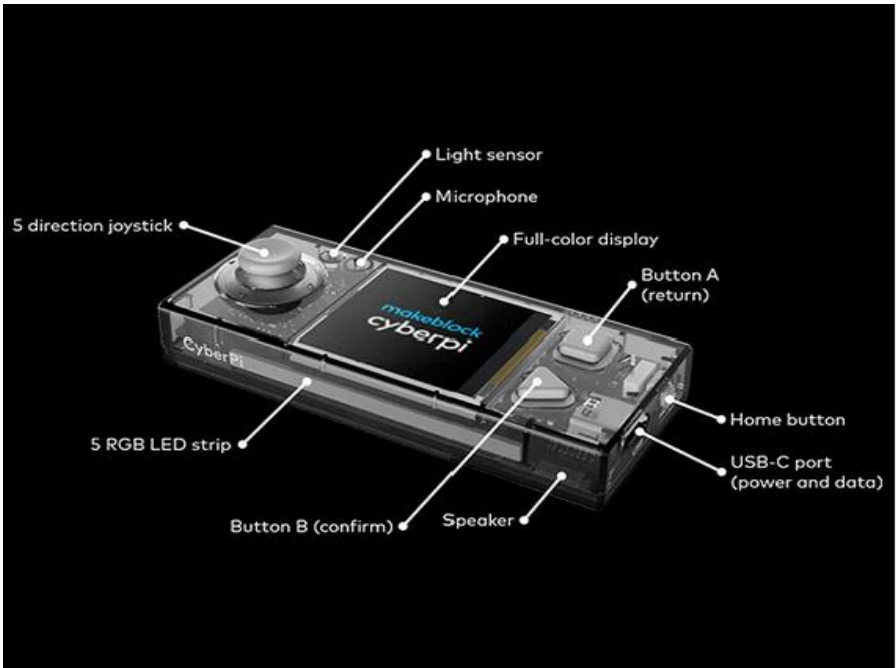
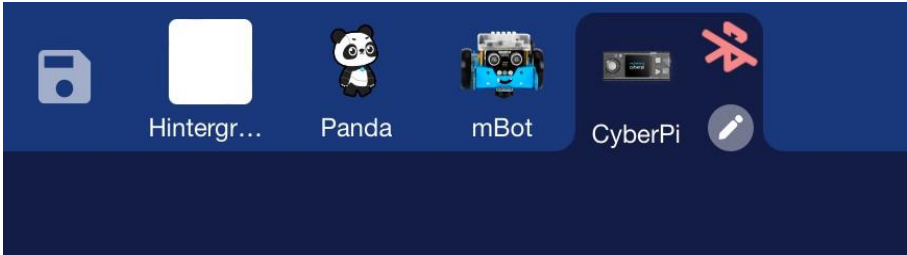


<p>Tag 01</p>	<p>Anlage - Kreative Robotik (mBot2)</p>
<p>Hier seht ihr die Bestandteile des mBot2:</p> <p>mBot 2 Shield: Schutzgehäuse</p> <p>Ultrasonic Sensor 2: Ultraschallsensor</p> <p>Quad RGB Sensor: Vierfacher RGB Sensor</p> <p>Encoder Motors: Encoder Motoren</p> <p>Cyber Pi: Die Fernbedienung des mBot2</p>	 <p>CyberPi ESP-32 microprocessor for wireless communication, Compatibility with block-based and Python coding</p> <p>mBot2 shield Compatible with a wide variety of external components, includes a built-in rechargeable battery</p> <p>Ultrasonic sensor 2 Object detection is accompanied with 8 programmable LEDs for an enhanced interaction</p> <p>Quad RGB sensor 4 sensor probes support color recognition, as well as basic and advanced line detection programs</p> <p>Encoder motors Measure rotation to 1-degree accuracy, distance travelled and precisely control speed up to 200RPM</p> <p>Bildquelle: https://education.makeblock.com/resources/res-mbot/85070/</p>
	<p>Eine Video zum Aufbau des mBot2 findet ihr in der beigelegten Anleitung und als visuelle Hilfestellung hier "Mit dem mBot2 kinderleicht programmieren", https://www.youtube.com/watch?v=l6_QLcm_gdU (Stand: 16.02.2024).</p>



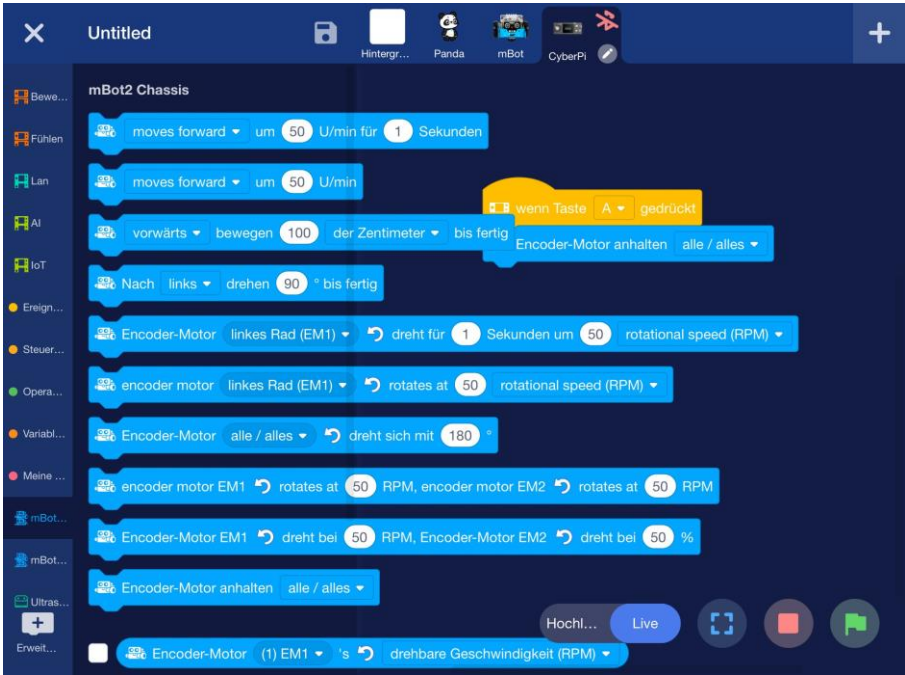
<p>Tag 02</p> <p>02.1) Was ist der Cyber Pi?</p> <p>Am mBot2 ist ein CyberPi montiert. Ein CyberPi ist ein programmierbarer Mikrocomputer. Ihr gebt dem CyberPi Befehle mithilfe der Codeblöcke in mBlock. Der CyberPi gibt diese Befehle an den mBot2 weiter. Ihr könnt den CyberPi vom mBot2 per Hand entfernen. Achtung: Der CyberPi muss auf den mBot2 bleiben, um ihn mit den iPads zu programmieren.</p>	 <p>Bildquelle: https://education.makeblock.com/resources/res-mbot/85070/</p>
<p>02.2)</p> <p>Ihr müsst den Cyber Pi mit dem Pluszeichen oben rechts in mBlock hinzufügen.</p>	



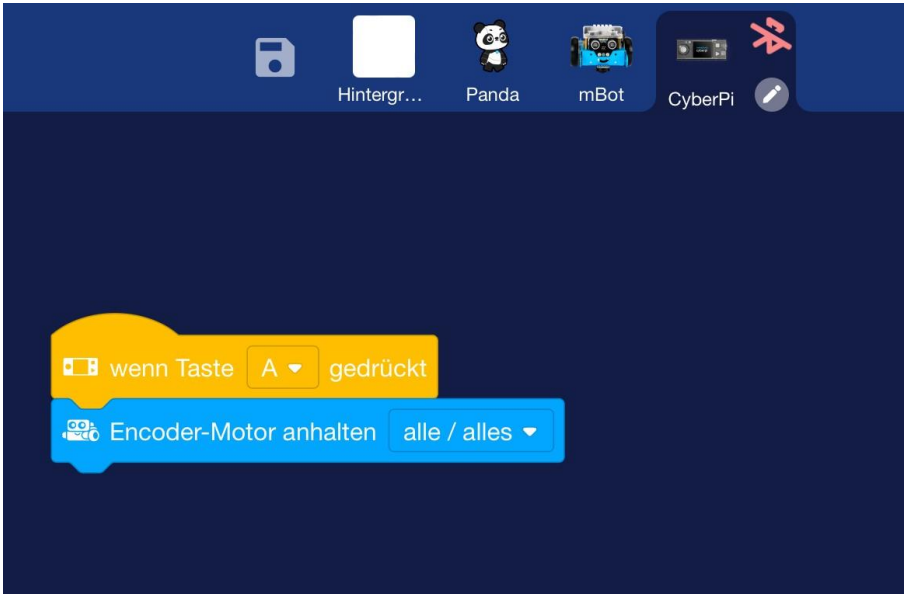
<p>02.3) Anschließend müsst ihr auf das Pluszeichen unten links in mBlock klicken, um ins Erweiterungszentrum zu gelangen. Dort müsst ihr einige Erweiterungen hinzufügen, um mit dem Programmieren beginnen zu können:</p> <ul style="list-style-type: none"> - mBot2 shield - Ultraschallsensor 2 - Lichtsensor - Dualer RGB Sensor 	
<p>02.4) Den Cyber Pi verbindet ihr mit dem iPad via USB-Adapter.</p> <p>Um ein Programm zu testen, stellt sicher, dass der Hochlade-Modus aktiviert ist und sendet das Programm mit der "grünen Flagge" an den Cyber Pi.</p>	
<p>02.5) Die Bewegungen des mBot2 mit dem CyberPi kontrollieren.</p>	<p>Aufbau des mBot2:</p> <p>Der mBot2 wird mit einem speziellen Paar Motoren geliefert, die die Drehung der Achse und damit die Geschwindigkeit und Entfernung, die der Roboter zurückgelegt hat, aufzeichnen können. Sie werden Encoder-Motoren genannt aufgrund des eingebauten Sensors (Encoders). Diese Art von Motoren ermöglicht es, bestimmte Parameter wie den Drehwinkel und die Geschwindigkeit der Motoren zu kontrollieren.</p>



02.6)
 Wenn ihr damit beginnt, den mBot2 zu programmieren, werdet ihr feststellen, dass es viele verschiedene Codeblöcke gibt, die ihr verwenden könnt, um den Roboter zu bewegen. Diese Blöcke findet ihr in mBlock in der Kategorie "Chassis". Diese Codeblöcke sind blau.

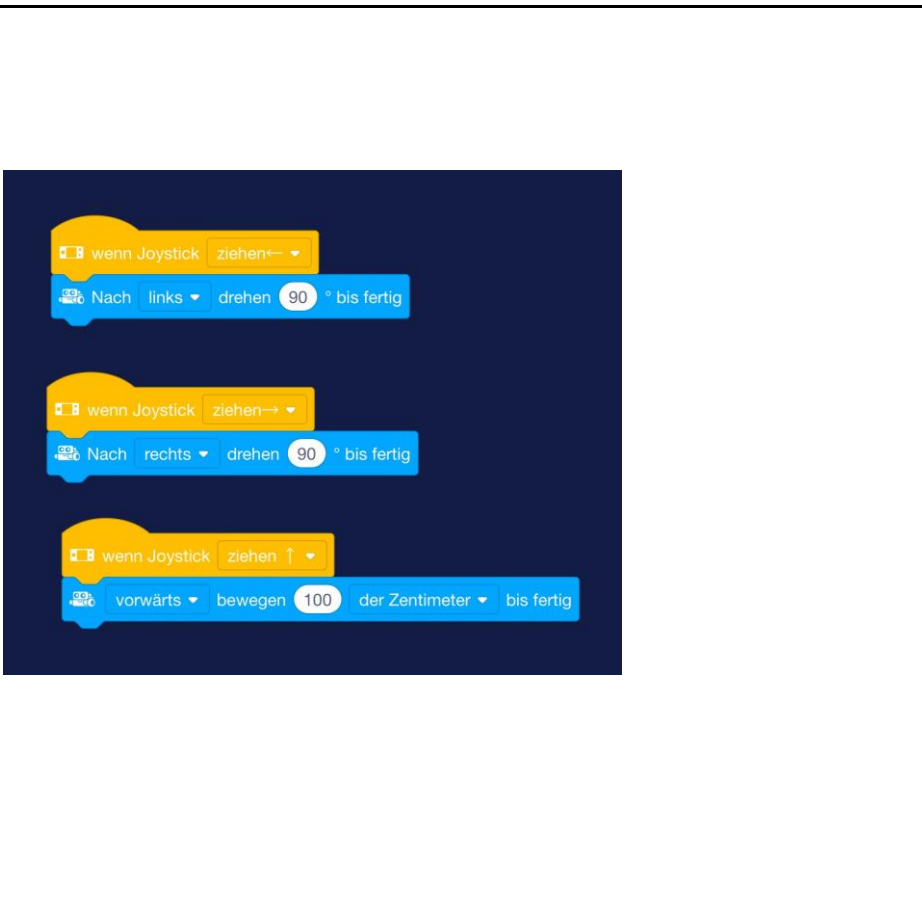


02.7)
 Dieser Codeblock ermöglicht es euch, den Roboter anzuhalten. Dies kann während einer Testphase nützlich sein. Wenn ein Programm nicht wie erwartet funktioniert, könnt ihr diesen Block verwenden, um den Roboter sofort anzuhalten.

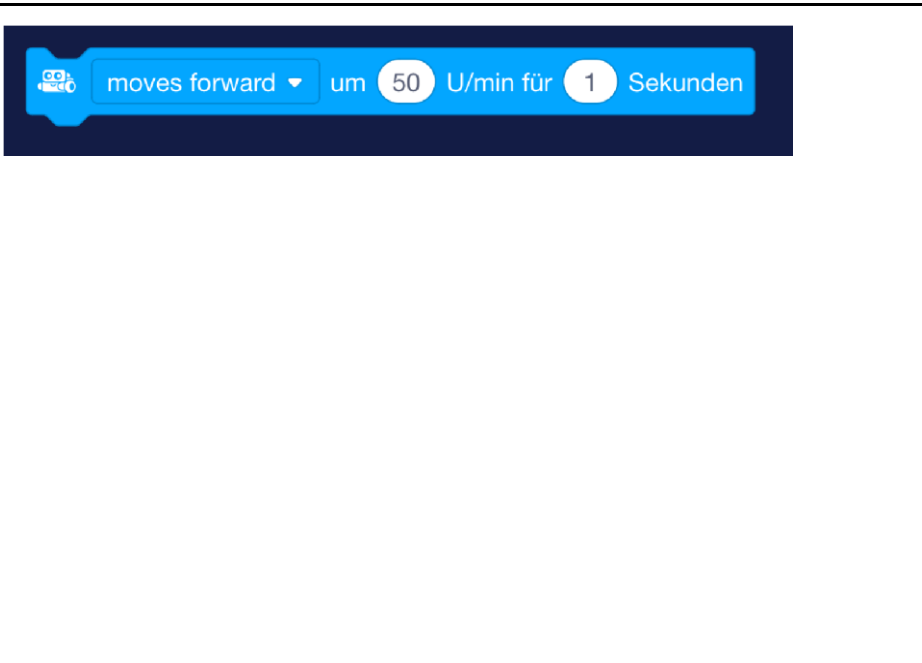


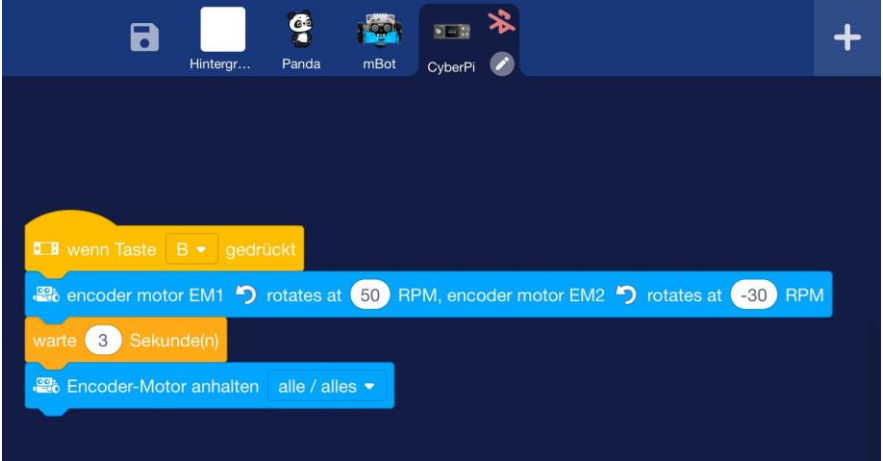
02.8 Mit diesen Codeblöcken könnt ihr den mBot2 dazu bewegen sich um eine bestimmte Anzahl von Grad zu drehen und auszuwählen, ob die Drehung nach links oder rechts erfolgen soll.

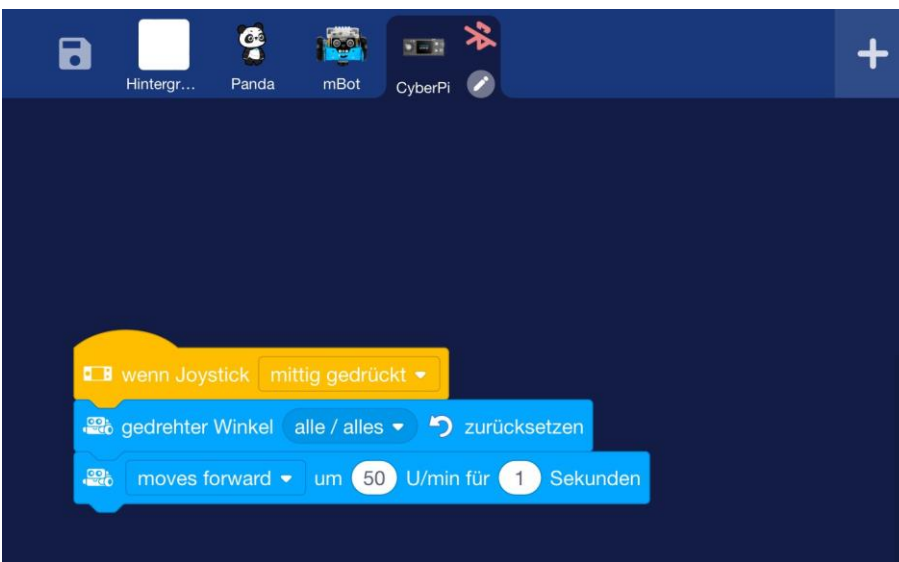
Wenn das folgende Programmbeispiel eingestellt ist, kann der mBot2 über den Joystick gesteuert werden.



02.9) Dieser Codeblock ermöglicht es euch, den mBot2 vorwärts, rückwärts, nach links und rechts mit einer spezifischen Rotationsgeschwindigkeit der Räder und für eine bestimmte Anzahl von Sekunden zu bewegen. Das folgende Beispiel zeigt, wie du den mBot2 eine Sekunde lang mit einer Geschwindigkeit von 50 Umdrehungen pro Minute bewegen kannst.



<p>02.10) Ihr könnt die Motoren des mBot2 auch unabhängig voneinander steuern. In diesem Beispiel wird der mBot2 für 3 Sekunden eine Kurve fahren, bevor er anhält. Beachtet, dass der Wert für einen der Motoren negativ ist. Dies liegt daran, dass die Motoren in entgegengesetzte Richtungen montiert sind. Um den Roboter in eine Richtung zu bewegen, müsst ihr einen der Motoren in die entgegengesetzte Richtung drehen lassen.</p>	
---	--

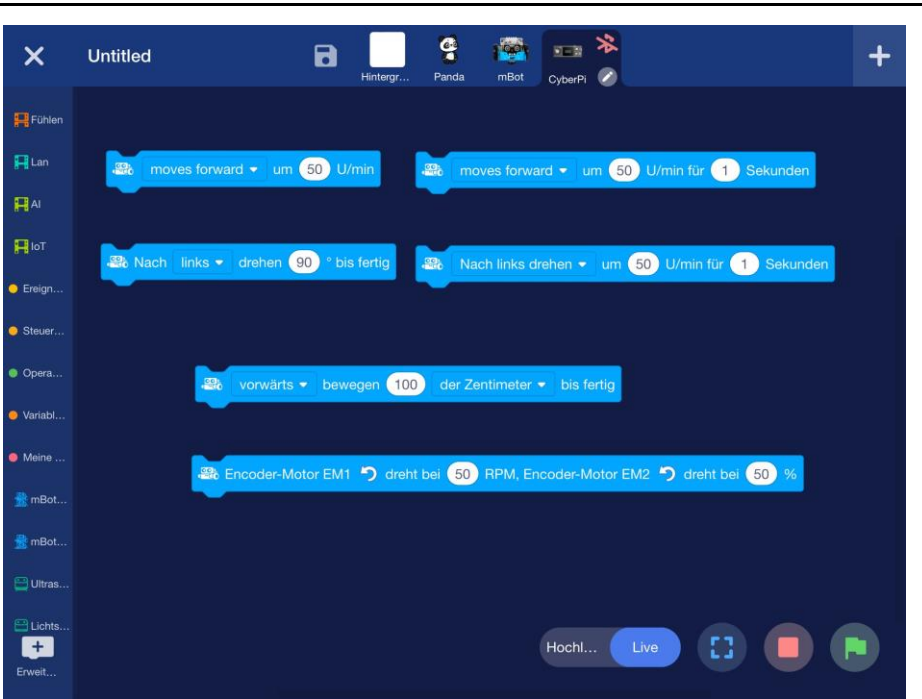
<p>02.11)</p> <p>Da die Motoren des mBot2 ihre Geschwindigkeit und Rotation messen können, können Sie auch diese Werte lesen. Die gemessene Geschwindigkeit ist die Rotationsgeschwindigkeit in Umdrehungen pro Minute ($360^\circ = 1$ Umdrehung; in der Wissenschaft ist es üblicher, "Grad pro Sekunde" zu verwenden ($1 \text{ U/min} = 6^\circ \text{ pro Sekunde}$). Das Programmbeispiel ändert die Werte der Rotation zurück auf null und befiehlt dann dem mBot2, eine Sekunde lang vorwärts zu fahren.</p>	
--	---



02.12) Hier könnt ihr einige Blöcke nebeneinander sehen. Probiert jedes Paar nacheinander aus.

Was passiert? Was sind die Unterschiede zwischen den Blöcken?

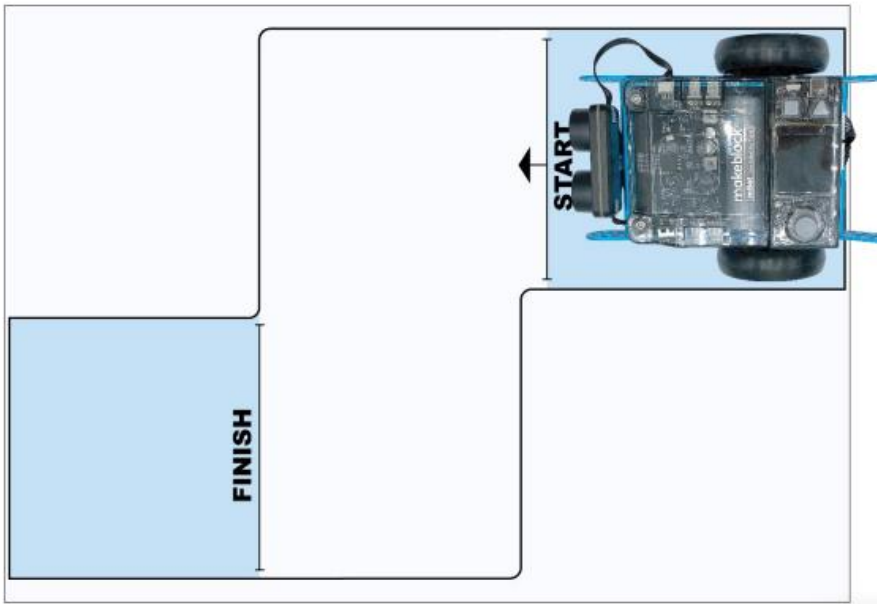
Vergesst nicht, den gelben Block für den Joystick einzufügen.

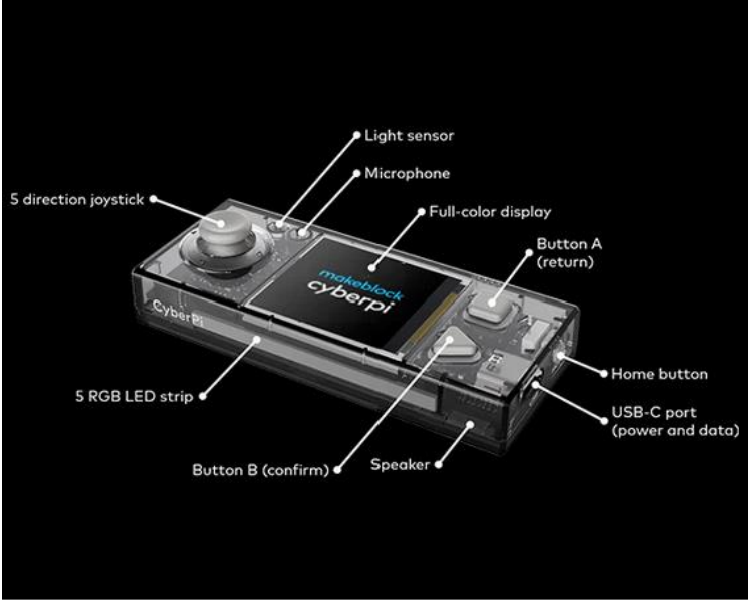



02.13) Hier seht ihr ein Beispiel für ein simples Labyrinth. Versucht den mBot2 so zu programmieren, dass er diesen Weg abfährt.

Oder ihr nutzt die Fahrbahn, die als A2 Poster der Box beigelegt ist.

Quelle:
<https://education.makeblock.com/resources/res-mbot/85070/>



<p>Tag 03</p>	
<p>03.1)</p> <p>Das Mikrofon befindet sich oben links, der Lautsprecher befindet sich unten rechts auf dem CyberPi.</p>	 <p>Bildquelle: https://education.makeblock.com/resources/res-mbot/85070/</p>
<p>03.2) In mBlock gibt es mehrere Codeblöcke, die ihr verwendet könnt, um den Lautsprecher und das Mikrofon zu programmieren. Ihr findet diese Codeblöcke in der Audio-Kategorie des Blockbereichs. Diese Codeblöcke sind lila.</p>	



03.3) In dem folgenden Programmierbeispiel startet beim Drücken der Taste A die Aufnahme für drei Sekunden, dann stoppt die Aufnahme. Wenn ihr die Taste B drückt, hört ihr, was aufgenommen wurde.

Durch die Verwendung dieser beiden Blöcke können zwei Programmieraufgaben unabhängig voneinander ausgeführt werden, ohne dass eine Hauptschleife benötigt wird.



<p>03.4) Im Programmbeispiel rechts beginnt der CyberPi mit der Aufnahme, wenn ihr die Taste A drückt. Wenn ihr die Taste B drückt, wird die Aufnahme gestoppt. Wenn ihr den Joystick nach unten bewegt, hört ihr, was der mBot2 aufgenommen hat. Nachdem die gesamte Aufnahme beendet ist, hört ihr einen Hinweiston.</p> <p>Die Codeblöcke in diesem Beispiel sind in einer Schleifenstruktur platziert.</p> <p>Dadurch wird sichergestellt, dass die Eingaben von unten und vom Joystick kontinuierlich überwacht werden.</p>	
<p>03.5)</p>	<p>Ihr könnt diese Programmierbeispiele in mBlock nachbilden und testen. Überlegt euch eine Erweiterung zu einem Programmierbeispiel. Testet die Programmbeispiele und Ihre Erweiterungen im Live-Modus.</p>



<p>Tag 04</p>	
<p>04.1) Autonome Roboter</p>	<p>Autonome Roboter müssen sich ihrer Umgebung bewusst sein, wenn sie unbeaufsichtigt fahren, um Kollisionen mit Hindernissen oder Personen zu vermeiden. Dies gilt auch für den mBot2. Er verfügt über einen speziellen Sensor dafür an der Vorderseite. Er wird Ultraschallsensor genannt und ermöglicht es dem Roboter, Objekte in seinem Weg mithilfe von Geräuschen zu erkennen, die Menschen nicht hören können Ultraschall.</p>
<p>04.2) Der Ultraschallsensor</p>	<p>Wie funktioniert ein Ultraschallsensor?</p> <p>Die Roboter haben einen Ultraschallsender und einen Empfänger an der Vorderseite der Roboter. Die kleinen Zylinder können als Augen betrachtet werden. Dieser Sensor sendet kurze Impulse von Ultraschalltönen aus und lauscht auf ein Echo. Kommt der Roboter einem Hindernis nahe, wird der Ultraschallton zum Roboter zurückgeworfen. Anhand der Zeit, die der Ton benötigt, um zum Sensor zurückzukehren, berechnen die Roboter intern die Entfernung zu ihm. Diese Daten können verwendet werden, um zu entscheiden, wie der Roboter reagieren soll. Die Programmierung einer solchen Aktion könnte zum Beispiel sein, dass der Roboter aufhören soll zu fahren oder eine Kurve machen soll.</p>
<p>04.3) Erweiterung hinzufügen</p>	<p>In mBlock gibt es mehrere Codeblöcke für den Ultraschallsensor, die ihr in euren Programmen verwenden könnt.</p> <p>Dazu müsst ihr die Erweiterung Ultraschallsensor aus der Erweiterungsbibliothek hinzufügen. Anschließend findet ihr die Codeblöcke in der Kategorie "Ultraschallsensor" im Blockbereich von mBlock. Diese Codeblöcke sind grün.</p>



04.4)

Mit diesem Code misst ihr die Entfernung zwischen dem Sensor und dem Hindernis.

Der Erfassungsbereich liegt zwischen 3 und 300 cm. Ihr könnt den Wert der Entfernung verwenden, um den Roboter bestimmte Aktionen ausführen zu lassen. Zum Beispiel könnt ihr verhindern, dass der Roboter mit Hindernissen kollidiert.

Im folgenden Programmierbeispiel fährt der Roboter vorwärts. Wenn der Roboter weniger als 10 cm von einem Hindernis entfernt ist, macht der Roboter eine 90°-Drehung nach links und fährt dann wieder vorwärts.

```

wenn Taste A gedrückt
  moves forward um 50 U/min
  wiederhole fortlaufend
    wenn ultrasonic 2 1 distance to an object (cm) < 10, dann
      Nach links drehen 90 ° bis fertig
    sonst
      moves forward um 50 U/min
  
```



04.5) Übung: Lasst den Mbot2 eine zufällige Runde im Zimmer fahren. Zufällig bedeutet, dass ihr die Route nicht vorher festlegt. Wenn der Roboter einem Hindernis wie einem Stuhl, Tisch oder Rucksack nahe kommt, hört ihr einen Ton, und der mBot2 wird sich umdrehen. Ihr werdet den Ton selbst mit dem Mikrofon aufnehmen. Der Ton könnte beispielsweise ein Piepton oder das Geräusch eines bremsenden Autos sein. Ihr könnt auch einen Text aufzeichnen oder den Roboter ein Lied spielen lassen.

Links seht ihr das Programm: Der Roboter gibt einen Ton von sich, wenn er sich einem Hindernis nähert, und dreht sich dann um. Ihr habt den Ton zuvor selbst aufgenommen.

Beide Programm müsst ihr zusammen in mBlock erstellen.

Setzt das Beispiel um und versucht es zu erweitern und anzupassen. Seid kreativ.

```

wenn Taste A gedrückt
  Aufnahme starten
  warte 3 Sekunde(n)
  Aufnahme beenden
    
```

```

wenn Taste B gedrückt
  wiederhole fortlaufend
    wenn ultrasonic 2 1 distance to an object (cm) < 50 , dann
      spiele Aufnahme bis fertig
      Encoder-Motor alle / alles dreht sich mit 90 °
    sonst
      moves forward um 50 U/min
  
```



<p>Tag 05</p>	
<p>05.1) Was sind Sensoren?</p>	<p>Jeder Roboter arbeitet mit Sensoren. Sensoren können mit euren Sinnen verglichen werden (Geschmack, Berührung, Geruch, Gehör, Sehen). Durch diese Sensoren sieht der mBot2 seine Umgebung. Es gibt verschiedene Arten von Sensoren, die den mBot2 sehen lassen können, wie z.B. Lichtsensor, Geräuschsensor, Gyroskop-Beschleunigungssensor, Quad-RGB-Sensor, Timer.</p>
<p>05.2) Erweiterungen hinzufügen</p>	<p>In mBlock gibt es mehrere Codeblöcke für die Sensoren, die ihr in euren Programmen verwenden könnt.</p> <p>Dazu müsst ihr die Erweiterungen Ultraschallsensor und Quad-RGB-Sensor aus der Erweiterungsbibliothek hinzufügen. Anschließend findet ihr die Codeblöcke in der Seitenleiste.</p>



05.3)

Der Lichtsensor misst die Stärke des Lichts, der Lichtenergie.

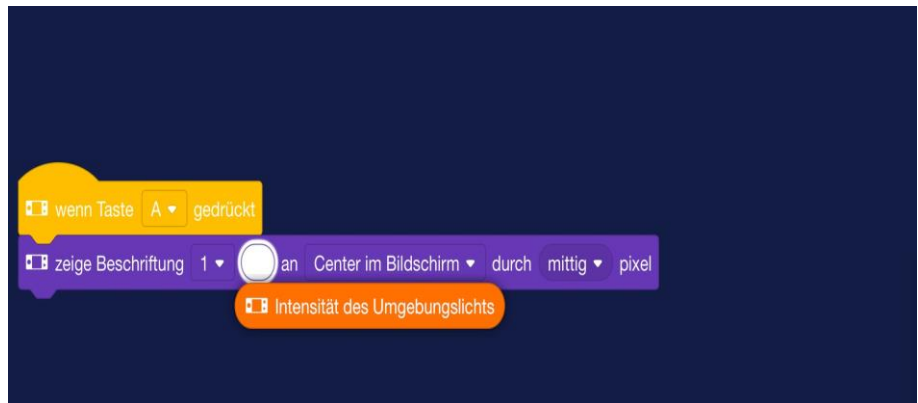
Der Lichtsensor ist ein Gerät, das Lichtenergie in elektrische Energie umwandelt.

Ein Lichtsensor wird häufig verwendet, um zum Beispiel die Beleuchtung in Häusern zu steuern oder die Scheinwerfer von Fahrzeugen automatisch einzuschalten.

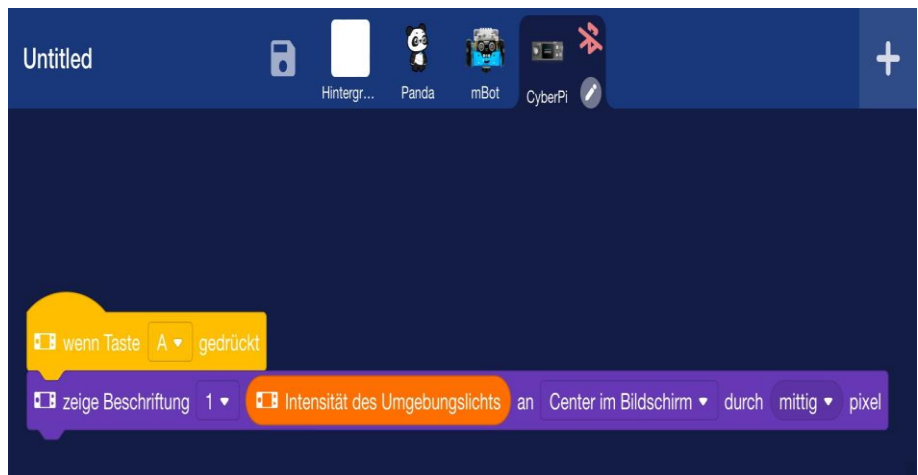
Im folgenden Programmierbeispiel wird die Lichtenergie der Umgebung auf dem Display des CyberPi angezeigt.

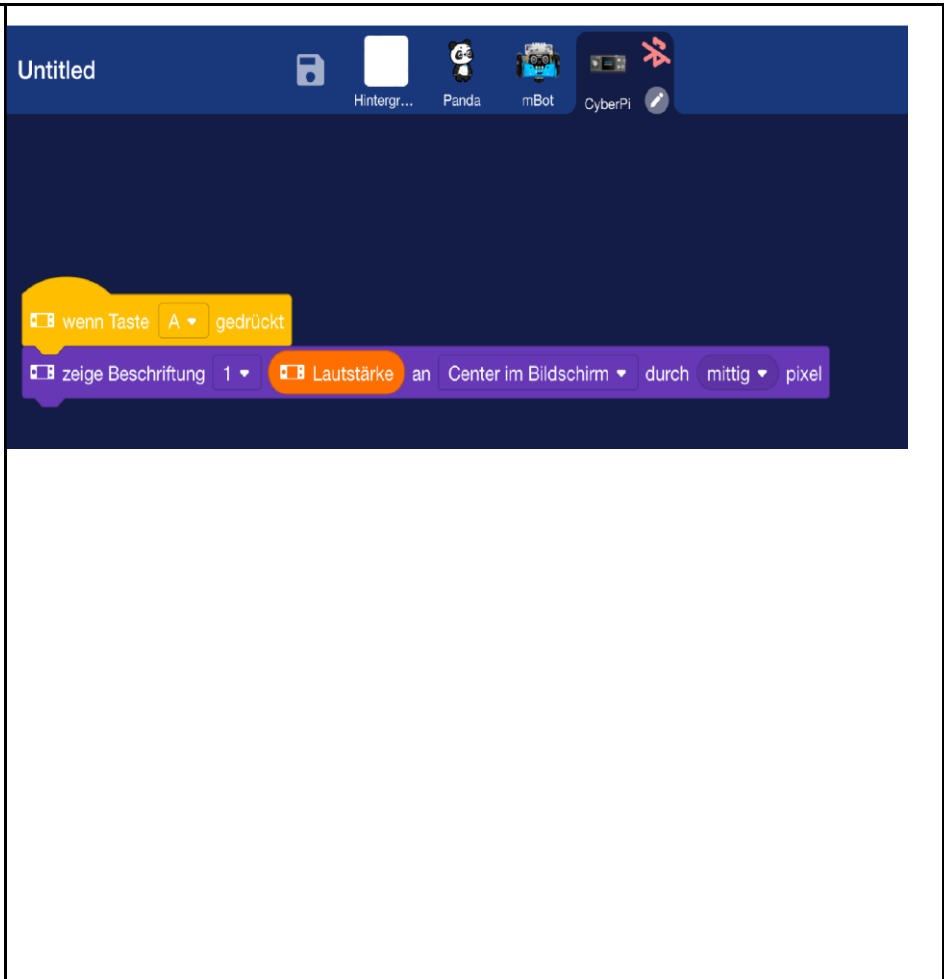
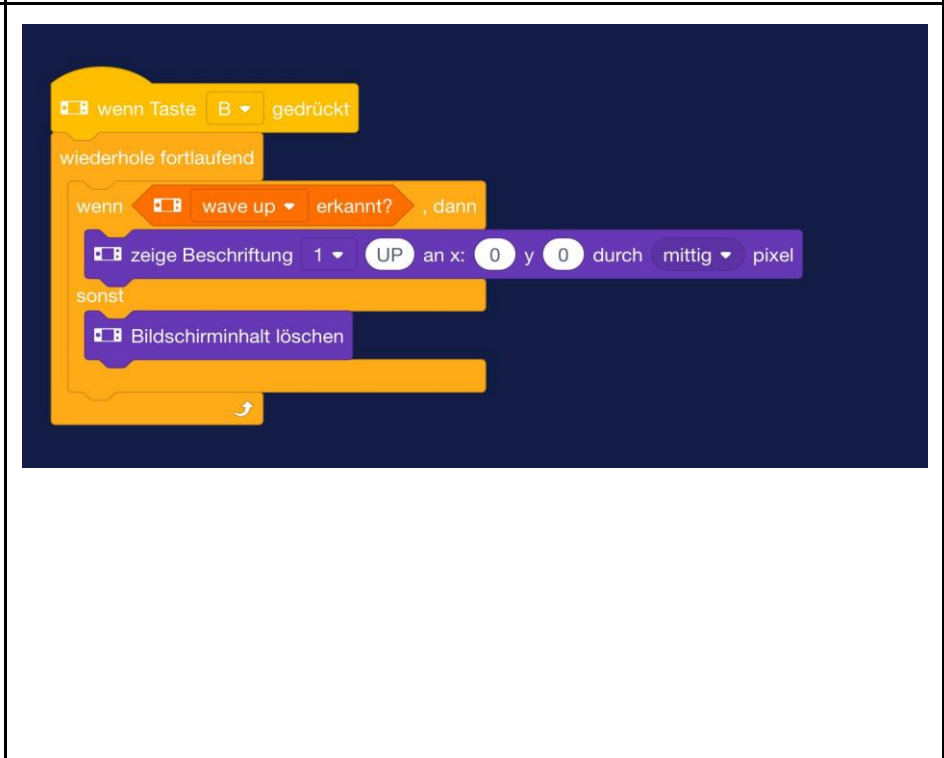
Ihr könnt das Programmierbeispiel erweitern, damit der mBot2 beispielsweise anhält, wenn die Lichtenergie der Umgebung unter einem bestimmten Wert liegt, also wie wenn es dunkel wird.

Schritt 01) Zieht den orangenen Block in das weiße Feld



Schritt 02) So sieht der fertige Programmblock aus



<p>05.4) Der Sound-Sensor im Mikrofon wandelt den Ton in ein elektrisches Signal um, das hinsichtlich der Tonhöhe und Lautstärke ausgewertet werden kann. Sound-Sensoren werden oft für Telefonanrufe und Sprachsteuerungssystemen verwendet.</p> <p>Im folgenden Programmierbeispiel wird die Schallintensität der Umgebung auf dem Display des CyberPi angezeigt. Ihr könnt das Programmierbeispiel erweitern, damit der mBot2 während einer Tour durch euer Zimmer kontinuierlich die Schallintensität misst und auf dem Display anzeigt.</p>	 <p>The image shows a Scratch script for a CyberPi. The script starts with a 'when key pressed' block for key 'A'. This is followed by a 'show text' block with the text 'Lautstärke an' (Volume on), centered on the screen, with a font size of 1 and a font color of purple. The text is positioned in the middle of the screen.</p>
<p>05.5) Das Gyroskop misst Kippbewegungen, genauer gesagt die Geschwindigkeit von Dreh- und Kippbewegungen.</p> <p>Im folgenden Programmierbeispiel wird die Kippbewegung des mBot2 auf dem Display angezeigt.</p> <p>Ihr könnt das Programmierbeispiel erweitern, damit der mBot2 während einer Runde in eurem Zimmer die Kippbewegung kontinuierlich misst und auf dem Display anzeigt</p>	 <p>The image shows a Scratch script for a mBot2. The script starts with a 'when key pressed' block for key 'B'. This is followed by a 'repeat' block set to 'forever'. Inside the repeat loop, there is a 'when wave up detected' block. If detected, a 'show text' block displays 'UP' at the top center of the screen (x: 0, y: 0) in purple font, size 1. If not detected, a 'clear screen' block is executed.</p>

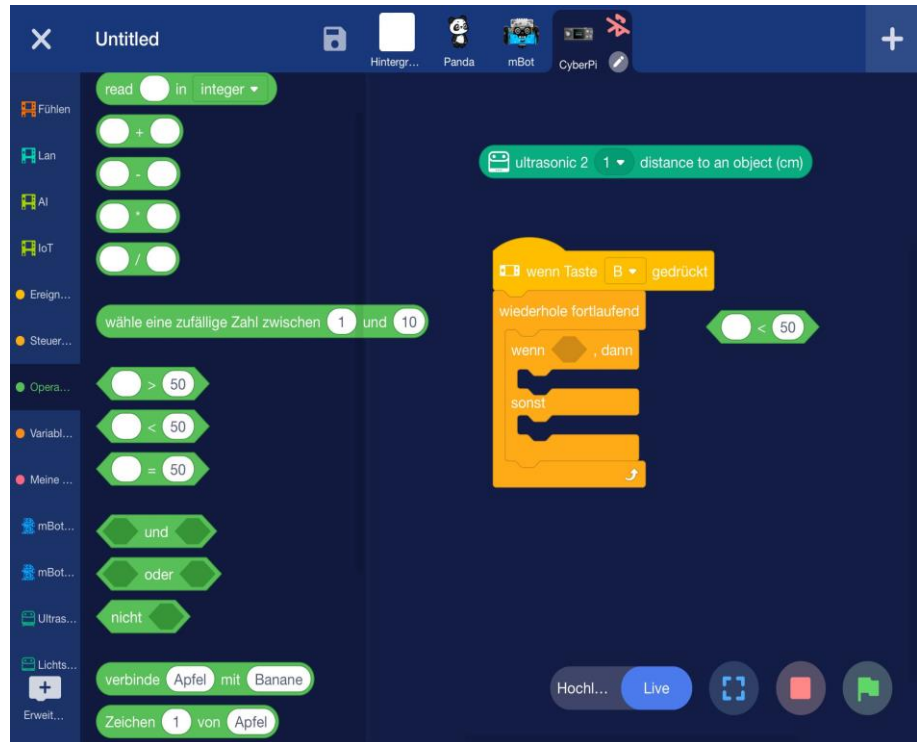


05.6) Der Ultraschallsensor sendet Schall aus und misst die Zeit, bis der Schall von einem Objekt zurückgeworfen wird, um Entfernungen zu Objekten zu bestimmen.

Ein Ultraschallsensor wird oft bei der Steuerung von Robotern verwendet, um Kollisionen zu verhindern.

Im folgenden Programmierbeispiel wird der Ultraschallsensor verwendet, um zu verhindern, dass der mBot2 in ein Hindernis fährt. Wenn der mBot2 weniger als 10 cm von einem Hindernis entfernt ist, macht der Roboter eine 90°-Drehung nach links und fährt dann einfach weiter.

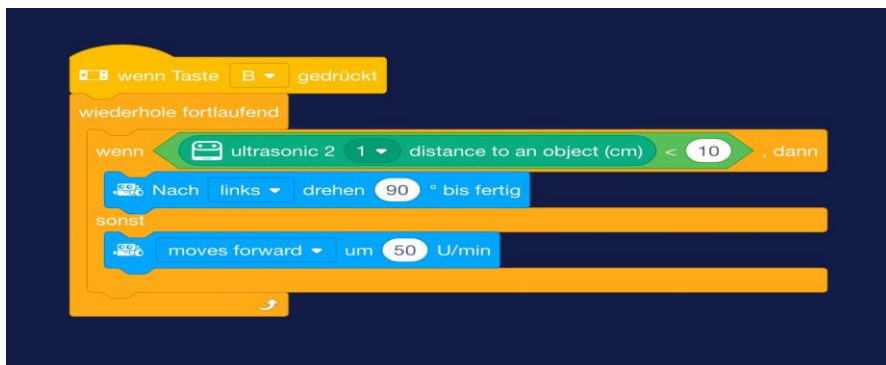
Schritt 01) Hier seht ihr wie ihr die Programmierblöcke zusammenbaut



Schritt 02) So fügt ihr die Blöcke zusammen.



Schritte 03) Der fertige Programmblock



05.7)

Optional:

Der Quad-RGB-Sensor besteht aus vier einzelnen Licht- und Farbsensoren. Sie messen die Intensität des Lichts, das den Sensor aus den roten, grünen und blauen Bereichen des Lichtspektrums erreicht. Dies ermöglicht es dem Sensor, die Farben von Objekten direkt vor ihm zu erkennen, wie Markierungen auf dem Boden, und ermöglicht es dem Roboter auch, einer schwarzen Linie zur Orientierung zu folgen.

Der Quad-RGB-Sensor wird in Anwendungen wie einem Lagerroboter verwendet, der sich durch das Lager bewegt. In diesem Programmbeispiel fährt der mBot2 vorwärts, wenn der Roboter die Farbe Weiß sieht.

```

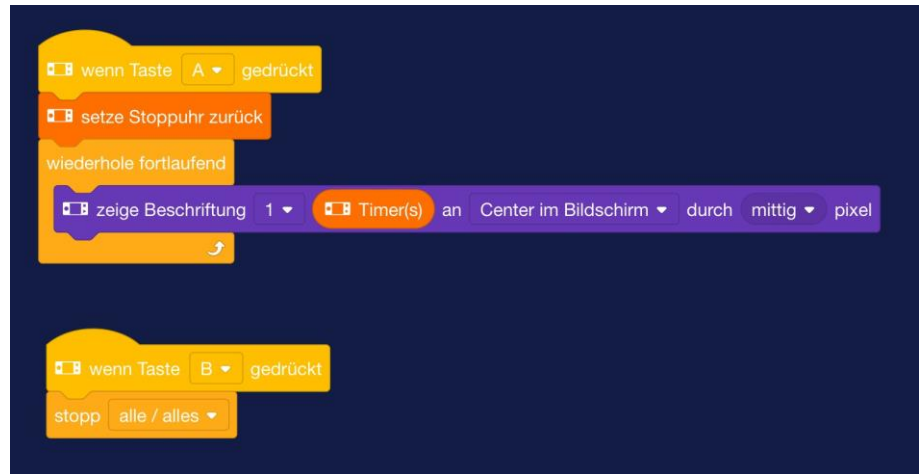
wenn Taste A gedrückt
  wiederhole fortlaufend
    wenn Dualer RGB-Sensor 1 die Sonde 2 Erkennt Farbe weiss dann
      moves forward um 50 U/min
    sonst
      Encoder-Motor anhalten alle / alles
  
```



05.8)

Optional:

Ein Timer ist eine Art Stoppuhr, die die Zeit in Sekunden seit dem Einschalten oder Zurücksetzen des CyberPi anzeigt. Das folgende Beispiel zeigt euch, wie ihr den Timer einstellen und verwenden könnt. Durch Drücken der Taste A wird der Timer auf Null zurückgesetzt und dann dauerhaft im Display angezeigt. Ihr könnt das Display und das gesamte Programm durch Drücken der Taste B beenden. Versucht, das Programmbeispiel zu erweitern, damit der Timer zu zählen beginnt, wenn der mBot2 sich zu bewegen beginnt.



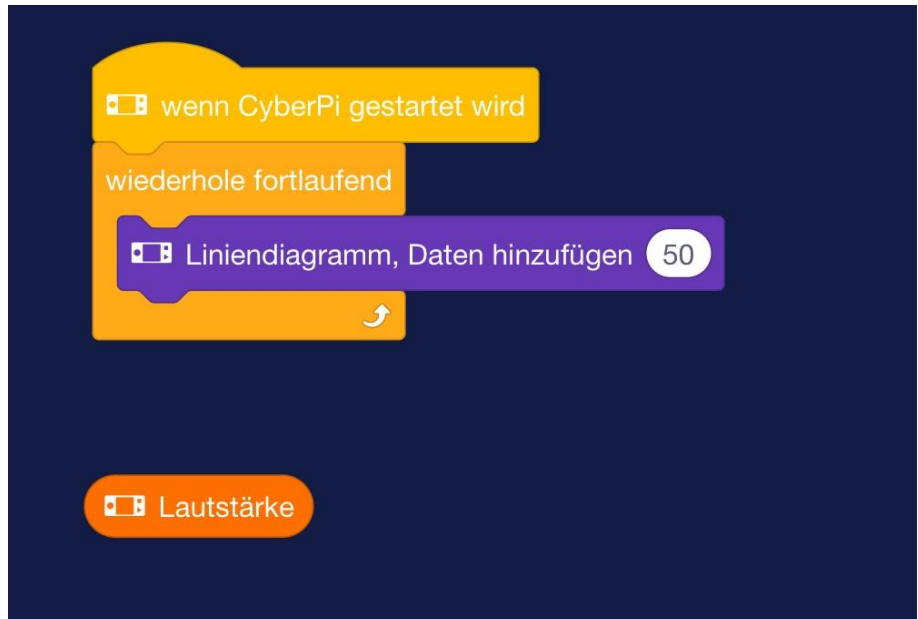
05.9)


Beim Testen der Programmbeispiele habt ihr gesehen, dass jeder Sensor etwas auf dem Display des CyberPi anzeigt. Alles, was ein Sensor registriert, wird als Daten bezeichnet. Ihr könnt die Daten von einem Sensor auf dem Display des mBot2 anzeigen lassen.

In den Programmierbeispielen, die ihr gesehen habt, könnt ihr die Daten beispielsweise mit einer Zahl oder einem Text anzeigen lassen. Ihr könnt die Daten von den Sensoren auf verschiedene Arten auf dem Display anzeigen.

Ihr könnt dafür verschiedene Codeblöcke verwenden.

Schritt 01)



<p>Diese Blöcke finden Sie in mBlock unter der Kategorie "Display" zu sehen.</p> <p>Wiederholt die Programmierbeispiele und testet Sie im Live-Modus. Schaut euch an, was auf dem Display passiert.</p>	<p>Schritt 02)</p> 
	<p>Jetzt werdet ihr selbst Daten mit den Umgebungssensoren sammeln und sie auf dem Display anzeigen. Ihr werdet diese Daten sammeln, während ihr den mBot2 in eurem Zimmer herumfahren lässt. Welche Daten ihr sammelt und wie ihr sie auf dem Display des CyberPi anzeigen lässt, liegt ganz bei euch. Natürlich könnt ihr selbst viel experimentieren.</p>

