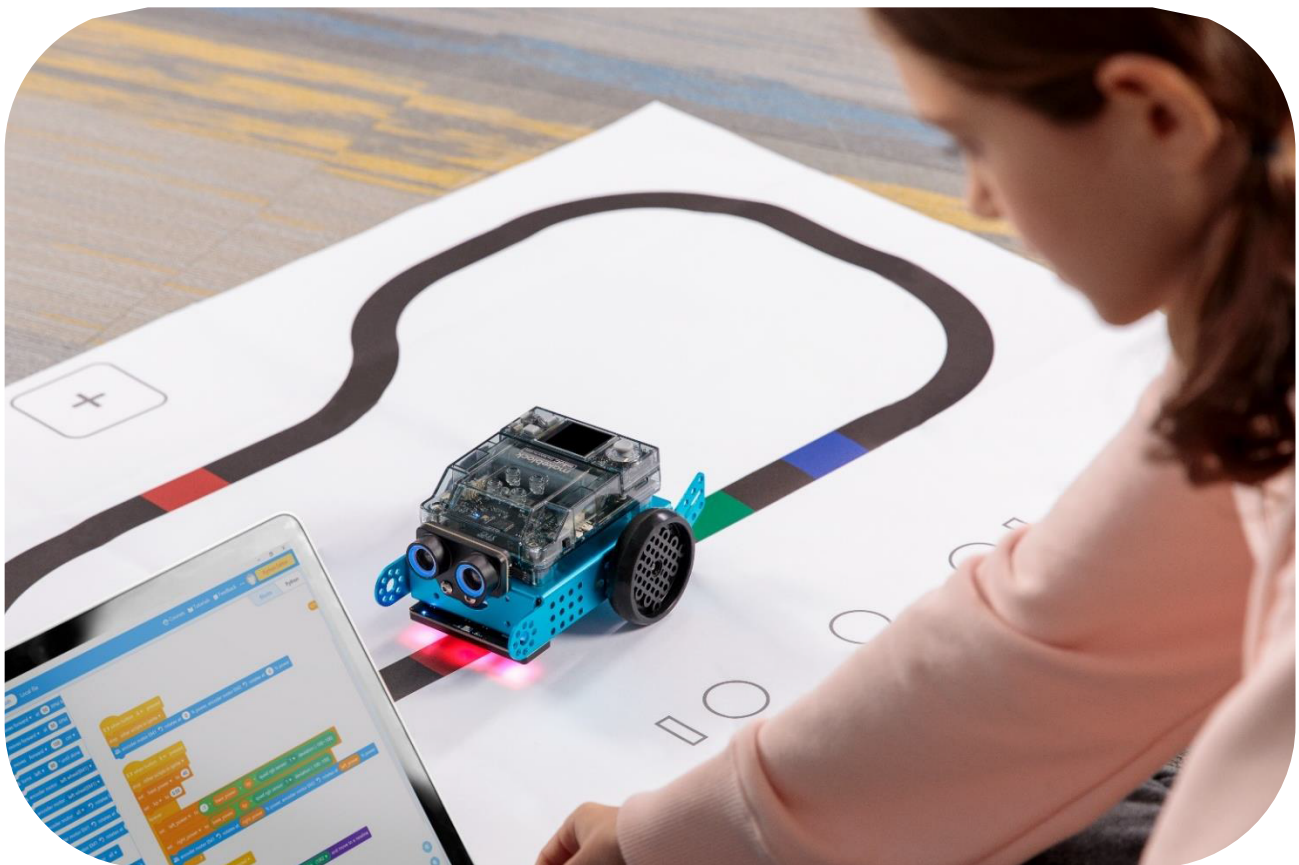


mBot2

Unterrichtseinheiten zum Einstieg



Inhalt

Einführung	2
Unterrichtseinheit 1: In Bewegung kommen	12
Unterrichtseinheit 2: Sensorik = Daten	22
Unterrichtseinheit 3: Dem mBot2 zuhören	38
Unterrichtseinheit 4: „Sehen“ mit Ultraschall	49
Unterrichtseinheit 5: Stadtbesichtigung	56
Unterrichtseinheit 6: Vorsichtiges Fahren	72
Unterrichtseinheit 7: Ein Netzwerkspiel	86
Unterrichtseinheit 8: mBot2 zu Ihren Diensten	98
Unterrichtseinheit 9: mBot2 in freier Wildbahn	116



Einführung

1. Pädagogischer Ansatz


Mit Makeblock Education ist die Entwicklung von Wissen und Fähigkeiten die Kombination von digitalen und physischen Tools, die zusammenarbeiten, um eine praktische Lernerfahrung zu bieten. Die Schüler setzen sich aktiv mit dem Thema der Lektion auseinander, um ein Problem zu lösen oder etwas Neues zu schaffen. So können sie theoretische Konzepte aus der Praxis heraus entwickeln sowie diese Konzepte auf praktische Aufgaben übertragen und anwenden, was ihr Verständnis vertieft. Dazu gehört auch das Lernen aus Fehlern durch detaillierte und systematische Analyse der beteiligten Prozesse.

Beim praktischen Lernen steuern und fördern die Schüler ihren Bildungsprozess, indem sie während des Unterrichts eine aktivere Rolle einnehmen, anstatt nur einem Vortrag des Kursleiters zuzuhören.

Diese Einführung wird einen Überblick über die neuen Funktionen von mBot2 und die Vorteile für den Einsatz im Unterricht geben. Sie zeigt kurz die Schritte, um mit der Programmierung des Roboters mit mBlock5, dem blockbasierten Editor von Makeblock Education, zu beginnen, und porträtiert dann die einzelnen Einführungsaktivitäten. Diese Aktivitäten stellen eine stufenweise Lernerfahrung dar und decken die neuen Funktionen Schritt für Schritt ab. Ausgehend von realen Bezügen in den gestellten Aufgaben werden die Hintergründe von Sensoren, Aktuatoren und rechnerischem Denken besprochen, zusammen mit Beispielcodes, die einfach zu verstehen und für jede Aufgabe zu vertiefen sind. Die Aktivitäten regen immer dazu an, weitere digitale Medien einzubeziehen und weiterzudenken, um das Gelernte auf neue Herausforderungen anzuwenden und enden mit einer Reflexion über die Aktivität.

2. Eine Einführung zu mBot2

mBot2 ist ein Bildungsroboter der nächsten Generation, der für den Informatik- und STEAM-Unterricht entwickelt wurde. Seine erweiterten Fähigkeiten machen ihn zu einer idealen Einstiegslösung für die Sekundarstufe I, aber er kann auch bis zur Sekundarstufe II und darüber hinaus eingesetzt werden. mBot2 wurde für Schüler entwickelt, um interaktive und intelligente



Lektionen auszuführen, die fesselnd sind, Spaß machen und reale Anwendungen von Spitzentechnologien widerspiegeln.

mBot2 wird von CyberPi angetrieben, einem leistungsstarken und vielseitigen Mikrocontroller. Seine integrierten Sensoren und Aktuatoren - wie Mikrofon und Lautsprecher, Inertialmesseinheit mit Gyroskop und Beschleunigungssensor, ein Lichtsensor, Schaltflächen für die Menübedienung und ein Farbdisplay - werden durch einen Ultraschall-Entfernungssensor und einen Linienverfolgungssensor mit vier RGB-Elementen ergänzt.

Neben dieser Auswahl an Sensoren und Aktuatoren ist der mBot2 Wi-Fi-kompatibel, was eine Vielzahl von Anwendungen zu Lehrplanthemen wie Coding, Robotik, Data Science und Künstliche Intelligenz in Verbindung mit anderen Fächern wie Mathematik, Physik usw. ermöglicht.

Lehrer können z. B. mehrere mBots in einem Klassenzimmer drahtlos miteinander verbinden, um ein lokales Netzwerk von Robotern zu schaffen, die untereinander kommunizieren, Informationen austauschen und Aufgaben ausführen. Sie können auch einen eigenständigen CyberPi als intelligentes Gerät verwenden, um mit dem mBot2 zu kommunizieren und so ein intelligentes Ökosystem oder eine spielerische Fernsteuerung zu schaffen. Wenn er mit dem Internet verbunden ist, kann der mBot2 erweiterte Funktionen wie Spracherkennung ausführen oder mit der Cloud kommunizieren, um Informationen zu erhalten.

Die für den mBot2 spezifischen Sensoren und Aktuatoren werden in den folgenden Unterabschnitten vorgestellt, und die gesamte Produktpalette ist Teil der Einstiegsaktivitäten. Eine Übersicht über sie finden Sie auch am Ende dieses Dokuments.

Hier finden Sie eine Übersicht über die verschiedenen Komponenten, aus denen der mBot2 besteht:

mBot2 Schutzschild

Kompatibel mit einer Vielzahl von externen Komponenten, enthält eine wiederaufladbare eingebaute Batterie.

Ultraschallsensor 2

"Die Objekterkennung wird von 8 programmierbaren LEDs für eine verbesserte Interaktion unterstützt".

Quad-RGB-Sensor

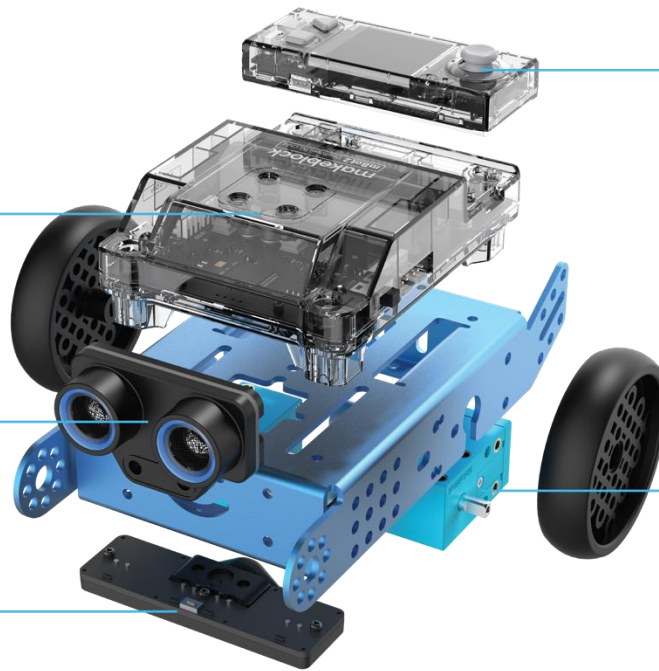
"4 Sensor-Sonden unterstützen die Farberkennung, sowie einfache und erweiterte Linienerkennungsprogramme".

CyberPi

Mikroprozessor für drahtlose Kommunikation. Kompatibilität mit blockbasierter und Python-Kodierung.

Encoder-Motoren

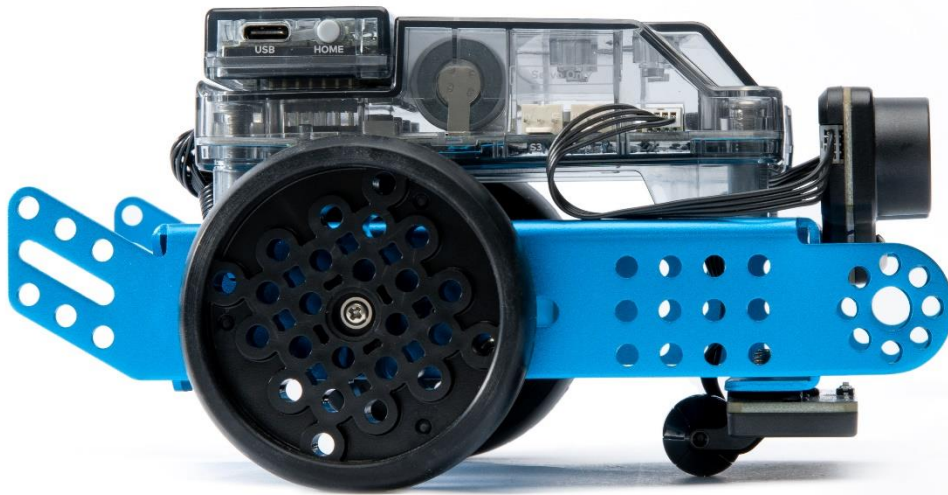
Messung der Rotation mit 1-Grad-Genauigkeit, den zurückgelegten Strecke und präzise Steuerung der Geschwindigkeit bis zu 200 U/min.



2.1 Encoder-Motoren

Die mBot2-Motoren sind mit optischen Encodern ausgestattet, die eine hochgenaue Steuerung ermöglichen. Damit können die Schüler die Rotation, Geschwindigkeit und Position der Räder und des Roboters präzise steuern. Zusätzlich können die Motoren auch als Servos und sogar als Knöpfe verwendet werden, um Rückmeldungsdaten an das System zu übertragen, genau wie ein Sensor.

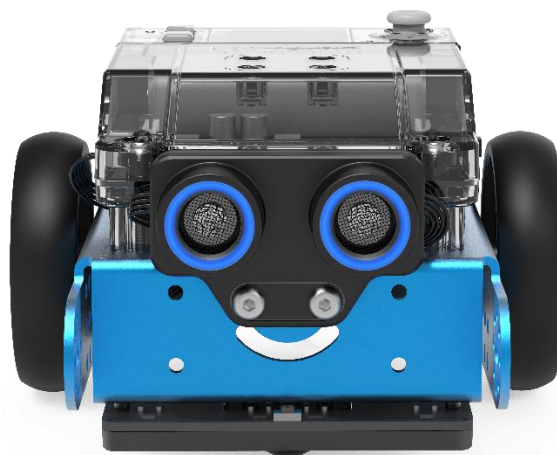
Dank dieser Funktionen kann der Unterricht im Vergleich zu Robotern, die weniger Kontrolle über ihre Motoren haben, realistischer und lehrreicher sein. Bei den Aktivitäten können mathematische Konzepte richtig integriert werden, z. B. durch das Fahren von präzisen Entfernungen, das Berechnen von exakten Kurven und sogar das Abbilden einer Route durch ein Labyrinth und das Übertragen der Ergebnisse zurück an den Computer.



Lektion 1 der Getting Started Activities bietet eine Einführung in die verschiedenen Codierblöcke für die Encoder-Motoren in der Programmierumgebung mBlock sowie eine einfache Aufgabe für die Schüler, um die Möglichkeiten selbst zu erkunden.

2.2 Ultraschallsensor und Quad-RGB-Sensor

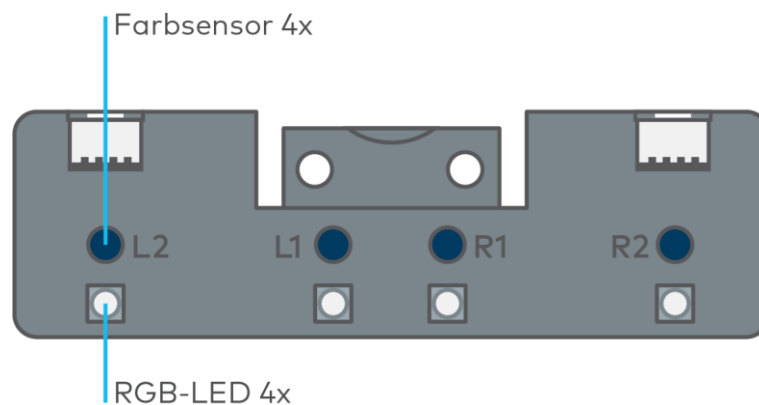
Der Ultraschallsensor, der mit dem mBot2 geliefert wird, kann genauere und konsistentere Messungen durchführen als seine Vorgänger. Außerdem enthält er blaue LEDs, die neue Möglichkeiten der Interaktion mit dem Roboter unterstützen. Die Schüler können diese zusätzlichen Leuchten verwenden, um Antworten anzuzeigen.



mBot2 integriert eine weitere fortschrittliche Komponente: den Quad-RGB-Sensor. Dieser Sensor ist in der Lage, nicht nur Farben zu erkennen, sondern auch Linien zu verfolgen, um dem mBot2 zu helfen, Pfaden zu folgen, oder Kreuzungen oder 90°-Drehungen zu erkennen - alles gleichzeitig.

Die grundlegenden Funktionen können mit wenig Vorkenntnissen ausgeführt werden, aber aufgrund seiner erweiterten Funktionen ist es wichtig zu wissen, was die besten Praktiken sind, um das Beste aus dem Sensor herauszuholen.

Anstatt mit 2 Sensoren eine einzelne Linie vor dem Roboter zu verfolgen, verfügt der Quad-RGB-Sensor über 4 Sensoren, die es ihm ermöglichen, ein breiteres Spektrum an Bedingungen zu erkennen: Während er einer Spur folgt, kann der Sensor Abzweigungen auf der linken, rechten oder sogar beiden Seiten erkennen, was bedeutet, dass der mBot2 durch komplexere Karten navigieren kann als andere Roboter. Und da jeder der vier Sensoren ein eigener Farbsensor ist, können farbige Markierungen Positionen auf der Karte anzeigen - der Roboter "kennt" seinen Standort.

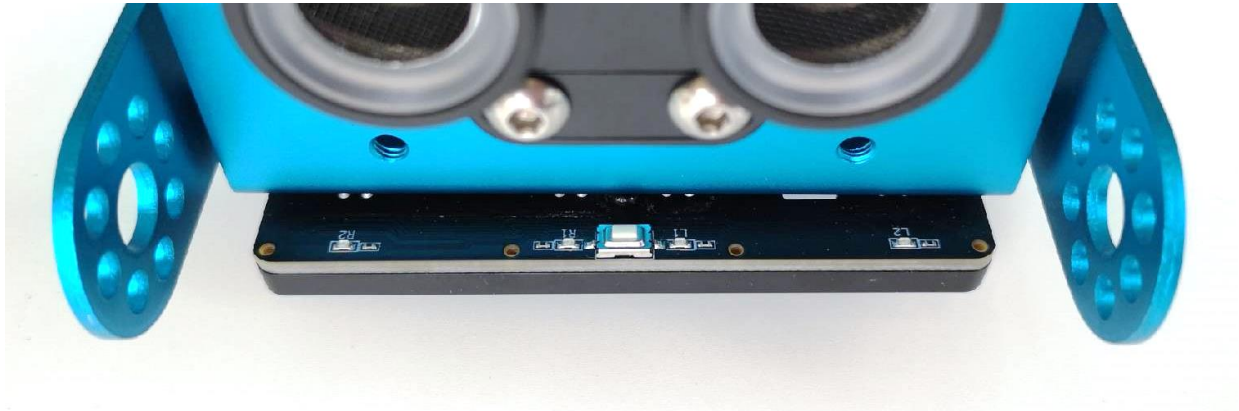


Dieser größere Erfassungsbereich kann für Karten mit einem Liniennetz genutzt werden, in dem sich mehrere mBot2 zwischen verschiedenen Stationen bewegen, wie in einem "Smart Warehouse", oder die Straßen einer Stadt simulieren, mit ihren unterschiedlichen Verkehrssignalen und Verkehrsregeln.

Bisherige Linienfolgesensoren verwendeten Infrarotlicht, das für das menschliche Auge nicht sichtbar ist. Diese Sensoren arbeiteten hingegen mit einer einzigen Wellenlänge, während der neue Sensor, genau wie das menschliche Auge, drei verschiedene Wellenlängen gleichzeitig betrachtet: rot, grün und blau. Eine Mischung aus der Intensität dieser Wellenlängen wird von unserem Gehirn als Farbwahrnehmung interpretiert. Der Sensor arbeitet ähnlich und versucht neben der Farbwahrnehmung, eine Spur vom Hintergrund für Linienverfolgungsaufgaben zu unterscheiden.

Mit der mitgelieferten Karte haben wir zusätzliche Farbmarkierungen innerhalb der Spur, um zusätzliche Aktionen auszulösen, wenn sie entsprechend programmiert sind. Damit der Sensor diese Farben als Teil der Linie interpretiert, muss er auf die hellste Farbe der Spur kalibriert

werden, die gelb ist. Platzieren Sie den Sensor über dem gelben Farbcode, während der mBot2 eingeschaltet ist, und doppelklicken Sie auf den kleinen Knopf auf der Oberseite des Sensors (siehe Abbildung unten). Wenn die LEDs zu blinken beginnen, streichen Sie mit dem Sensor über die Farbe und den weißen Hintergrund, bis das Blinken aufhört (2-3 Sek.). Danach wird die Farbe als Teil der Spur betrachtet (für eine ausführliche Erklärung siehe die zusätzlichen Informationen über den Quad-RGB-Sensor in verschiedenen Umgebungen und Lektion 5).

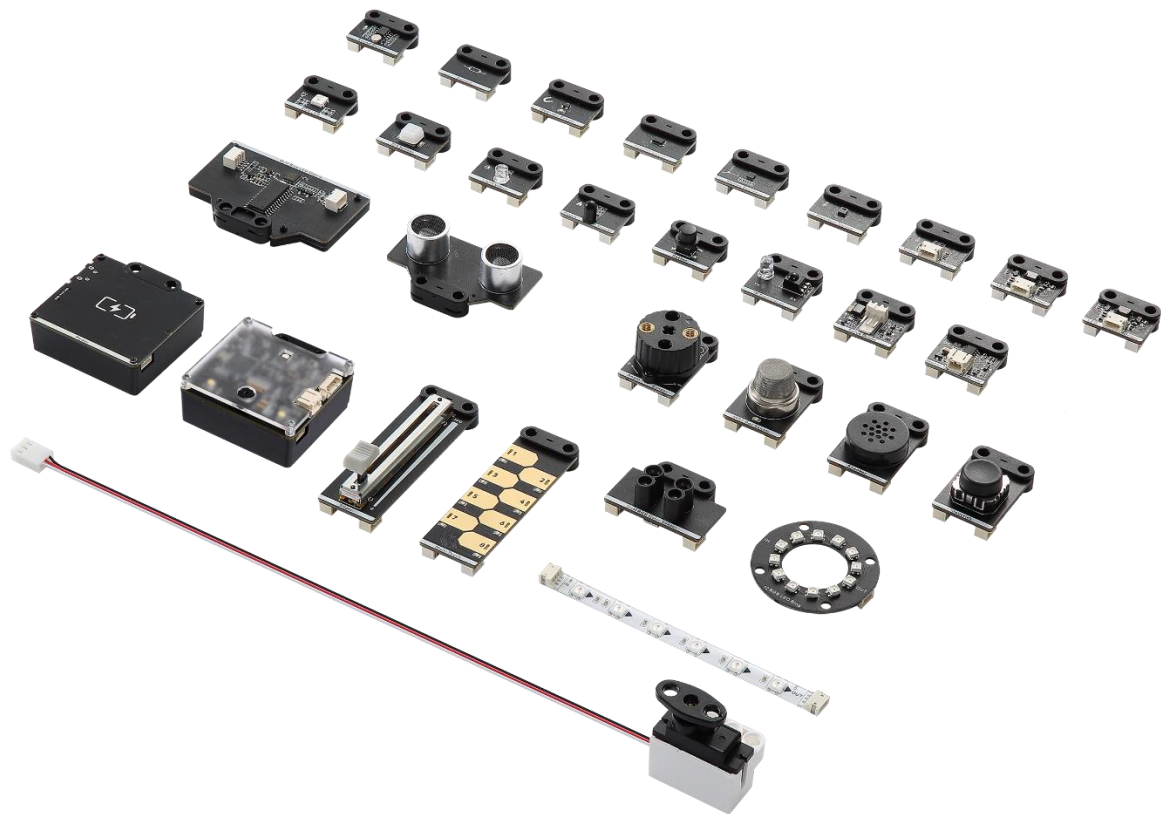


In Lektion 5 aus den "Getting Started Activities" werden die verschiedenen Blöcke erklärt, die in mBlock für den Quad-RGB-Sensor zur Verfügung stehen, und die Schüler können das Gelernte auch mit einer einfachen Aufgabe üben.

2.3 Mehr Möglichkeiten mit mBot2

Der Umfang der Aktivitäten kann noch erweitert werden, indem der mBot2 mit den von Makeblock Education entwickelten mBuild-Modulen kombiniert wird. Dank ihrer internen Micro-Controller Unit (MCU) können diese intelligenten Sensoren und Aktuatoren direkt angeschlossen werden, ohne dass eine komplexe Verdrahtung oder Einrichtung erforderlich ist, so dass die Schüler mehr Zeit für ihre Ideen und Kreationen haben. Diese Komponenten verwenden einen einzigen Steckertyp, der nicht vertauscht werden kann, so dass die Schüler beim Anschließen keine Fehler machen können. Das gibt ihnen mehr Sicherheit und damit eine bessere Lernerfahrung.

Einige Beispiele für die Sensoren und Aktuatoren sind: Smart-Kamera, Multi-Touch, Schieberegler, Temperatursensor, usw. Mit all diesen Sensoren haben Lehrer eine große Auswahl, um reale Szenarien zu erstellen und sich an die Bedürfnisse des Unterrichts und des Lehrplans anzupassen - und an die Interessen der Schüler.



Aber schon mit dem mBot2 allein haben Sie viele Möglichkeiten (siehe die Aktivitäten). Zusätzlich bieten die mBot2 Shields weitere Anschlüsse, denn über die integrierten 2-Pin- und 3-Pin-Schnittstellen können DC-Motoren, Servos, LED-Streifen und sogar Komponenten von Drittanbietern direkt angeschlossen werden, die allgemein verfügbar sind, einschließlich einer Vielzahl von Arduino-kompatiblen Sensoren, oder Sie können sogar eigene Sensoren herstellen.

3. Programmierung des mBot2

mBlock ist die Codierungsplattform für mBot2, die entwickelt wurde, um ein verbessertes Bildungserlebnis und einen kontinuierlichen Wachstumspfad für den Schüler zu bieten. Dank der Erweiterungen in mBlock5 können Lehrkräfte einige der neuesten und einflussreichsten Technologien, wie das Internet der Dinge oder künstliche Intelligenz, einfach in ihren Unterricht einbeziehen. Außerdem bietet mBlock5 durch die Integration von blockbasiertem Coding und Python einen Lernpfad für Schüler, um sich von grundlegenden bis hin zu professionellen rechnerischen Fähigkeiten weiterzuentwickeln.

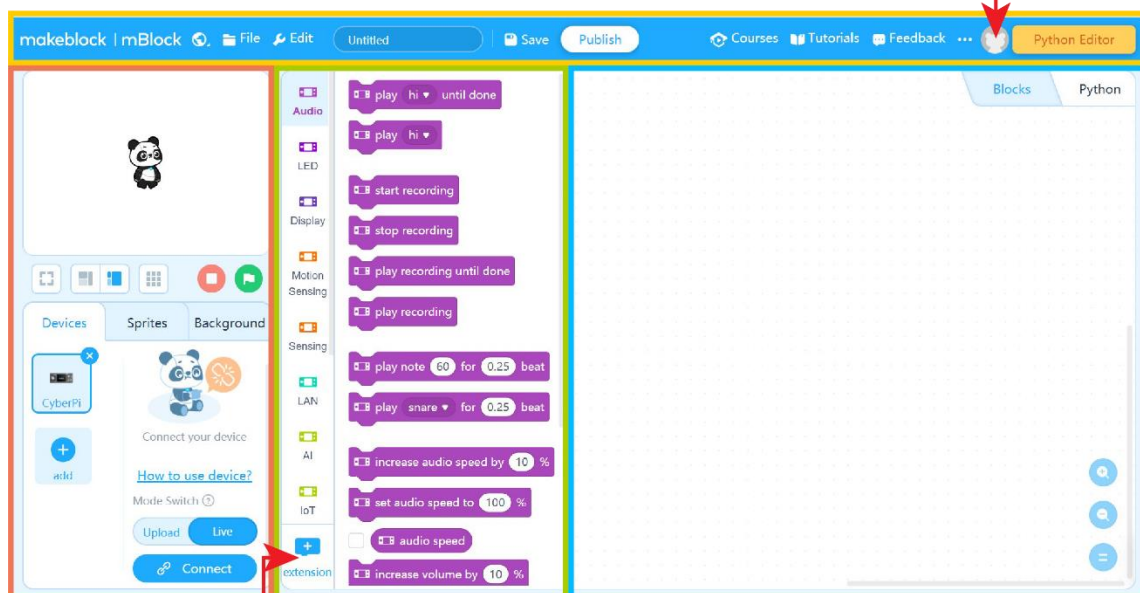
mBlock5 kann als installierte Software auf Computern, Laptops und mobilen Geräten oder in einem Webbrowser verwendet werden. Es ist kompatibel mit verschiedenen Betriebssystemen wie Windows, Mac, Linux, Chromebook, iOS, Android. Da mBlock5 Open Source ist, bietet es die Möglichkeit, neue Erweiterungen für Software und Hardware zu erstellen, was bedeutet, dass Lehrkräfte die Coding-Tools nach ihren Bedürfnissen anpassen können. Benutzer können auch Projekte in der Math Makeblock Community aussuchen und teilen.

Das Programmieren mit mBlock zu beginnen ist so einfach wie das Drag-and-Drop von Blöcken. Die mBlock-Schnittstelle umfasst die folgenden Elemente:

Menüleiste

Sprache wählen, Datei öffnen oder speichern, Beispielprogramme

Zugang für die Cloud-Dienste einrichten



Bühnenbereich

Anzeigen der Projektbühne, Anpassen von Figuren und Hintergründen, Anschließen von Geräten.

Programmierblöcke

Bibliothek mit Programmierblöcken, farblich sortiert nach Kategorien.

Bereich für blockbasierte Programmierung

Ziehen Sie die Programmierblöcke in diesen Bereich und verknüpfen sie sie, um angeschlossene Geräte oder Figuren sowie den Bühnenbereich zu steuern.

Erweiterungszentrum

Detaillierte Informationen über die Eigenschaften und die Verwendung von mBlock finden Sie unter <https://education.makeblock.com/help/category/mblock-block-based/>

4. Übersicht über die Aktivitäten

Die mBot2-Einstiegsaktivitäten wurden für Schüler im Alter von 11 bis 14 Jahren und ihre Lehrkräfte entwickelt. Insgesamt sind die Lektionen so gestaltet, dass sie trotz unterschiedlicher und ansteigender Komplexitätsstufen allgemein zugänglich sind.

Jede Lektion stellt eine andere und spannende Funktion des mBot2 vor, zusammen mit einigen grundlegenden Konzepten der Programmierung. Das in jeder Lektion erworbene Wissen hilft dabei, auf weiterführende Aktivitäten aufzubauen, während die Lektionen mit zunehmender Komplexität für den Lernenden ansteigen. Zum Beispiel werden die Bewegungen des Roboters in Lektion 1 eingeführt und dies wird in vielen der nachfolgenden Lektionen verwendet. Die Möglichkeiten des Quad-RGB-Sensors werden in Lektion 5 vorgestellt, um später in Lektion 8 verwendet zu werden.

Die Lektionen werden im Folgenden kurz beschrieben:

Aktivitätsname	Beschreibung	Schlüsselkonzepte
1. In Bewegung kommen	Die Schüler entdecken den mBot2 und die mBlock-Software und lernen, wie sie den Roboter präzise steuern können. Dieses Wissen wird in den meisten der folgenden Lektionen verwendet werden. Die Schüler entwerfen außerdem ein einfaches Labyrinth und programmieren den mBot2 so, dass er (manuell) durch das Labyrinth navigiert.	Präzise Bewegungen und entsprechende Codierblöcke.
2. Sensorik = Daten	Die Schüler arbeiten mit den verschiedenen im mBot2 integrierten Sensoren; sie lernen, diese mit den entsprechenden Codeblöcken zu verwenden und die Daten der Sensoren auf dem integrierten Vollfarbdisplay zu visualisieren.	Funktionsweise der Sensoren. Verschiedene Ansätze zur Darstellung und Visualisierung von Daten auf dem Display. Unterschiede zwischen Live- und Upload-Modus im mBlock 5.
3. Dem mBot2 zuhören	Die Schüler lernen, wie sie den Lautsprecher und das Mikrofon mit den Codeblöcken in mBlock 5 steuern können. Sie werden auch ein Programm erstellen, bei dem der mBot2 einen aufgenommenen Ton abspielt, wenn er eine bestimmte Bedingung erfüllt, während er herumfährt.	Text zu Sprache (TTS) und Spracherkennung (Speech to Text, STT) unter Verwendung des eingebauten Lautsprechers und Mikrofons. Ausführen mehrerer Aufgaben nebeneinander.
4. Sehen mit Ultraschall	Die Schüler lernen, was Ultraschall ist und wie er in einem Sensor verwendet wird. Außerdem erstellen sie ein Programm, um den mBot2 in einer Schleife fahren zu lassen, indem sie nach dem Erkennen von Hindernissen auf der Straße wenden.	Erkennen eines Hindernisses oder eines Bereichs mit Hilfe des Ultraschallsensors. Verwendung von Schleifen und bedingten Anweisungen, um den mBot2 fahren zu lassen, während er Hindernissen ausweicht.

5. Stadtbesichtigung	Die Schüler lernen, wie ein Farbsensor funktioniert, wie sie im wirklichen Leben verwendet werden, und sie programmieren den mBot2 so, dass er zu einem Tourbus wird, der verschiedene Sehenswürdigkeiten in einer Stadt besucht. Dieses Wissen wird auch in den Lektionen 7 und 8 angewandt.	Funktionsweise (Physik des Lichts) des Farbsensors/Linienverfolgers. Farb- und Linienerkennung - Der mBot2 soll einer Linie folgen und Aktionen basierend auf der Farberkennung durchführen.
6. Vorsichtiges Fahren	Die Schüler lernen den Umgang mit dem Gyroskop-Beschleunigungsmesser des mBot2 und seinen Codeblöcken und programmieren den mBot2 so, dass er sein Fahrverhalten anpasst, wenn er Neigungen auf der Straße feststellt.	Arbeitsweise von Gyroskop und Beschleunigungsmesser (als Inertial Measurement Unit, IMU). Codierung des mBot2 zur Anpassung an die Straßenbedingungen auf Basis der IMU-Daten.
7. Ein Netzwerkspiel	Die Schüler lernen, wie mehrere mBot2 drahtlos miteinander kommunizieren können, ohne dass ein WIFI-Zugangspunkt erforderlich ist. Sie werden ein einfaches Spiel programmieren, bei dem mehrere mBot2 nach einer Farbe suchen und der erste, der sie findet, gewinnt. Dieses Wissen wird auch in Lektion 8 verwendet.	Drahtlose Datenübertragung in Ad-hoc-Netzwerken. Datenaustausch in Schleifen und Ereignissen.
8. mBot2 zu Ihren Diensten	Durch das Erlernen des Aufbaus einer WIFI-Verbindung mit dem mBot2 lernen die Schüler auch die Verwendung der integrierten Spracherkennung und wenden dieses Wissen in einer Aktivität an, in der der mBot2 zu einem Roboterkellner wird, der mit seinen Kunden spricht.	Verwendung des WIFI-Infrastrukturmodus mit dem mBot2 für Spracherkennung und Sprachsynthese. Auslagerung von schwerer Rechenarbeit wie Spracherkennung an Cloud-Services. Strukturierung des Codes durch Anwendung "eigener Blöcke" (Funktionen).
9. mBot2 in freier Wildbahn	In dieser speziellen Lektion lernen die Schüler einige Prinzipien der künstlichen Intelligenz durch die Verwendung der Teachable Machine-Erweiterung in mBlock5. Sie werden ihr Wissen anwenden, um ein natürliches Ökosystem nachzubilden, in dem sich der mBot2 wie ein Tier verhält.	Sie lernen über maschinelles Lernen und wenden es mit lokaler Verarbeitung nur auf blockbasierte Programmierung an. Ein neues Kommunikationsprotokoll zwischen dem mBot2 und dem Computer einrichten.

Während des gesamten Unterrichts werden die Schüler ermutigt, mit Zustimmung der Lehrer ihre Lernergebnisse per Video zu dokumentieren und zu veröffentlichen, als eine Möglichkeit, Stolz auf ihre Arbeit zu empfinden und eine Diskussion über Computer Science und STEAM im Klassenzimmer zu führen.

Lektion 1: In Bewegung kommen!

Thema: STEAM

Klasse(n): 5. Klasse und aufwärts

Dauer: 45 Minuten

Schwierigkeitsgrad: Anfänger

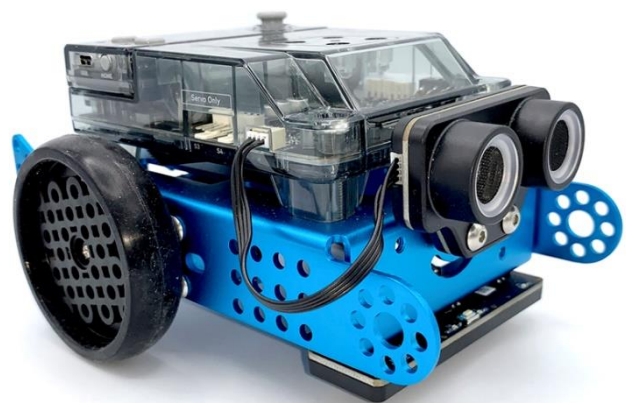
★ Lektionsziele

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Den mBot2 mit Präzision zu fahren.

★ Überblick

Roboter sind autonome Maschinen, die menschliche Arbeit ersetzen; sie können ihre Umgebung wahrnehmen und Computerprogramme ausführen, um Entscheidungen zu treffen und Aktionen vorzunehmen. Ein Computerprogramm ist ein Satz von Anweisungen und Bedingungen, die von einem Computer ausgeführt werden, um eine Aufgabe zu erfüllen. Sie denken vielleicht, Sie hätten noch nie einen Roboter gesehen oder benutzt, aber die Wahrscheinlichkeit ist gering. Roboter können viele verschiedene Formen annehmen und haben unterschiedliche Fähigkeiten, je nach Kontext. Zum Beispiel gibt es Roboter in Haushalten, wie den Staubsaugerroboter oder einen Rasenmäherroboter; Sie können sie auch in Fabriken finden, wo sie Produkte zusammenbauen. Wir haben sogar Roboter auf andere Planeten geschickt.



⚙️ Schwerpunkt

Am Ende der Lektion werden die Schüler wissen:

- Welche Bewegungen der mBot2 ausführen kann
- Welche verschiedenen Programmierblöcke sie verwenden können, um den mBot2 zu bewegen

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle
- A3-Papier
- Stifte

Lehrplan

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none">• Sensoren und Daten im Alltag.• Was ist CyberPi?
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none">• Kennenlernen der verschiedenen Programmierblöcke des mBot2.
25 Minuten	3. Ausprobieren <ul style="list-style-type: none">• Ihren eigenen Roboter durch ein selbstgebautes Labyrinth programmieren.
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none">• Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zur späteren Diskussion.• Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass• Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Roboter im Alltag
2. Kennenlernen des mBot2

1. Roboter im Alltag

Wir haben Roboter an vielen verschiedenen Orten.

Manche Menschen haben sie in ihrem Zuhause und andere arbeiten täglich mit ihnen. Roboter können viele verschiedene Formen annehmen und haben unterschiedliche Fähigkeiten, je nach Kontext. Können Sie sich drei Roboter vorstellen, mit denen Sie im Alltag (regelmäßig) zu tun haben?



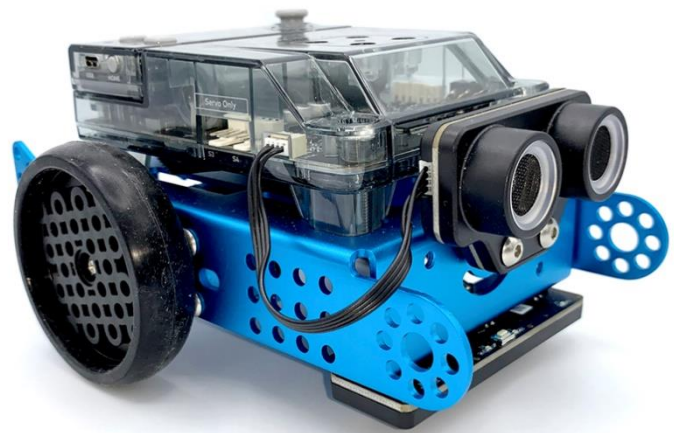
Wahrscheinlich gibt es noch viel mehr Roboter, die entwickelt wurden und die Sie noch nie gesehen haben. Suchen Sie im Internet nach drei weiteren Robotern, die Sie im Alltag verwenden können.

Es besteht eine gute Chance, dass Sie später im Berufs- oder Privatleben mit Robotern zu tun haben werden. Es gibt Roboter, die in vielen verschiedenen Unternehmen arbeiten. Normalerweise führen diese Roboter Aufgaben aus, die sich wiederholen, die mit großer Genauigkeit ausgeführt werden müssen oder die gefährlich sind. In dieser Lektion werden wir hauptsächlich daran arbeiten, einen Roboter so zu programmieren, dass er sich genau bewegt. Fallen Ihnen drei Berufe oder Unternehmen ein, in denen es sehr wichtig ist, dass ein Roboter sehr genau arbeitet?

2. Kennenlernen des mBot2

Der mBot2 ist ein programmierbarer Roboter, der mit verschiedenen Komponenten ausgestattet ist, die es ihm ermöglichen, seine Umgebung zu erfassen, zu handeln und mit ihr zu kommunizieren. Der mBot2 ist für Schüler gedacht, um etwas über Informatik und Technologie zu lernen, einschließlich Möglichkeiten, wie man einen Roboter im echten Leben einsetzen kann.

Der mBot2 kann mit Hilfe von mBlock programmiert werden. In mBlock kann das Programmieren so einfach sein wie das Drag & Drop und das Zusammensetzen von Blöcken. Wie Sie in dieser Lektion sehen werden, können Sie den mBot2 zum Beispiel so programmieren, dass er durch ein Labyrinth fährt - mit Ihrer Hilfe (nicht autonom). mBot2 wird mit einem Paar spezieller Motoren geliefert, die die Drehung der Achse und damit die Geschwindigkeit und die zurückgelegte Strecke des Roboters erfassen können. Sie werden aufgrund des eingebauten Sensors (Encoder) Encoder-Motoren genannt. Mit diesem Motortyp lassen sich bestimmte Parameter wie Drehwinkel und Geschwindigkeit der Motoren steuern. In den folgenden Schritten erfahren Sie, wie Sie diese Motoren nutzen können.



2. Praxisorientiert (10 Min.)

Schritt 2: Praxisorientiert

Dieser Schritt besteht aus zwei Teilabschnitten:

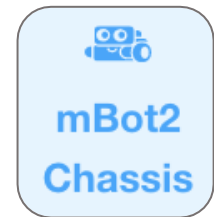
1. Die verschiedenen Programmierblöcke des mBot2 kennenlernen.
2. Einige Programmierbeispiele zur Steuerung des mBot2 nachbilden und testen.

1. Kennenlernen der verschiedenen Programmierblöcke des mBot2

Wie Sie bereits festgestellt haben, können Roboter auf viele verschiedene Arten eingesetzt werden. Wenn Sie sich einen Zweck für einen Roboter überlegt haben, bedeutet das höchstwahrscheinlich, dass Sie wissen, was der Roboter tun muss, um Ihnen zu helfen, und

einige dieser Aufgaben müssen möglicherweise mit einer bestimmten Präzision ausgeführt werden.

Wenn Sie mit der Programmierung des mBot2 beginnen, werden Sie feststellen, dass es viele verschiedene Codierblöcke gibt, die Sie verwenden können, um den Roboter zu bewegen. Sie finden diese Blöcke im mBlock 5 in der Kategorie "Chassis". Diese Codierblöcke sind blau.



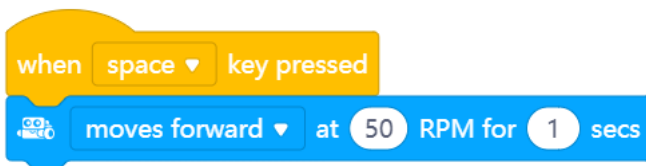
Unten sehen Sie einige Beispiele für die Codierblöcke, die für die Bewegung des Roboters benötigt werden. Einige von ihnen sehen sich ähnlich, aber sie machen alle etwas anderes. In dieser Lektion werden wir im Live-Modus arbeiten. Stellen Sie sicher, dass der richtige Modus in der Software ausgewählt ist. Die Unterschiede zwischen dem Live-Modus und dem Upload-Modus werden in Lektion 2 erklärt.

Codeblock:

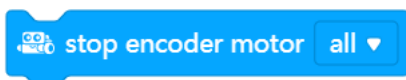


Mit diesem Codeblock können Sie den mBot2 mit einer bestimmten Drehgeschwindigkeit der Räder und für eine bestimmte Anzahl von Sekunden vorwärts, rückwärts, links und rechts bewegen.

Das folgende Beispiel zeigt, wie Sie den mBot2 für zwei Sekunden mit einer Geschwindigkeit von 50 Umdrehungen pro Minute bewegen können. Dies ist z. B. nützlich, wenn der mBot2 eine Last vorwärts schieben muss.

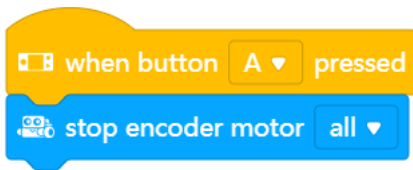


Codeblock:



Mit diesem Codeblock können Sie Ihren Roboter dazu bringen, anzuhalten. Dies kann ein nützlicher Block während einer Testphase sein. Wenn ein Programm nicht wie erwartet funktioniert, können Sie diesen Block verwenden, um den Roboter sofort anzuhalten.

So können Sie z. B. dafür sorgen, dass der Roboter anhält, wenn die A-Taste des CyberPi gedrückt wird. Verwenden Sie dazu das folgende Programmierbeispiel.

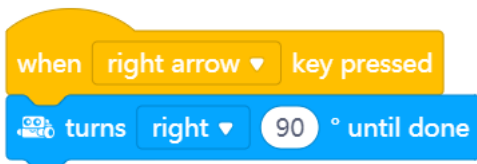
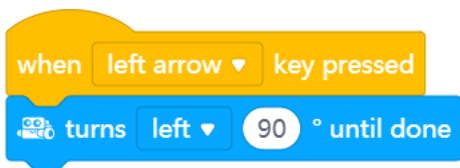


Codeblock:



Mit diesem Codeblock können Sie den mBot2 um eine bestimmte Gradzahl rotieren lassen, und Sie können wählen, ob die Rotation nach links oder nach rechts erfolgen soll.

Wenn das unten stehende Programmierbeispiel eingestellt ist, kann der mBot2 mit den Pfeiltasten gesteuert werden. Wenn Sie den rechten Pfeil drücken, dreht sich der mBot2 um 90 Grad nach rechts, wenn Sie den linken Pfeil drücken, dreht er sich um 90 Grad nach links.

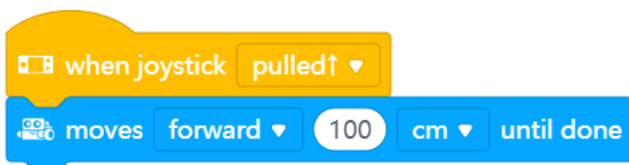


Codeblock:



Mit diesem Codeblock können Sie den mBot2 eine bestimmte Strecke vorwärts oder rückwärts fahren lassen.

Wenn Sie das unten stehende Programmierbeispiel verwenden, wird sich der mBot2 100 cm vorwärts bewegen. Um diese Bewegung zu starten, bewegen Sie den Joystick des CyberPi nach oben.







Codeblock:


 encoder motor EM1  rotates at **50** RPM, encoder motor EM2  rotates at **50** RPM


Sie können die Motoren des mBot2 auch unabhängig voneinander steuern, und dies ist einer der Blöcke, die Sie dafür verwenden können.

Im folgenden Beispiel bewegt sich der mBot2 3 Sekunden lang in einer Kurve, bevor er anhält. Beachten Sie, dass der Wert für einen der Motoren negativ ist; das liegt daran, dass die Motoren in entgegengesetzte Richtungen montiert sind. Um den Roboter also in eine Richtung zu bewegen, müssen Sie einen der Motoren in die entgegengesetzte Richtung drehen lassen. Was würde passieren, wenn sich beide Räder mit 40 RPM drehen?

 when button **B** pressed

 encoder motor EM1  rotates at **50** RPM, encoder motor EM2  rotates at **-30** RPM


 wait **3** seconds



 stop encoder motor **all**


Codeblock:

 encoder motor **(1) EM1**  rotated angle (°)

Da die Motoren des mBot2 ihre Geschwindigkeit und Rotation messen können, können Sie diese Werte auch ablesen. Die gemessene Geschwindigkeit ist die Rotationsgeschwindigkeit in Runden pro Minute ($360^\circ = 1$ Runde; im Sprachgebrauch der Wissenschaft ist es üblicher, "Grad pro Sekunde" zu verwenden ($1 \text{ RPM} = 6^\circ \text{ pro Sekunde}$)). Wenn Sie in mBlock 5 das Kästchen neben diesem Block ankreuzen, können Sie die Werte auf der Stufe oberhalb des Pandas ablesen. Im folgenden Programmierbeispiel werden die Werte der Rotation auf Null zurückgesetzt, und dann wird dem mBot2 befohlen, sich eine Sekunde lang vorwärts zu bewegen. Welchen Wert lesen Sie auf der Stufe ab, nachdem sich der Roboter bewegt hat?

 when joystick **middle pressed**

 reset encoder motor **all**  rotated angle

 moves forward at **50** RPM for **1** secs

2. Einige Programmierbeispiele zur Steuerung des mBot2 nachbilden und testen.

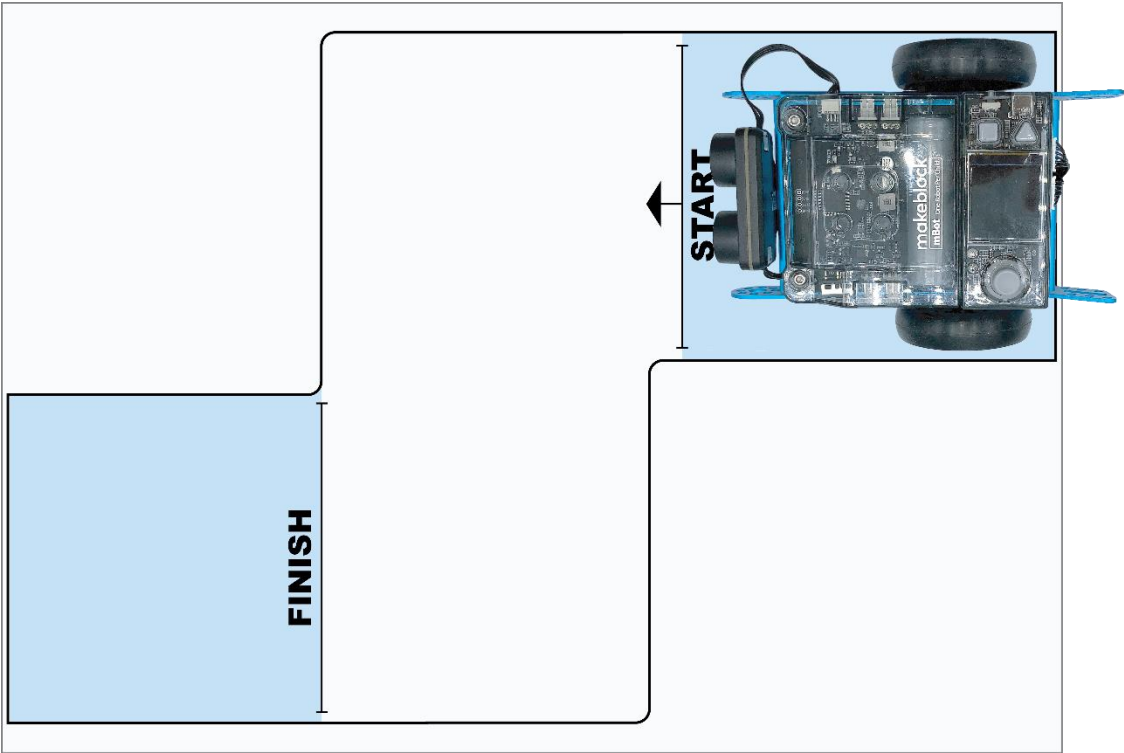
Unten sehen Sie einige Paare von Blöcken. Probieren Sie jedes Paar nebeneinander aus. Was passiert dabei? Was sind die Unterschiede zwischen den Codeblöcken?

	Option 1	Option 2
1		
2		
3		

3. Ausprobieren (25 Min.)

Schritt 3: Ihn Ausprobieren

Es ist nun an der Zeit, Ihren mBot2 in Betrieb zu nehmen. Nehmen Sie ein A3-Papier und zeichnen Sie ein Labyrinth, durch das Ihr mBot2 fahren muss. Machen Sie es nicht zu kompliziert und berücksichtigen Sie die Breite des Roboters. Der mBot2 muss nicht von selbst den richtigen Weg finden. Der mBot2 muss nur die Route fahren, die Sie programmiert haben. Benötigen Sie Inspiration? Unten können Sie ein Beispiel sehen.



Verwenden Sie das Wissen, das Sie in "Schritt 2" dieser Lektion gewonnen haben. Natürlich können Sie auch selbst viel mit den verschiedenen Programmierbeispielen in mBlock5 experimentieren.

Bei der Durchführung dieser Aufgabe ist es hilfreich, den folgenden Schritt-für-Schritt-Plan zu verwenden.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none"> • Welche Strecke soll der mBot2 fahren? • Aus welchen Teilen besteht das Labyrinth? • Wie lang sind die Strecken, die der Roboter zurücklegen muss? • Muss der Roboter auch Kurven fahren? In welche Richtung und um wie viel Grad?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none"> • Was benötigen Sie zusätzlich zum mBot2?
Schritt 3: Welche Codeblöcke benötigen Sie, um den mBot2-Antrieb herzustellen?	<ul style="list-style-type: none"> • Wie werden Sie den mBot2-Antrieb herstellen? • Welche Codeblöcke werden Sie verwenden? • Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML) • Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codeblock in mBlock gibt es auch eine Hilfe.
Schritt 4: Test und Implementierung	<ul style="list-style-type: none"> • Ist die erste Version fertig? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf. • Arbeiten Sie an den Verbesserungspunkten, bis Ihr mBot2 fehlerfrei durch Ihr Labyrinth fährt.



Gelingt es, den mBot2 fehlerfrei durch das Labyrinth fahren zu lassen? Dann fordern Sie sich mit einem schwierigeren Labyrinth heraus. Oder lassen Sie einen Ihrer Klassenkameraden eine Route für Sie ausarbeiten.

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Ist es Ihnen gelungen, den Roboter komplett fehlerfrei durch das Labyrinth zu fahren?

In dieser Lektion haben Sie Roboter im Alltag und den mBot2 kennengelernt. Sie wissen, wie Sie die Bewegungen des mBot2 steuern können und welche Programmierblöcke Sie dafür verwenden können.

Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was ist Ihnen Ihrer Meinung nach gut gelungen?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?

Lektion 2: Sensorik = Daten

Thema: STEAM

Dauer: 45 Minuten

Klasse(n): 5. Klasse und aufwärts

Schwierigkeitsgrad: Anfänger

★ Lektionsziele

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Die Codeblöcke zur Bedienung der Sensoren zu erkennen und zu verwenden
- Die Daten der Sensoren auf dem Display des mBot2 anzuzeigen.
- Ihr eigenes Computerprogramm in mBlock für den mBot2 zu erstellen, um die Sensoren zu steuern und die Daten auf dem Display anzuzeigen



★ Überblick

Ein Roboter kann ein lustiges Spielzeug sein, aber wussten Sie, dass Sie einen Roboter auch in der Forschung einsetzen können? Ein Roboter hat mehrere Sensoren (= Sinne), mit denen er seine Umgebung wahrnehmen und auch Daten sammeln kann. Zum Beispiel können Sie den mBot2 Schall und Temperatur messen lassen. Die Daten können Sie auf dem Bildschirm des mBot2 ablesen und z. B. zur Erforschung der Schalldichte oder der Temperatur im Klassenzimmer verwenden.

🔗 Schwerpunkt

Am Ende der Lektion werden die Schüler wissen:

- Welche Sensoren der mBot2 hat
- Wie die Sensoren des mBot2 funktionieren
- Was der Unterschied zwischen Live-Modus und Upload-Modus ist
- Wie man die Daten der Sensoren auf dem Display des CyberPi anzeigt

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle

Lehrplan

Diese Lektion besteht aus vier Schritten und dauert insgesamt 45 Minuten.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none">• Sensoren und Daten im Alltag.• Was ist CyberPi?
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none">• Sich mit den Sensoren des mBot2 vertraut machen.• Einige Programmierbeispiele der Sensoren erweitern und testen.• Die Daten der Sensoren auf dem Display anzeigen lassen.• Den Unterschied zwischen Live-Modus und Upload-Modus kennen.
25 Minuten	3. Ausprobieren <ul style="list-style-type: none">• Ihr eigenes Programm für den Roboter schreiben.
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none">• Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zur späteren Diskussion.• Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass• Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

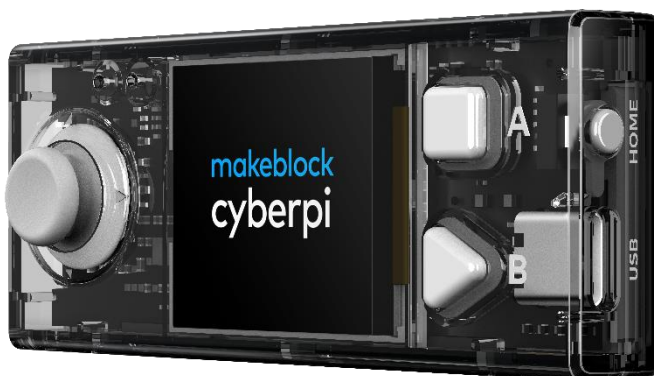
1. Sensoren und Daten im Alltag
2. Was ist CyberPi?

1. Sensoren und Daten im Alltag

Sensoren, die Daten sammeln, findet man im Alltag an vielen verschiedenen Stellen. Und zwar viel mehr, als Sie vielleicht zunächst vermuten. Zum Beispiel wird mit Hilfe von Sensoren die Außenbeleuchtung Ihres Hauses automatisch eingeschaltet, wenn es draußen dunkel wird. Oder die Heizung wird automatisch ausgeschaltet, wenn es im Klassenzimmer zu heiß wird. Die Menge des Außenlichts und die Wärme im Klassenzimmer werden alle mit Hilfe von Sensoren erfasst. Fallen Ihnen und Ihren Mitschülern noch weitere Beispiele ein?

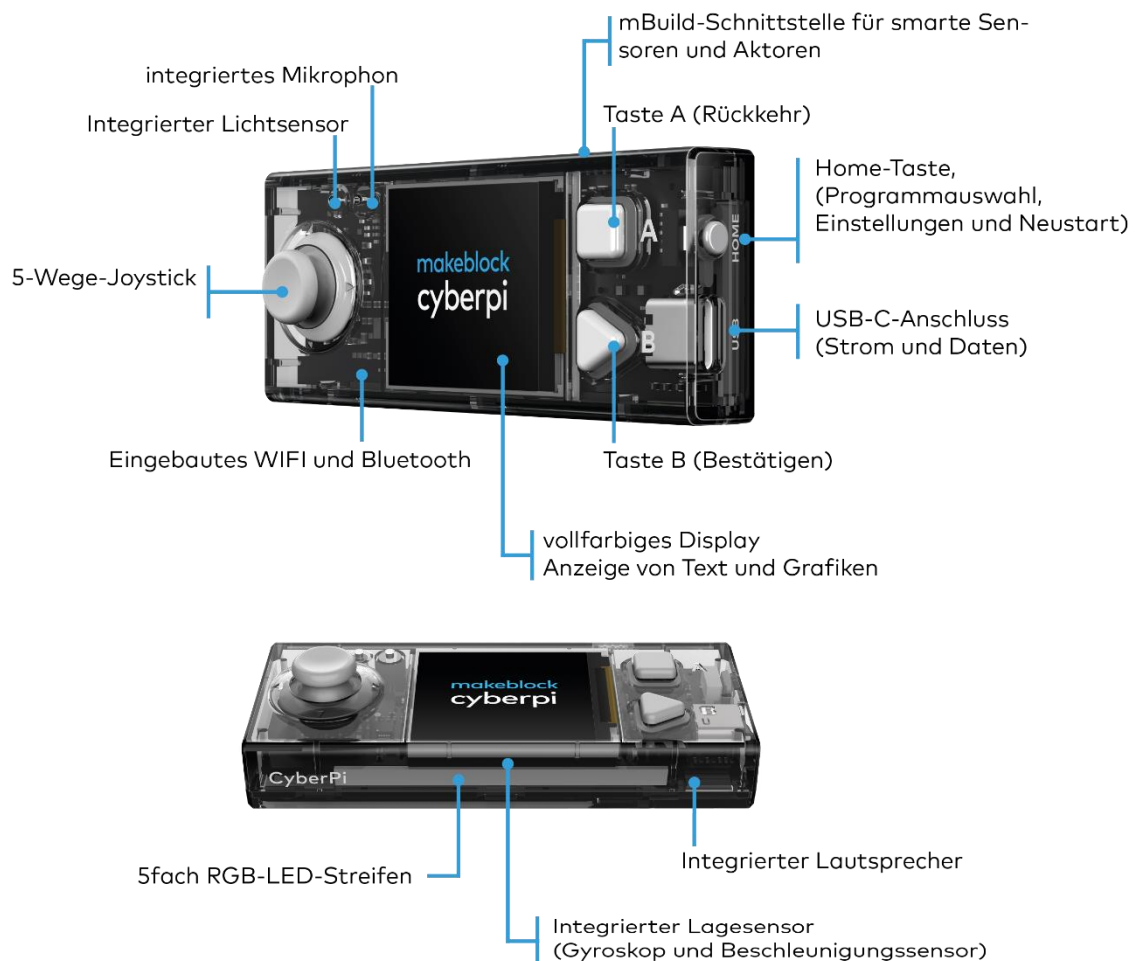
2. Was ist CyberPi?

Um den mBot2 zum Laufen zu bringen, müssen Sie ein Computerprogramm schreiben. Das tun Sie mit dem Code-Editor mBlock. Das Computerprogramm, das Sie schreiben, besteht aus einer Reihe von Befehlen, die der mBot2 ausführen muss.



Auf dem mBot2 ist ein CyberPi montiert. Ein CyberPi ist ein kleiner, programmierbarer Mikrocomputer. Sie geben dem CyberPi Befehle mit Hilfe der Codeblöcke in mBlock. Der CyberPi gibt diese Befehle an das mBot2 zurück. Sie können den CyberPi vom mBot2 entfernen.

Der CyberPi hat viele verschiedene Funktionen, wie ein Mikrofon, einen Lautsprecher und einen Joystick. Außerdem befinden sich viele Sensoren auf dem CyberPi. Sehen Sie sich die Abbildung unten an.



2. Praxisorientiert (10 Min.)

Schritt 2: Praxisorientiert

Dieser Schritt besteht aus vier Teilabschnitten:

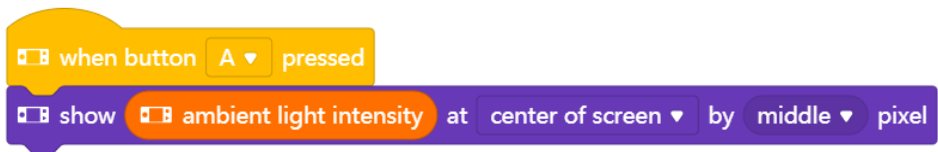
1. Sich mit den Sensoren des mBot2 vertraut machen.
2. Einige Programmierbeispiele der Sensoren erweitern und testen.
3. Die Daten von den Sensoren anzeigen.
4. Den Unterschied zwischen Live-Modus und Download-Modus kennen.

1. Kennenlernen der mBot2-Sensoren

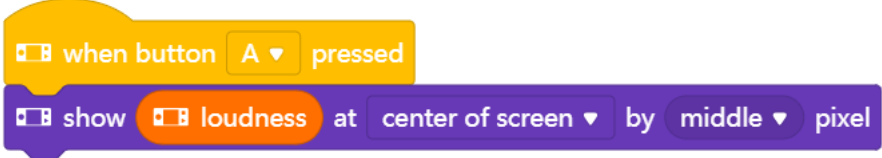
Jeder Roboter arbeitet mit Sensoren. Sensoren können mit Ihren Sinnen (= Geschmack, Tastsinn, Geruch, Hören, Sehen) verglichen werden. Durch diese Sensoren "sieht" der mBot2 seine Umgebung. Es gibt verschiedene Arten von Sensoren, die den mBot2 "sehen" lassen können, wie z. B. ein:

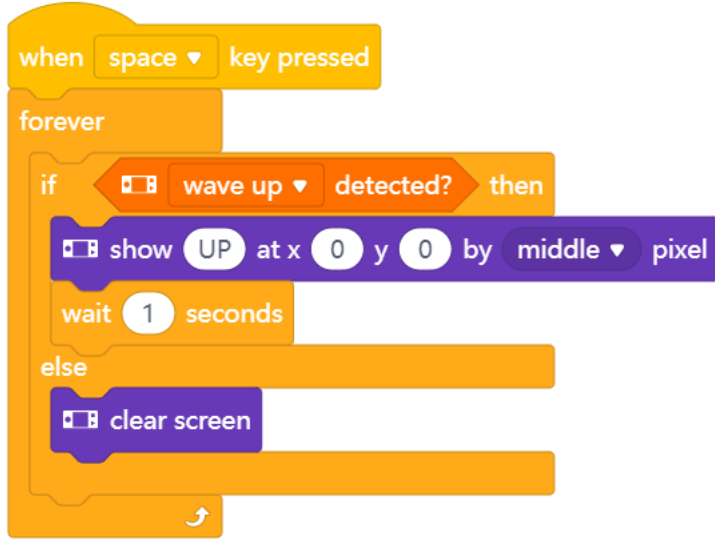
- Lichtsensor
- Schall-Sensor
- Gyroskop-Beschleunigungsmesser
- Quad-RGB-Sensor
- Timer

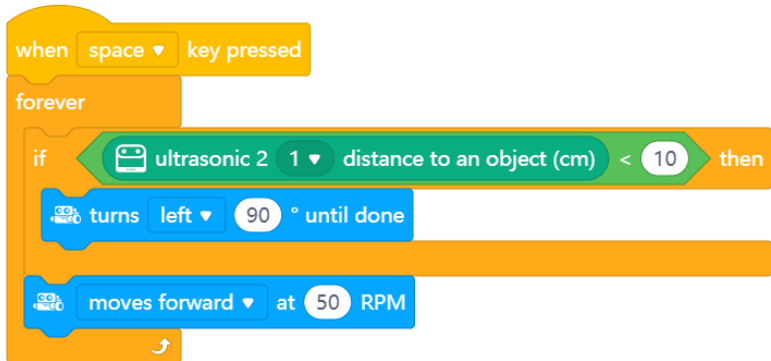
In der folgenden Tabelle sind die Sensoren des mBot2 aufgeführt. Zu jedem Sensor gibt es eine kurze Erläuterung und ein Programmierbeispiel.

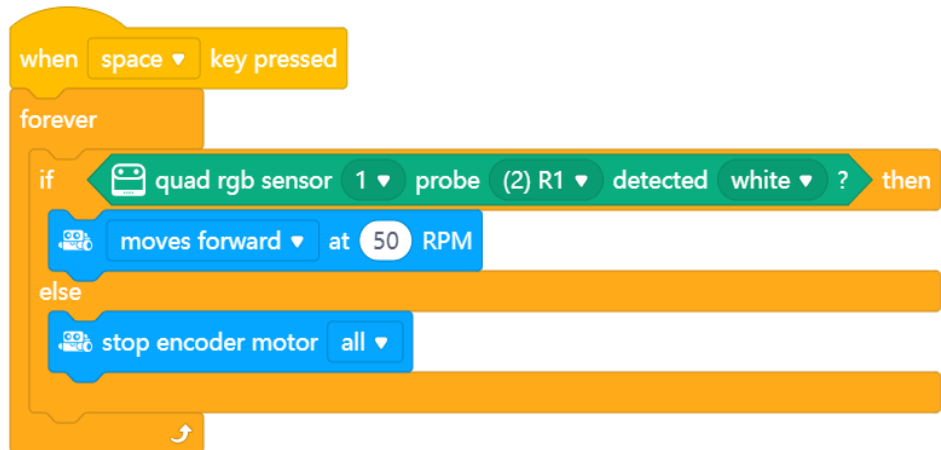
Sensor-Typ	Was macht dieser Sensor?
Lichtsensor	<p>Licht hat eine bestimmte Stärke (= Lichtenergie). Der Lichtsensor ist eine Vorrichtung, die Lichtenergie in elektrische Energie umwandelt.</p> <p>Ein Lichtsensor wird z. B. häufig verwendet, um:</p> <ul style="list-style-type: none"> • die Helligkeit des Displays von Smartphones automatisch an die Umgebung anzupassen; • die Beleuchtung in Wohnungen zu steuern oder Scheinwerfer von Fahrzeugen automatisch einzuschalten. <p>Im folgenden Programmierbeispiel wird die Lichtenergie der Umgebung auf dem Display des CyberPi angezeigt. Sie können das Programmierbeispiel so erweitern, dass der mBot2 z. B. aufhört zu fahren, wenn die Lichtenergie der Umgebung unter einem bestimmten Wert liegt, wie z. B. wenn es dunkel wird.</p>
	Programmierbeispiel
	 <pre> when button A pressed show ambient light intensity at center of screen by middle pixel </pre>



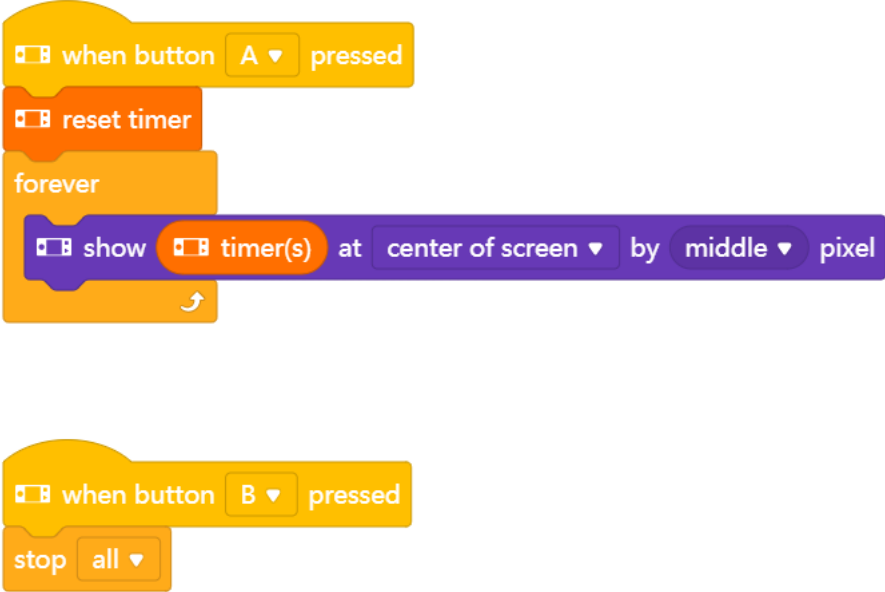
Sensor-Typ	Was macht dieser Sensor?
Schall-Sensor	<p>Schall ist eine mechanische Schwingung, die sich in Wellen ausbreitet. In der Luft sind dies Druck- und Dichteschwankungen. Liegt die Schwingung im hörbaren Bereich (zwischen 16 und 20000 Schwingungen pro Sekunde) und ist sie ausreichend intensiv, können wir sie als Töne oder Klänge hören.</p> <p>Die Stärke des Schalls wird als Schallintensität bezeichnet. Der Sensor im Mikrofon wandelt den Schall in ein elektrisches Signal um, das in Bezug auf Tonhöhe und Leistung ausgewertet werden kann.</p> <p>Ein Schallsensor wird häufig für folgende Dinge verwendet:</p> <ul style="list-style-type: none">• Telefongespräche, Tonaufzeichnung• Sprachgesteuerte Assistenzsysteme, z. B. auf einem Mobiltelefon oder in einem Smart Home <p>Im folgenden Programmierbeispiel wird die Schallintensität der Umgebung auf dem Display des CyberPi angezeigt. Sie können das Programmierbeispiel so erweitern, dass der mBot2 während eines Rundgangs durch das Klassenzimmer ständig die Schallintensität misst und auf dem Display anzeigt.</p>
	Programmierbeispiel
	 <pre>when button A pressed show loudness at center of screen by middle pixel</pre>

Sensor-Typ	Was macht dieser Sensor?
Gyroskop & Beschleunigungsmesser	<p>Ein Gyroskop misst Kippbewegungen, genauer gesagt die Geschwindigkeit von Dreh-/Kippbewegungen. Ein Beschleunigungssensor misst die Änderung der Geschwindigkeit. Diese Sensoren können als mikromechanische Komponenten auf einem elektronischen Bauteil realisiert werden. Beide Sensoren liefern unterschiedliche Informationen über die Position im Raum. Ein Gyroskop wird häufig eingesetzt, um z. B:</p> <ul style="list-style-type: none"> • Schiffe auf hoher See stabil zu halten oder • einen Segway oder ein "Hoverboard" auszubalancieren, damit man mit ihm nicht zu schnell umkippt <p>Beschleunigungssensoren wiederum zeigen an, ob ein Smartphone nach oben oder unten zeigt oder ob ein Fahrzeug einen Unfall hat (heftige, sehr schnelle Geschwindigkeitsänderung) - um dann die Airbags auszulösen.</p> <p>Im untenstehenden Programmierbeispiel wird die Kippbewegung des mBot2 auf dem Display angezeigt. Sie können das Programmierbeispiel so erweitern, dass der mBot2 während einer Runde um das Klassenzimmer ständig die Kippbewegung auf dem Display misst und aufzeichnet.</p> <p>Mehr über diesen Sensor und seine Möglichkeiten erfahren Sie in Lektion 6.</p>
	<p>Programmierbeispiel</p>  <pre> when space key pressed forever if wave up detected? then show UP at x 0 y 0 by middle pixel wait 1 seconds else clear screen </pre>

Sensor-Typ	Was macht dieser Sensor?
Ultraschallsensor	<p>Der Schall ist die Schwingung in Form von Dichte- und Druckänderungen und breitet sich in der Luft mit konstanter Geschwindigkeit in Form von Wellen aus (Schallgeschwindigkeit in Luft ca. 334 m/s). Je höher die Schwingung, desto höher erscheint der Schall - bis zu einer Grenze, ab der der Mensch diesen Schall nicht mehr wahrnehmen kann. Diese sehr hohen Schwingungen werden als Ultraschall bezeichnet. Da die Schallgeschwindigkeit in der Luft bekannt ist, kann man mit Hilfe des Schalls Entfernungen zu Objekten bestimmen; dazu sendet man einen Schall aus und misst die Zeit, bis dieser Schall vom Objekt zurückreflektiert wird. Diese reflektierte Schallwelle wird auch als Echo bezeichnet. Zu diesem Zweck werden in der Regel Ultraschalle eingesetzt.</p> <p>Zum Beispiel wird oft ein Ultraschallsensor verwendet:</p> <ul style="list-style-type: none"> • für ein bildgebendes Verfahren, z. B. in der Schwangerschaft (jedes Pixel ist eine Abstandsmessung) oder • bei der Steuerung von Robotern zur Vermeidung von Kollisionen <p>Im folgenden Programmierbeispiel wird der Ultraschallsensor verwendet, um zu verhindern, dass der mBot2 auf 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. Sie können dieses Programmierbeispiel so erweitern, dass sich der mBot2 willkürlich durch ein Klassenzimmer bewegt, ohne gegen Tische und Stühle zu stoßen. Mehr über diesen Sensor und seine Möglichkeiten erfahren Sie in Lektion 4.</p>
	<p style="text-align: center;">Programmierbeispiel</p>  <pre> when space key pressed forever loop if ultrasonic 2 1 distance to an object (cm) < 10 then turns left 90° until done moves forward at 50 RPM </pre>

Sensor-Typ	Was macht dieser Sensor?
Quad-RGB-Sensor	<p>Der Quad-RGB-Sensor besteht aus vier einzelnen Licht- und Farbsensoren. Sie messen die Intensität des Lichts, das aus den roten, grünen und blauen Bereichen des Lichtspektrums in den Sensor einströmt. Dadurch kann der Sensor Farben von Objekten direkt vor ihm erkennen, wie z. B. Markierungen auf dem Boden, und ermöglicht es dem Roboter auch, einer schwarzen Linie zur Orientierung zu folgen.</p> <p>Ein Quad-RGB-Sensor wird häufig in Anwendungen eingesetzt, wie z. B.:</p> <ul style="list-style-type: none"> • einem Lagerroboter einen Weg durch das Lager zu zeigen (auch unterschiedliche Farben für unterschiedliche Wege), oder • sicherzustellen, dass beim Lackieren genau die richtige Farbe verwendet wird <p>Im folgenden Programmierbeispiel fährt der mBot2 vorwärts, wenn der Roboter die Farbe Weiß sieht. Sie können das Programmierbeispiel so erweitern, dass der mBot2 eine Route durch das Klassenzimmer fährt, anhält oder basierend auf verschiedenen Farben wendet.</p> <p>Mehr über diesen Sensor und seine Möglichkeiten erfahren Sie in Lektion 5.</p>
	<p>Programmierbeispiel</p>  <pre> when space key pressed forever if quad rgb sensor 1 probe (2) R1 detected white ? then moves forward at 50 RPM else stop encoder motor all </pre>



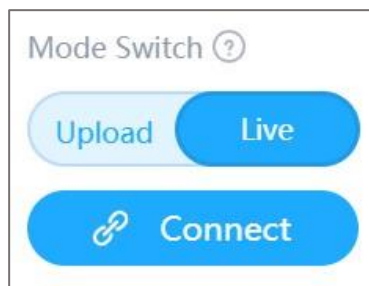
Sensor-Typ	Was macht dieser Sensor?
Timer	<p>Der Timer ist eine Art Stoppuhr, die die Zeit in Sekunden seit dem Einschalten oder Zurücksetzen des CyberPi angibt. Dieser Zähler kann auch mit einem Befehl auf Null gesetzt werden, um Zeitmessungen zu erleichtern.</p> <p>Sie können den Timer z. B. verwenden, um ein Rennen mehrerer mBots untereinander auszutragen und genau zu messen, wie schnell sie im Ziel sind.</p> <p>Das folgende Programmierbeispiel zeigt Ihnen, wie Sie den Timer einstellen und verwenden können. Durch Drücken der Taste 'A' wird der Timer auf Null zurückgesetzt und dann dauerhaft im Display angezeigt. Mit der Taste 'B' können Sie die Anzeige und das gesamte Programm beenden. Versuchen Sie, das Programmierbeispiel so zu erweitern, dass der Timer zu zählen beginnt, wenn sich der mBot2 in Bewegung setzt.</p>
	Programmierbeispiel
	

2. Programmierbeispiele für die Sensoren testen und erweitern

In der obigen Tabelle befindet sich für jeden Sensor ein Programmierbeispiel. Sie werden diese Programmierbeispiele in mBlock nachbilden und testen. Überlegen Sie sich für zwei der

Programmierbeispiele eine Erweiterung. In der obigen Tabelle ist für jedes Programmierbeispiel bereits ein Erweiterungsvorschlag angegeben. Vielleicht kennen Sie eine noch bessere Erweiterung!

Während Sie ein Computerprogramm verfassen, können Sie sofort testen, was Sie erstellen. Dazu verwenden Sie den Live-Modus. Sie wählen den Live-Modus, indem Sie den Mode-Schalter nach rechts schieben. Sehen Sie sich die Abbildung unten an.



3. Die Daten von den Sensoren anzeigen

Beim Testen der Programmierbeispiele haben Sie gesehen, dass jeder Sensor etwas auf dem Display des CyberPi anzeigt. Der Lichtsensor erfasst zum Beispiel die Lichtintensität der Umgebung. Der Timer registriert die Zeit, und der Gyrosensor hält fest, in welche Richtung sich der mBot2 neigt. Alles, was ein Sensor registriert, wird als Daten bezeichnet.

Sie können die Daten von einem Sensor auf dem Display des mBot2 anzeigen. In den Programmierbeispielen haben Sie gesehen, dass Sie die Daten z. B. mit einer Zahl oder einem Text anzeigen können.





Sie können die Daten von den Sensoren auf unterschiedliche Weise auf dem Display des mBot2 anzeigen. Dazu können Sie verschiedene Codeblöcke verwenden. Diese Codeblöcke finden Sie in mBlock unter der Kategorie „Display“.

In der Abbildung unten sehen Sie drei Beispiele, wie Sie Daten auf dem Display des CyberPi anzeigen können. Reproduzieren Sie die Programmierbeispiele und testen Sie sie im Live-Modus. Beobachten Sie, was auf dem Display des mBot2 passiert.

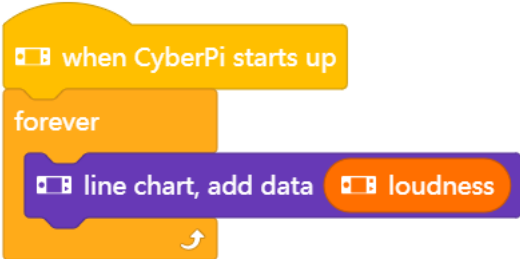
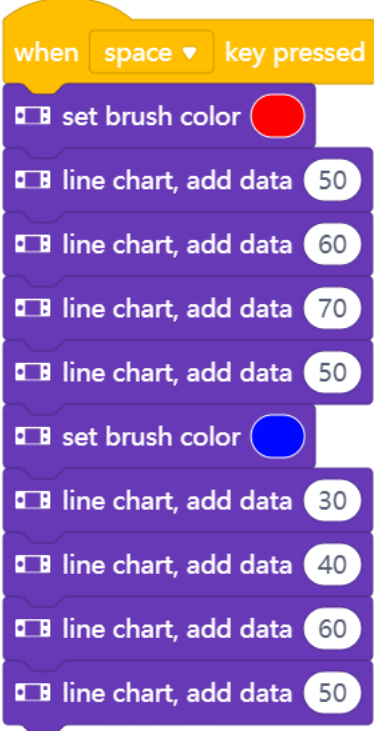
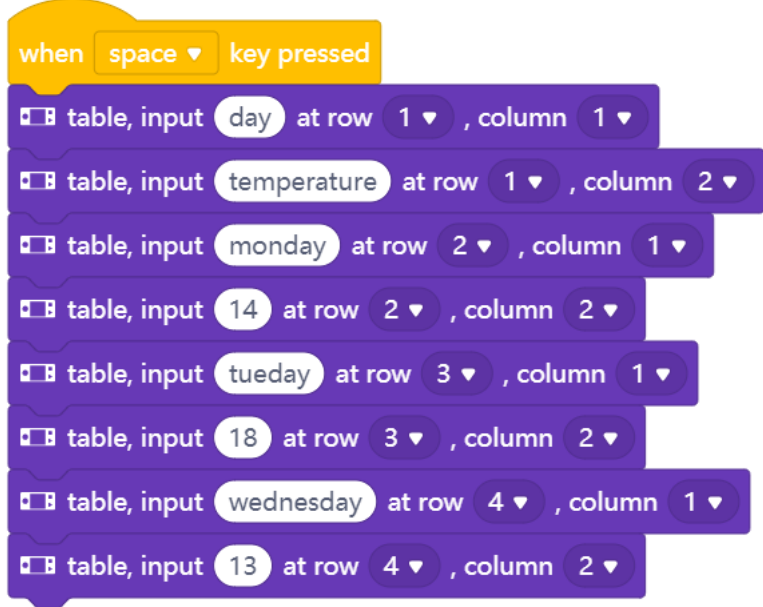
	Programmierbeispiel
Liniendiagramm	
Liniendiagramm (Farbe)	

Tabelle	 <pre> when space key pressed table, input day at row 1, column 1 table, input temperature at row 1, column 2 table, input monday at row 2, column 1 table, input 14 at row 2, column 2 table, input tuesday at row 3, column 1 table, input 18 at row 3, column 2 table, input wednesday at row 4, column 1 table, input 13 at row 4, column 2 </pre>
---------	--

4. Live-Modus und Upload-Modus

Sie müssen das Computerprogramm, das Sie in mBlock erstellt haben, auf den CyberPi übertragen.

Der CyberPi setzt den mBot2 gemäß den Befehlen, die Sie in mBlock geschrieben haben, in Betrieb. Während Sie ein Computerprogramm verfassen, können Sie direkt testen, was Sie machen. Dazu verwenden Sie den Live-Modus. Sie wählen den Live-Modus, indem Sie den Mode-Schalter nach rechts schieben. Sie dürfen den CyberPi nicht vom Computer trennen, während Sie im Live-Modus testen!

Der CyberPi verfügt auch über einen Upload-Modus. Im Upload-Modus wird das Computerprogramm auf den CyberPi übertragen. Das Programm bleibt so lange auf dem CyberPi gespeichert, bis Sie es durch ein anderes Programm ersetzen. Sie können den CyberPi einfach ausstecken.

In der folgenden Grafik sehen Sie die Unterschiede zwischen Live-Modus und Upload-Modus.



Modus	Erklärung zu den Unterschieden
Live-Modus	<ul style="list-style-type: none">• Die Programme laufen auf Ihrem Computer (= Ihrem PC oder Laptop). Sie werden nicht auf dem CyberPi gespeichert.• Der CyberPi muss mit Ihrem Computer verbunden bleiben.• Der mBlock Code-Editor muss immer auf Ihrem Computer geöffnet sein.• Dieser Modus wird für die Programmierung in diesem Stadium verwendet.
Upload-Modus	<ul style="list-style-type: none">• Die Programme laufen auf dem CyberPi statt auf Ihrem Computer und werden auch auf dem CyberPi gespeichert.• Der CyberPi muss nicht mit Ihrem Computer verbunden sein.• Der mBlock Code-Editor muss nicht auf Ihrem Computer geöffnet sein.• Ein Programm, das Sie auf den CyberPi laden, wird auf dem CyberPi gespeichert, bis Sie es durch ein anderes Programm ersetzen.

3. Ausprobieren (25 Min.)

Schritt 3: Ihn Ausprobieren

Sie haben bereits viel über die Sensoren des mBot2 gelernt und wie Sie die Daten der Sensoren auf dem Display des CyberPi anzeigen können. Sie werden nun selbst Daten mit den Sensoren des mBot2 sammeln und diese auf dem Display des CyberPi anzeigen. Sie werden dies tun, während Sie den mBot2 im Klassenzimmer oder in der Schule herumfahren. Welche Daten Sie sammeln und wie Sie sie auf dem Display des CyberPi darstellen, bleibt Ihnen überlassen. Verwenden Sie das Wissen, das Sie in "Schritt 2" dieser Lektion erworben haben. Natürlich können Sie auch selbst viel mit den verschiedenen Programmiermöglichkeiten in mBlock experimentieren.

Bei den Überlegungen zu dieser Aufgabe ist es hilfreich, den folgenden Schritt-für-Schritt-Plan zu verwenden. Haben Sie eine Idee, was Sie machen wollen? Wenn ja, besprechen Sie zunächst mit Ihrem Lehrer, ob dies machbar ist.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none"> • Welche Strecke soll der mBot2 fahren? • Welche Daten soll der mBot2 sammeln?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none"> • Was benötigen Sie zusätzlich zum mBot2?
Schritt 3: Welche Codeblöcke benötigen Sie, um den mBot2-Antrieb herzustellen?	<ul style="list-style-type: none"> • Wie werden Sie den mBot2 zum Fahren bringen? • Welche Daten werden Sie den mBot2 sammeln lassen? • Welche Codeblöcke werden Sie verwenden? • Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML) • Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe.
Schritt 4: Auf welche Weise möchten Sie die Daten der Sensoren auf dem Display anzeigen?	<ul style="list-style-type: none"> • Wie wollen Sie die Daten auf dem Display anzeigen? • Welche Codeblöcke werden Sie verwenden? • Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML) • Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe.



Schritt 5: Test und Implementierung	<ul style="list-style-type: none">• Ist die erste Version fertig? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf.• Arbeiten Sie an den Verbesserungspunkten, bis der mBot2 genau das tut, was Sie sich vorgestellt haben.• Ist es Ihnen gelungen? Filmen Sie das Endergebnis und fragen Sie Ihren Lehrer, ob Sie es auf sozialen Medien mit dem Hashtag #mbot2inclass posten dürfen.
-------------------------------------	---

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Welche Daten hat Ihr mBot2 gesammelt? Ist es gut gelaufen? Konnten Sie alle Daten auf dem Display des CyberPi sehen?

In dieser Lektion haben Sie die verschiedenen Sensoren des mBot2 kennengelernt und wissen, wo Sie sie im Alltag antreffen könnten. Sie wissen, wie man diese Sensoren programmiert und wie man die Daten auf dem Display anzeigt. Sie haben auch den Unterschied zwischen dem Upload- und dem Live-Modus des mBlocks kennengelernt.

Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was hat Ihrer Meinung nach gut funktioniert?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?

Lektion 3: Dem mBot2 zuhören

Thema: STEAM

Dauer: 45 Minuten

Klasse(n): 5. Klasse und aufwärts

Schwierigkeitsgrad: Anfänger

★ Lektionsziele

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Die Codeblöcke zur Steuerung des Lautsprechers und des Mikrofons zu erkennen und zu verwenden
- Ihr eigenes Computerprogramm in mBlock zu erstellen, damit der mBot2 Ton aufnimmt und wiedergibt
- Mehrere Programmieraufgaben nebeneinander in einem Computerprogramm auszuführen

★ Überblick

Roboter können oft sprechen. Nehmen Sie zum Beispiel den Staubsaugerroboter, der Sie mit einer Sprachnachricht darauf hinweist, dass sein Akku leer ist. Oder ein Spielzeugroboter, der ein Lied singt. Der mBot2 kann auch Ton produzieren und aufnehmen. Dafür hat er ein Mikrofon und einen Lautsprecher. Mit dem Mikrofon nimmt der mBot2 Ton auf und der Lautsprecher gibt den aufgenommenen Ton wieder.

Schwerpunkt

Am Ende der Lektion werden die Schüler wissen:

- Wie man den Lautsprecher in der Codierung und Robotik einsetzt
- Wie man Ton aufnimmt und abspielt
- Wie man mehrere Programmieraufgaben nebeneinander ausführt

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle

Lehrplan

Diese Lektion besteht aus vier Schritten und dauert insgesamt 45 Minuten.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none">• Lautsprecher im Alltag• Wie funktioniert ein Lautsprecher?
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none">• Die verschiedenen Codeblöcke des Lautsprechers und des Mikrofons kennenlernen.• Einige Programmierbeispiele des Lautsprechers und des Mikrofons nachbauen, testen und erweitern.
25 Minuten	3. Ausprobieren <ul style="list-style-type: none">• Ihr eigenes Programm für den Roboter schreiben.
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none">• Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zur späteren Diskussion.• Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass• Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Lautsprecher im Alltag
2. Wie funktioniert ein Lautsprecher?

1. Lautsprecher im Alltag

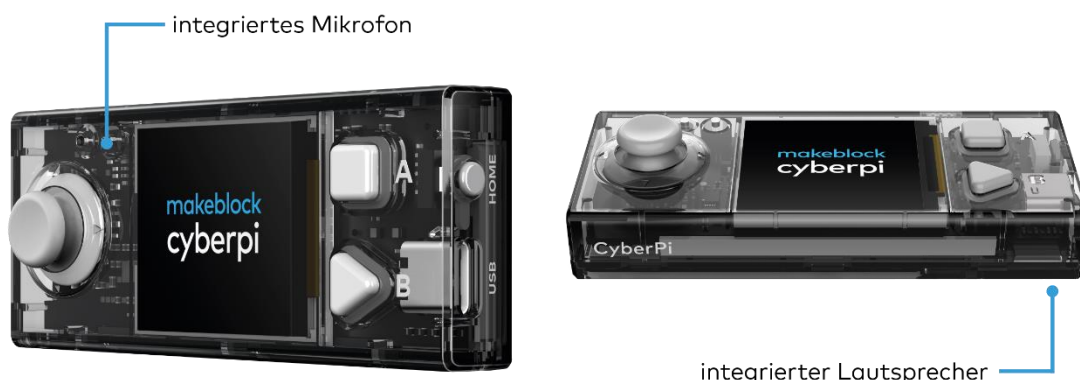
Lautsprecher werden im täglichen Leben überall um Sie herum eingesetzt. Sie finden große Lautsprechersysteme in Theatern, Konzertsälen und Veranstaltungshallen. Diese Lautsprecher werden verwendet, um den Klang für eine große Gruppe von Menschen zu verstärken. Kleinere Lautsprechersysteme findet man in Geräten wie Fernsehern, Musikinstrumenten und Smartphones. Fallen Ihnen und den Mitschülern in Ihrer Gruppe weitere Anwendungen ein?

2. Wie funktioniert ein Lautsprecher?

Der Zweck eines Lautsprechers ist es, Schall zu erzeugen. Zu diesem Zweck wandeln Lautsprecher elektromagnetische Wellen in Schallwellen um. Neben seinem Hauptzweck kann ein Lautsprecher auch Schallwellen wieder in elektromagnetische Wellen umwandeln.

Mikrofone sind Geräte, die das gleiche Prinzip verwenden, aber für den Zweck der Umwandlung von Schallwellen in elektrische Signale optimiert sind.

Auf dem mBot2 befinden sich ein Lautsprecher und ein Mikrofon. Das Mikrofon befindet sich oben links auf dem CyberPi. Der Lautsprecher befindet sich an der Vorderseite des CyberPi. Mit dem Mikrofon können Sie Töne aufnehmen. Diese Töne können später mit dem Lautsprecher wiedergegeben werden.



Durch die Verwendung des Mikrofons und des Lautsprechers in Kombination mit den anderen Sensoren des mBot2 können Sie z. B. Ihren eigenen aufgenommenen Text zu einem bestimmten Zeitpunkt abspielen lassen. Denken Sie zum Beispiel an einen öffentlichen Bus, der Ihnen sagt, an welcher Haltestelle Sie sich gerade befinden. Es gibt unzählige weitere Möglichkeiten, das Mikrofon und den Lautsprecher zu nutzen. Denken Sie gemeinsam mit den Klassenkameraden aus Ihrer Gruppe an weitere Beispiele, bei denen der mBot2 den Lautsprecher und das Mikrofon sinnvoll einsetzen kann.

2. Praxisorientiert (10 Min.)

Schritt 2: Praxisorientiert

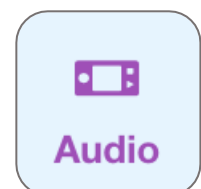
Dieser Schritt besteht aus zwei Teilabschnitten:

1. Die verschiedenen Codeblöcke für den Lautsprecher und das Mikrofon kennenlernen.
2. Einige Programmierbeispiele mit dem Lautsprecher nachbauen, testen und erweitern.

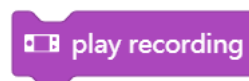
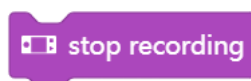
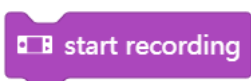
1. Die verschiedenen Codeblöcke des Lautsprechers und des Mikrofons kennenlernen

In mBlock 5 gibt es mehrere Codeblöcke, mit denen Sie den Lautsprecher und das Mikrofon programmieren können. Sie finden diese Codeblöcke in der Kategorie "Audio" des Blockbereichs in mBlock. Diese Codeblöcke sind lilafarbig.

In der folgenden Tabelle finden Sie einige dieser Codeblöcke zum Programmieren des Lautsprechers und des Mikrofons.



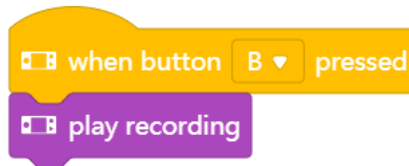
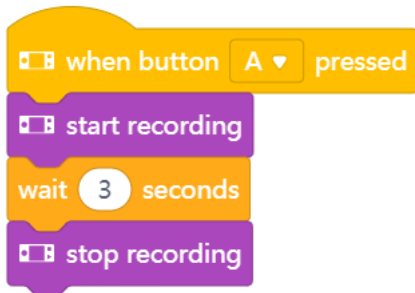
Codeblock:



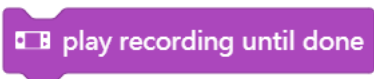
Bei der Aufnahme und Wiedergabe eines Tons verwenden Sie eigentlich immer drei Codeblöcke. Mit dem Codebaustein "Aufnahme starten" lassen Sie CyberPi die Aufnahme eines Tons starten. Das elektrische Signal vom Mikrofon wird im Speicher des CyberPi abgespeichert, und die Aufnahme stoppt automatisch nach 10 Sekunden. Sie können auch angeben, wann die Aufnahme beendet werden soll. Dies geschieht mit dem Codeblock "Aufnahme stoppen". Der aufgenommene Ton wird im Speicher des CyberPi zwischengespeichert.

Schalten Sie den mBot2 aus? Dann ist der aufgenommene Ton verloren. Das Abspielen des aufgenommenen Tons wird mit dem Codeblock "Aufnahme abspielen" gesteuert.

Im folgenden Programmierbeispiel startet der CyberPi die Aufnahme, wenn Sie die Taste A drücken. Nach 3 Sekunden stoppt die Aufnahme. Wenn Sie die Taste B drücken, hören Sie, was aufgenommen wurde. Durch die Verwendung von zwei Ereignisblöcken können zwei Programmieraufgaben unabhängig voneinander ablaufen, ohne dass eine Hauptschleife erforderlich ist.

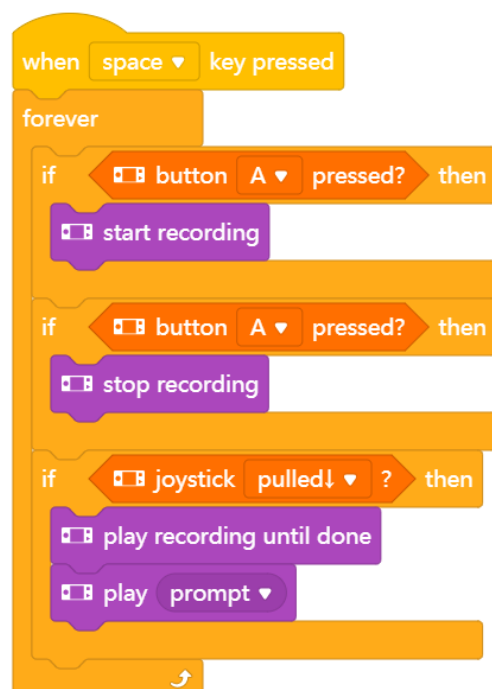


Codeblock:



Sie haben gerade gelernt, wie man einen aufgenommenen Ton wiedergibt. Mit diesem Codebaustein sorgen Sie dafür, dass der vom Mikrofon aufgenommene Ton vollständig abgespielt wird. Das bedeutet, dass die Ausführung weiterer Blöcke blockiert ist, bis dieser beendet ist. Nachdem die Aufnahme und das Abspielen des Tons beendet ist, können Sie den mBot2 etwas anderes tun lassen. Zum Beispiel ein bestimmtes Geräusch machen, damit Sie wissen, dass die Aufnahme beendet ist.

Im Programmierbeispiel auf der rechten Seite startet der CyberPi die Aufnahme, wenn Sie den Knopf A drücken. Wenn Sie den Joystick nach unten bewegen, werden Sie hören, was der mBot2 aufgenommen hat. Nachdem die gesamte Aufnahme beendet ist, hören Sie einen "Prompt"-Ton. Die Codeblöcke in diesem Beispiel sind in einer sogenannten Schleifenstruktur angeordnet. Dadurch wird sichergestellt, dass die Eingaben von den Tasten und dem Joystick kontinuierlich überwacht werden.



2. Einige Programmierbeispiele mit dem Lautsprecher nachbauen, testen und erweitern

In der obigen Tabelle ist jedem Codeblock des Lautsprechers ein Programmierbeispiel zugeordnet. Sie werden diese Programmierbeispiele in mBlock nachbilden und testen. Überlegen Sie sich eine Erweiterung zu einem Programmierbeispiel. Testen Sie die Programmierbeispiele und Ihre eigene Erweiterung im Live-Modus.

3. Ausprobieren (25 Min.)

Schritt 3: Ihn Ausprobieren

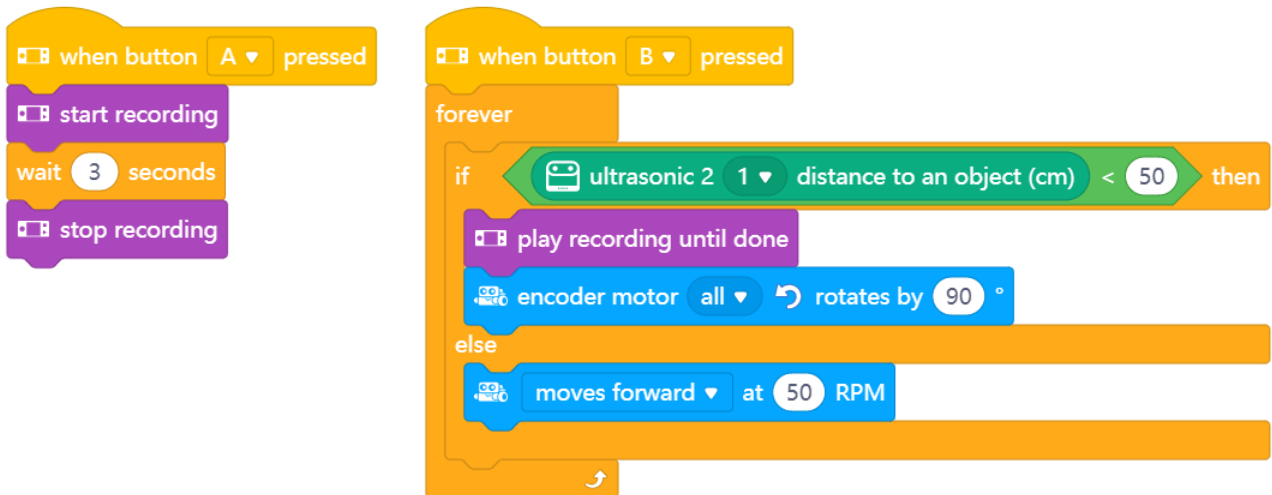
Sie haben bereits viel über den Lautsprecher und das Mikrofon des mBot2 gelernt. Sie werden nun selbstständig an ihnen arbeiten. Sie werden auch den Ultraschallsensor verwenden. Dieser Sensor kann ein Objekt aus der Ferne erfassen. Sie werden in Lektion 4 mehr über den Ultraschallsensor erfahren.

In dieser Aufgabe werden Sie den mBot2 eine zufallsgenerierte Runde durch das Klassenzimmer fahren lassen. Zufallsbedingt bedeutet, dass Sie keine Route vorgeben. Wenn der mBot2 in die Nähe eines Hindernisses kommt, z. B. eines Stuhls, eines Tisches oder eines Rucksacks, werden Sie ein Geräusch hören und der mBot2 wird sich umwenden. Sie selbst werden das Geräusch mit dem Mikrofon aufnehmen. Ein solches Geräusch könnte z. B. ein lauter Piepton oder das Geräusch eines langsamer werdenden Autos sein. Sie können auch einen Text aufnehmen oder den mBot2 ein Lied abspielen lassen.

Es ist ziemlich schwierig, sich selbst ein Computerprogramm für diese Aufgabe auszudenken! Zum Glück gibt es Hilfe. Auf der nächsten Seite sehen Sie zwei Beispiele für ein Computerprogramm. Sie können diese Computerprogramme selbst erweitern und anpassen. Seien Sie kreativ!

Beispiel 1

Der mBot2 macht ein Geräusch, wenn er sich einem Hindernis nähert und sich umwendet. Sie haben das Geräusch vorhin selbst aufgenommen.



Beispiel 2

Der mBot2 fährt herum und spielt einen von Ihnen aufgenommenen Ton ab, wenn das Umgebungslicht schwach ist. Wenn sich der mBot2 einem Hindernis nähert, gibt er einen Warnton ab und wendet sich.





Verwenden Sie das Wissen, das Sie in "Schritt 2" dieser Lektion erworben haben. Natürlich können Sie auch selbst viel mit den verschiedenen Programmiermöglichkeiten in mBlock experimentieren.

Bei den Überlegungen zu dieser Aufgabe ist es hilfreich, den folgenden Schritt-für-Schritt-Plan zu verwenden. Haben Sie eine Idee, was Sie machen wollen? Wenn ja, besprechen Sie zunächst mit Ihrem Lehrer, ob dies machbar ist.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none">• Wann soll der mBot2 wenden?• Welches Geräusch sollte der mBot2 machen, wenn er sich einem Hindernis nähert?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none">• Was benötigen Sie zusätzlich zum mBot2?
Schritt 3: Welche Codeblöcke benötigen Sie, um den mBot2-Antrieb herzustellen?	<ul style="list-style-type: none">• Wie werden Sie den mBot2-Antrieb herstellen?• Welche Codeblöcke werden Sie verwenden?• Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML)• Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe.
Schritt 4: Auf welche Weise möchten Sie die Daten der Sensoren auf dem Display anzeigen?	<ul style="list-style-type: none">• Welches Geräusch soll der mBot2 machen?• Welche Codeblöcke werden Sie verwenden?• Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML)• Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe

Schritt 5: Test und Implementierung	<ul style="list-style-type: none"> • Ist die erste Version fertig? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf. • Arbeiten Sie an den Verbesserungspunkten, bis Ihrer mBot2 genau das tut, was Sie sich vorgestellt haben. • Ist es Ihnen gelungen? Filmen Sie das Endergebnis und fragen Sie Ihren Lehrer, ob Sie es auf sozialen Medien mit dem Hashtag #mBot2inclass posten dürfen
-------------------------------------	--

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Wie ist diese Aufgabe gelaufen? Hat der mBot2 sofort das gemacht, was Sie sich vorgestellt haben?

In dieser Lektion haben Sie gelernt, wie Lautsprecher funktionieren und wo sie im Alltag eingesetzt werden. Sie wissen, wie Sie den CyberPi und den mBot so programmieren können, dass er über das eingebaute Mikrofon aufnimmt und über die Lautsprecher wiedergibt. Sie wissen auch, wie Sie mehrere Programme unabhängig voneinander laufen lassen können. Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was ist Ihnen Ihrer Meinung nach gut gelungen?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?

Lektion 4: Sehen mit Ultraschall

Thema: STEAM

Klasse(n): 5. Klasse und aufwärts

Dauer: 45 Minuten

Schwierigkeitsgrad: Anfänger

★ Lektionsziele

Am Ende der Lektion werden die Schüler wissen:

- Was Ultraschall ist und wie ein auf Ultraschall basierender Sensor funktioniert
- Welche Anwendungsmöglichkeiten es im Alltag und in der Robotik für diese Sensoren gibt

★ Überblick

Autonome Roboter müssen sich beim unbeaufsichtigten Fahren ihrer Umgebung bewusst sein, um eventuelle Kollisionen mit Hindernissen oder Personen zu vermeiden. Dies gilt auch für den mBot2. Der mBot2 hat dafür einen speziellen Sensor an der Vorderseite. Er wird als Ultraschallsensor bezeichnet und ermöglicht es dem Roboter, Objekte in seiner Fahrbahn anhand von Tönen zu erkennen, die Menschen nicht hören können - daher der Name (Ultraschall).

🔗 Schwerpunkt

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Einen Ultraschallsensor für die Entfernungs- und Hinderniserkennung zu verwenden.
- Den mBot2 dazu bringen, auf Hindernisse zu reagieren und ihnen beim Fahren auszuweichen

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle
- Kleine Hindernisse (Mindestgröße ist 10x10x10 cm)
- Ein Arbeitsbereich von mindestens 1 m², in dem der mBot2 fahren kann

Lehrplan

Diese Lektion besteht aus vier Schritten und dauert insgesamt 45 Minuten.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none"> • Ultraschallsensoren im Alltag • Wie funktioniert ein Ultraschallsensor?
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none"> • Die verschiedenen Codeblöcke des Ultraschallsensors kennenlernen • Einige Programmierbeispiele der Ultraschallsensoren nachbilden und testen.
25 Minuten	3. Ausprobieren <ul style="list-style-type: none"> • Ihr eigenes Programm für den Roboter schreiben.
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none"> • Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zur späteren Diskussion. • Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass • Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Ultraschallsensoren im Alltag
2. Wie funktioniert ein Ultraschallsensor?

1. Ultraschallsensoren im Alltag

Ultraschall ist ein Ton mit einer sehr hohen Frequenz. So hoch, dass Menschen ihn nicht hören können. Mit speziellen Mikrofonen kann dieser Ton aufgezeichnet und auf einem Computer dargestellt werden: Er erscheint wie Schwingungen in Form einer Welle. Da Ultraschall eine sehr hohe Frequenz hat, zeigt die Wellenform sehr dicht beieinander liegende Gipfel und Tiefpunkte. Ultraschall kann eine nützliche Sache sein. Zum Beispiel kann man die Schwingungen des Ultraschalls nutzen, um Objekte zu reinigen, oder man kann ihn auch nutzen, um Objekte zu erkennen.

Delfine und Fledermäuse nutzen Ultraschall auf diese Weise. Sie senden einen Ultraschallton aus, wie einen kurzen Schrei. Wenn dieser Ton von etwas abprallt, fangen sie mit ihren Ohren die Schwingungen (Echo) auf, die zurückkommen. So wissen sie zum Beispiel, wo andere Tiere sind oder ob ein Hindernis im Weg ist. Das Radar, das wir heute haben, nutzt das gleiche Prinzip, aber mit Radiowellen anstelle von Ton.

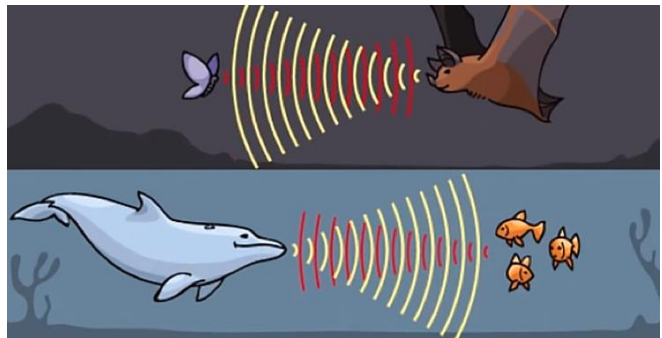


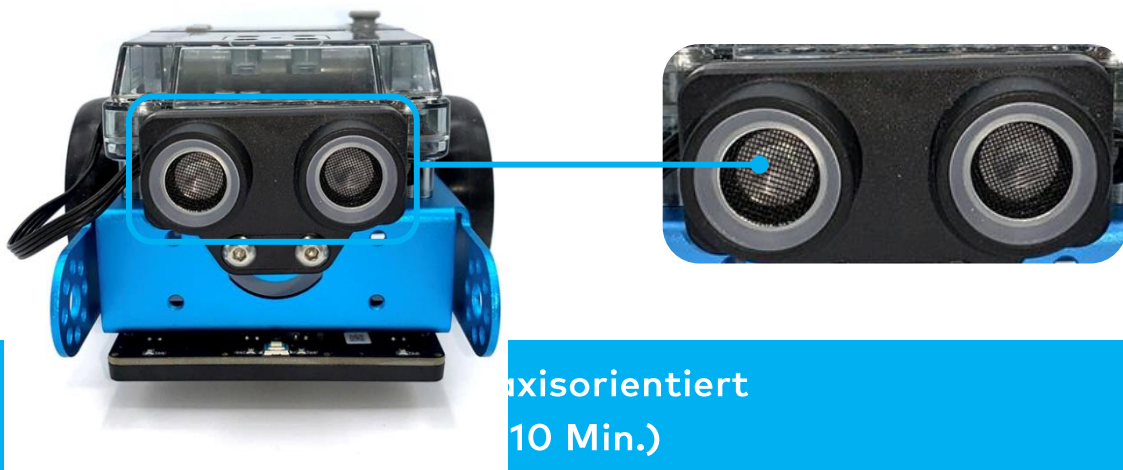
Abbildung 2 Beispiel mit einer Fledermaus und einem Delfin

2. Wie funktioniert ein Ultraschallsensor?

Der mBot2 verfügt über einen Ultraschallsender und einen Empfänger. Diese befinden sich an der Vorderseite des mBot2 in den kleinen Zylindern, die als "Augen" angesehen werden können. Dieser Sensor sendet kurze Impulse von Ultraschalltönen aus und lauscht auf ein Echo. Kommt der mBot2 in die Nähe eines Hindernisses? Dann wird der Ultraschallschall zum mBot2

zurückreflektiert. Anhand der Zeit, die der Ton benötigt, um zum Sensor zurückzukehren, berechnet der mBot2 intern die Entfernung zum Hindernis.

Anhand dieser Daten kann entschieden werden, wie der Roboter reagieren soll: Sie sagen es dem Roboter, indem Sie ihn programmieren. Eine solche Aktion könnte z. B. sein, dass der mBot2 anhalten oder eine Wendung machen soll.



Schritt 2: Praxisorientiert

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Die verschiedenen Codeblöcke des Ultraschallsensors kennenlernen
2. Einige Programmierbeispiele der Ultraschallsensoren nachbilden und testen.

1. Die verschiedenen Codeblöcke des Ultraschallsensors kennenlernen

In mBlock gibt es mehrere Codeblöcke für den Ultraschallsensor, die Sie in Ihren Programmen verwenden können. Dazu müssen Sie die Erweiterung "Ultraschallsensor 2" aus der Erweiterungsbibliothek hinzufügen. Dann finden Sie die Codeblöcke in der Kategorie "Ultraschallsensor" des Blockfeldes in mBlock. Diese Codeblöcke sind grün.



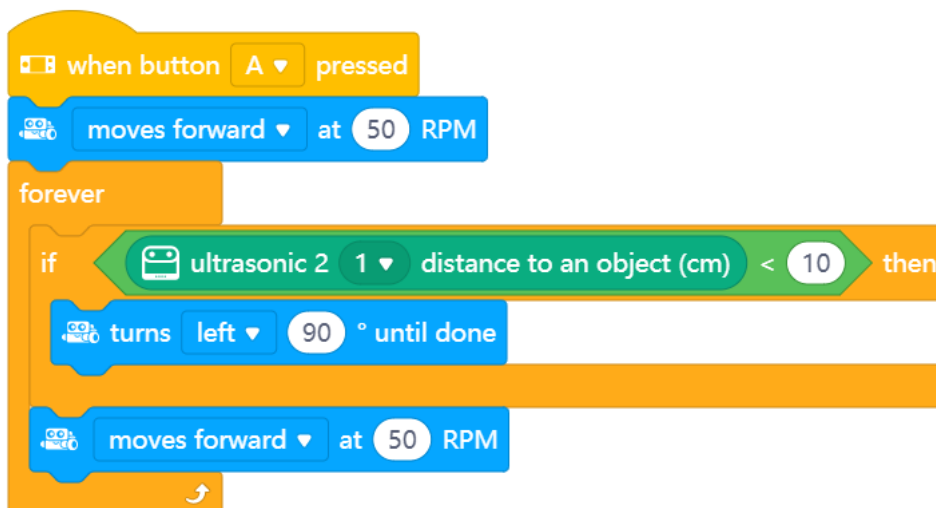
In der folgenden Tabelle finden Sie einige Beispiele für die Codeblöcke zur Programmierung des Ultraschallsensors.

Codeblock:

 ultrasonic 2 1 ▼ distance to an object (cm)

Mit diesem Codeblock messen Sie den Abstand zwischen dem Sensor und einem Hindernis. Der Erfassungsbereich liegt zwischen 3 und 300 cm. Sie können den Wert des Abstands verwenden, damit der Roboter eine bestimmte Aktion ausführt. Sie können z. B. verhindern, dass der mBot2 mit Hindernissen kollidiert.

Im folgenden Programmierbeispiel fährt der mBot2 vorwärts. Wenn der mBot2 weniger als 10 cm von einem Hindernis entfernt ist, macht der Roboter eine 90-Grad-Drehung nach links. Dann fährt der mBot2 wieder vorwärts.

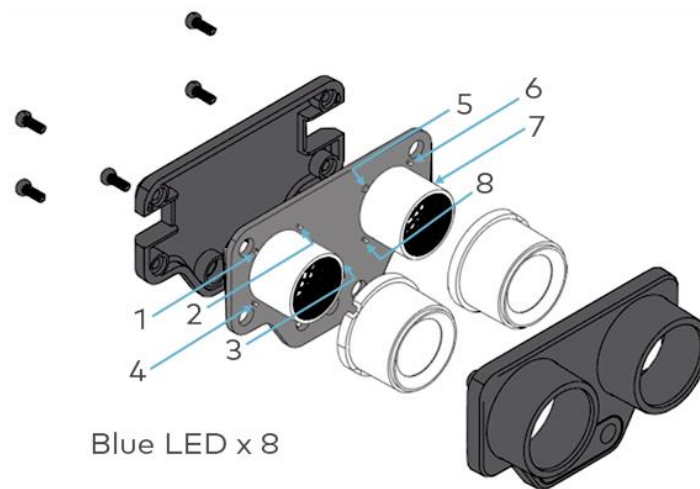


Codeblock:

 ultrasonic 2 1 ▼ increases ambient light all ▼ brightness by 20 %

Der Ultraschallsensor des mBot2 verfügt über acht blaue LED-Leuchten. Sie können diese Leuchten verwenden, um den mBot2 z. B. eine bestimmte "Emotion" zeigen zu lassen, oder als allgemeine visuelle nonverbale Kommunikation. Lassen Sie die LED-Lichter sehr hell leuchten, wenn der mBot2 glücklich ist und weniger hell, wenn er niedergeschlagen ist. Es

gibt auch einen speziellen Codeblock, um vordefinierte "Emotionen" anzuzeigen. In der Abbildung unten können Sie sehen, wo genau sich die LED-Leuchten befinden.



In mBlock gibt es mehrere Codeblöcke, mit denen Sie die Helligkeit der LED-Leuchten programmieren können: Sie können entweder die Helligkeit der LEDs einstellen, indem Sie die Helligkeit um einen bestimmten Prozentsatz verringern oder erhöhen. Oder Sie können die Helligkeit direkt auf einen bestimmten Prozentsatz einstellen. Der Bereich liegt zwischen 0 % und 100 %. Beides kann entweder für eine einzelne LED oder für alle gleichzeitig erfolgen. Im folgenden Programmierbeispiel wird die Helligkeit der LED-Leuchte 1 um 50 % erhöht.



2. Einige Programmierbeispiele der Ultraschallsensoren nachbilden und testen

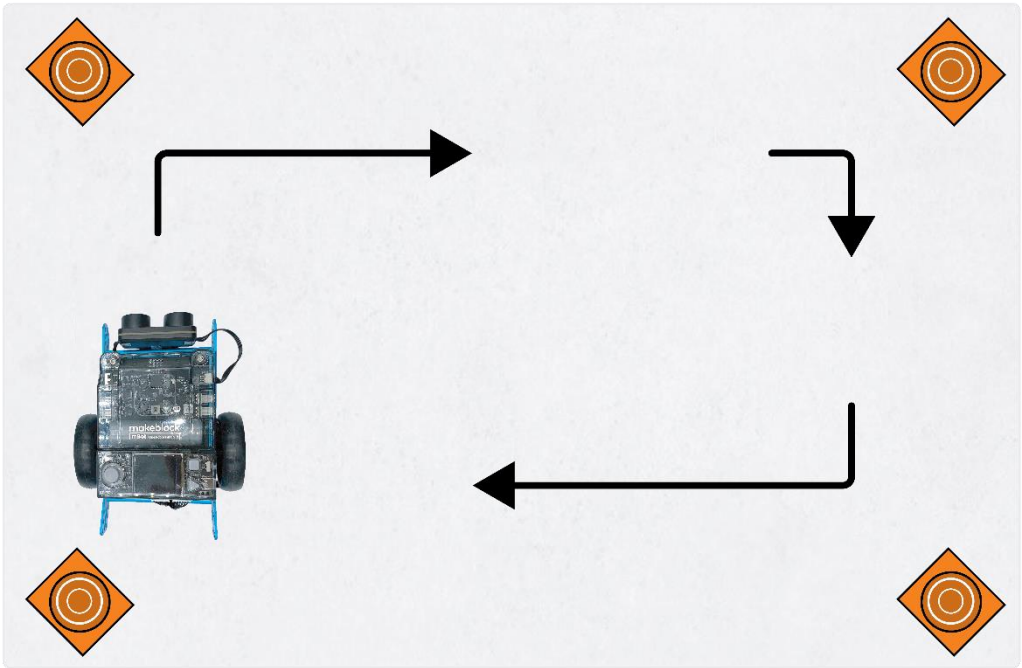
In der obigen Tabelle ist jedem Codeblock des Ultraschallsensors ein Programmierbeispiel zugeordnet. Sie werden diese Programmierbeispiele in mBlock nachbilden und testen. Lassen Sie sich für ein Programmierbeispiel eigene Ideen einfallen, wie Sie die Anwendung erweitern können.

3. Ausprobieren (25 Min.)



Schritt 3: Ihn Ausprobieren

Sie haben bereits viel über den Ultraschallsensor des mBot2 gelernt. Sie werden nun Ihr eigenes Computerprogramm in mBlock erstellen und dabei den Ultraschallsensor verwenden. In dieser Aufgabe werden Sie den mBot2 in unendlichen "Runden" fahren lassen. Dazu platzieren Sie an jeder Ecke ein Objekt. Sie programmieren den mBot2 so, dass er an jedem Objekt eine Wende macht und schließlich zum Startpunkt zurückkehrt. Sehen Sie sich die Abbildung unten an.



Verwenden Sie das Wissen, das Sie in "Schritt 2" dieser Lektion erworben haben. Natürlich können Sie auch mit den verschiedenen Programmiermöglichkeiten in mBlock experimentieren. Bei der Erarbeitung dieser Aufgabe ist es hilfreich, den folgenden Schritt-für-Schritt-Plan zu verwenden. Haben Sie eine Idee, was Sie machen wollen? Wenn ja, besprechen Sie zunächst mit Ihrem Lehrer, ob dies machbar ist.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none">• In welche Richtung soll sich der mBot2 bewegen?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none">• Was benötigen Sie zusätzlich zum mBot2?

<p>Schritt 3: Welche Codeblöcke benötigen Sie, damit sich der mBot2 bewegt?</p>	<ul style="list-style-type: none"> • Wie können Sie den mBot2 an jeder Ecke die Richtung ändern lassen? • Welche Codeblöcke benötigen Sie dafür? • Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML) • Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe
<p>Schritt 4: Test und Implementierung</p>	<ul style="list-style-type: none"> • Ist die erste Version fertig? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf • Arbeiten Sie an den Verbesserungspunkten, bis Ihr mBot2 fehlerfrei über die Strecke läuft • Ist es Ihnen gelungen? Filmen Sie das Endergebnis und fragen Sie Ihren Lehrer, ob Sie es auf sozialen Medien mit dem Hashtag #mBot2inclass posten dürfen


4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Haben Sie es geschafft, den mBot2 dazu zu bringen, eine Runde zu fahren?

In dieser Lektion haben Sie etwas über Ultraschallsensoren gelernt und wo Sie ihnen im Alltag begegnen können. Sie wissen, wie Sie den Ultraschallsensor des mBot2 programmieren können, um z. B. zu verhindern, dass der mBot2 gegen Hindernisse fährt. Sie wissen auch, wie Sie die LED-Leuchten im Ultraschallsensor verwenden können, damit der mBot2 eine bestimmte Emotion zeigt.

Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- 
- Was ist Ihnen Ihrer Meinung nach gut gelungen?
 - Was könnte noch besser sein?
 - Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
 - Worüber hätten Sie gerne mehr Erklärungen?
 - Wer könnte Ihnen dabei helfen?

Lektion 5: Stadtbesichtigung

Thema: STEAM

Dauer: 45 Minuten

Klasse(n): 5. Klasse und aufwärts

Schwierigkeitsgrad: Anfänger

★ Lektionsziele

Am Ende der Lektion werden die Schüler wissen:

- Wie Farbsensoren funktionieren
- Welche Anwendungsmöglichkeiten sie im Alltag und in der Robotik haben

★ Überblick

Stellen Sie sich vor, Sie sind in einer fremden Stadt und machen eine Tour in einem Reisebus vorbei an den schönsten Sehenswürdigkeiten. Vielleicht haben Sie selbst schon einmal eine solche Reise gemacht, und wenn nicht, kennen Sie wahrscheinlich die Bilder aus dem Fernsehen und dem Internet. In dieser Lektion wird der mBot2 in einen solchen Reisebus verwandelt.

Da es für einen Fahrer sehr langweilig wäre, jeden Tag die gleiche Strecke zu fahren (und jede Sehenswürdigkeit immer wieder zu beschreiben), werden wir diesen Reisebus programmieren. Auf diese Weise können die Leute eine schöne Fahrt genießen, ohne dass der Fahrer sich bei sich wiederholenden Aufgaben langweilt.

🔗 Schwerpunkt

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Den mBot2 einer Linie folgen zu lassen
- Den mBot2 Aktionen basierend auf Farben ausführen zu lassen

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle

Lehrplan

Diese Lektion besteht aus vier Schritten und nimmt insgesamt 45 Minuten in Anspruch.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none">• Farbsensoren im Alltag• Wie funktionieren Farbsensoren? Was ist das Besondere am Farbsensor des mBot2?
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none">• Die verschiedenen Codeblöcke des Quad-RGB-Sensors kennenlernen• Programmierbeispiele des Quad-RGB-Sensors nachbilden und testen• Wie kann der mBot2 einer Linie folgen?
25 Minuten	3. Ausprobieren <ul style="list-style-type: none">• Ihr eigenes Programm für den Roboter schreiben
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none">• Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zum Zwecke späterer Diskussion.• Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass• Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Farbsensoren im Alltag
2. Wie funktionieren Farbsensoren? Was ist das Besondere am Farbsensor des mBot2?

1. Farbsensoren im Alltag

Farbsensoren können Lichtintensität und Farben erkennen. Diese Art von Sensor begegnet Ihnen im Alltag an vielen verschiedenen Stellen.

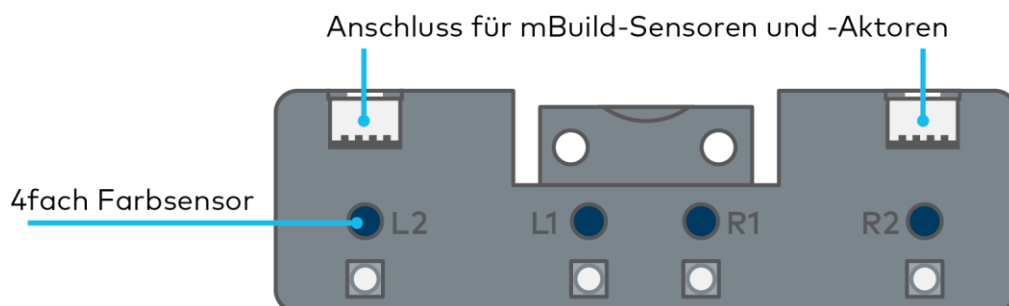
So kann z. B. ein Lagerroboter anhand von unterschiedlichen Farben auf dem Boden den richtigen Weg durch ein Lagerhaus finden. Diese Art von Sensor wird auch eingesetzt, wenn eine genaue Farbabstimmung erforderlich ist, z. B. bei der Fahrzeuglackierung. Ein Messgerät mit einem Farbsensor, das Farbmessgerät, kann prüfen, ob beim Lackieren eines Fahrzeugs die richtige Farbe verwendet wird. Fallen Ihnen und den Mitschülern aus Ihrer Gruppe noch weitere Beispiele ein?

2. Wie funktionieren Farbsensoren? Was ist das Besondere am Farbsensor des mBot2?

Ein Farbsensor besteht aus drei verschiedenen (internen) Sensoren, die das von einem Objekt reflektierte Licht anhand der Intensität der Farbwerte Rot, Grün und Blau messen.

Farbsensoren werden daher auch "RGB-Sensoren" genannt. Jede Farbe wird vom Sensor in drei Werte dekodiert - eine bestimmte "Mischung" dieser Werte stellt also eine Farbe dar.

Der mBot2 hat vier dieser RGB-Sensoren in einem einzigen Sensor integriert. Sehen Sie sich die Abbildung unten an.



Dies ist der Quad-RGB-Sensor, der sich auf dem mBot2 befindet. Die RGB-Sensoren heißen L1, L2, R1 und R2 (L für die linke Seite, R für die rechte Seite). Sie erkennen automatisch die RGB-Werte der reflektierten Farbe und vergleichen die Mischung der Werte intern mit verschiedenen Voreinstellungen für Farben. Dies erleichtert die Codierung erheblich, da der Sensor dann die Farbe melden kann und Sie die RGB-Codes nicht selbst überprüfen müssen. Der Sensor kann 6 verschiedene Farben plus Schwarz und Weiß erkennen. Anstelle von Farbe kann er auch Grautöne oder Linien und Übergänge melden.

Der Quad-RGB-Sensor ermöglicht es dem mBot2, einer Linie auf dem Boden zu folgen und auf verschiedene Farbmarkierungen zu reagieren. Der Quad-RGB-Sensor befindet sich an der Vorderseite des mBot2. Auf der Unterseite sehen Sie die vier Sensoren. Oben finden Sie die Namen der einzelnen Sensoren, so dass Sie die Sensoren gezielt identifizieren und programmieren können. Sehen Sie sich die Abbildungen unten an.



Quad-RGB-Sensor, Vorderansicht

Ansicht von unten

Die kleine Taste auf der Oberseite des Sensors dient zur Kalibrierung. Wenn Sie darauf doppelklicken, beginnen die LEDs zu blinken und Sie können den mBot2 mit dem Sensor über die zu verfolgende Linie "wischen". Dadurch wird der Sensor kalibriert, um die Linie vom Hintergrund zu unterscheiden. Wenn Sie eine schwarze Linie auf einer weißen Oberfläche verwenden, ist dies normalerweise nicht erforderlich.

Die mit dem mBot2 mitgelieferte Karte hat Farben, die in der Strecke selbst markiert sind, um auf die Farbcodes zu reagieren. Dies kann dazu führen, dass der Sensor eine helle Farbe (wie Gelb) für den Hintergrund und nicht für die Strecke missversteht. Kalibrieren Sie also bitte als erste Übung den Sensor auf den gelben Farbcode, er wird dann automatisch diese und alle "dunkleren" Farben zuverlässig als die Strecke erkennen.

Wenn Sie den mBot2 programmieren, werden Sie einen oder mehrere dieser vier Sensoren verwenden, um entlang der Strecke nach bestimmten Farben auf dem Boden zu suchen. Sie können den mBot2 einen Befehl ausführen lassen, wenn eine Farbe entdeckt wird. Zum Beispiel "Stopp", wenn er rot ist und "nach links fahren", wenn er grün ist.

2. Praxisorientiert (10 Min.)

Schritt 2: Praxisorientiert

Dieser Schritt besteht aus drei Teilabschnitten:

1. Die verschiedenen Codeblöcke des Quad-RGB-Sensors kennenlernen
2. Einige Programmierbeispiele des Quad-RGB-Sensors nachbilden und testen
3. Wie kann der mBot2 einer Linie folgen?

1. Die verschiedenen Codeblöcke des Quad-RGB-Sensors kennenlernen

In mBlock gibt es mehrere Codeblöcke, mit denen Sie den Quad-
RGB-Sensor programmieren können. Sie finden diese Codeblöcke in
der Kategorie "Quad RGB" des Blockbereichs in mBlock. Diese
Codeblöcke sind grün.



In der folgenden Tabelle finden Sie einige der Codeblöcke, die Sie in Ihrem Programm mit dem Quad-RGB-Sensor verwenden können. Die ersten vier Codeblöcke sind für die (einfache) Linienidentifikation. Sie können für einfache und erweiterte Linienverfolgung verwendet werden, aber auch für die Erkennung von Kreuzungen und scharfen (90°) Kurven. Die Kalibrierung des Sensors auf unterschiedliche Farbspuren ist im obigen Abschnitt kurz erwähnt. Die blauen LEDs auf der Oberseite des Sensors zeigen die Linien-/Hintergrunderkennung an: Wenn sie aus sind, erkennen sie eine dunkle Farbe (Linie), wenn sie blau leuchten, zeigen sie eine helle Farbe (Hintergrund) an. Der Status der LED zeigt also die Reflektivität des Bodens oder der Karte an.









Codeblöcke:



Diese Codeblöcke berücksichtigen nur die beiden inneren Sensoren L1 und R1 zur
Linienerkennung. Diese Blöcke sind für die einfache Linienverfolgung gedacht.

Verwenden Sie den ersten Block in bedingten Anweisungen. Verwenden Sie den zweiten Block,
wenn Sie den Wert auf dem Bildschirm anzeigen oder in einer Variablen speichern möchten.

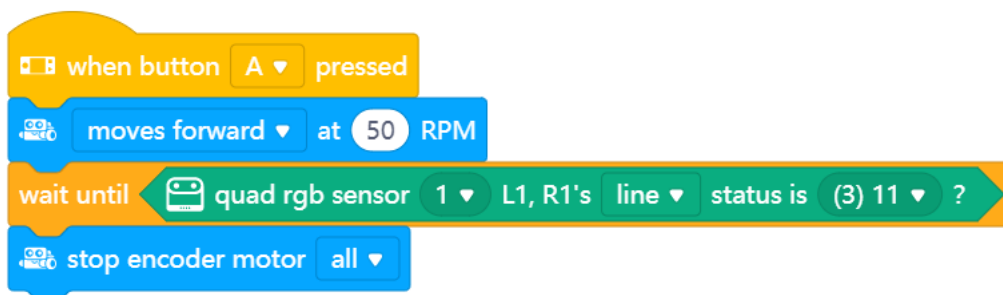
Wenn sich der Sensor über einer schwarzen Linie befindet, sollten beide inneren LEDs aus sein. Da jeder Sensor (L1, R1) mit diesem Block entweder die Linie oder den Hintergrund erkennen kann, gibt es 4 mögliche Kombinationen: sie werden durch Zahlen von 0-3 dargestellt (zur Information zeigt der Block auch das binäre Muster):

L1	R1	Bedeutung	Status (Dezimal)
		Sensor auf weißem Hintergrund, beide LEDs sind an	0
		R1 erkennt eine schwarze Linie	1
		L1 erkennt eine schwarze Linie	2
		Beide inneren Sensoren erkennen die schwarze Linie	3

Vergessen Sie nicht, dass die LEDs den Hintergrundstatus anzeigen (an oder logisch "wahr" = Hintergrund erkannt), während dieser Codeblock die Linienfarbe erkennt (logisch "wahr" = Linie erkannt).

Wenn Sie mehr als einen Quad-RGB-Sensor verwenden möchten, können Sie dem Programm mit der ersten Zahl in allen folgenden Codeblöcken mitteilen, welcher Sensor verwendet werden soll. Sehen Sie sich die Abbildung unten an. Nummer 1 gibt den ersten angeschlossenen Sensor an, Nummer 2 den zweiten und so weiter.

Das folgende Beispiel wird die Linienerkennungsfunktion dieses Blocks verwenden und den Roboter stoppen, sobald er an einer schwarzen Linie ankommt. Dieses Beispiel wird später auf Linienverfolgung umgestellt.



Da das Reagieren während der Fahrt zeitkritisch sein kann, ist es wichtig, den Upload-Modus zu verwenden. Sie können es selbst ausprobieren und den Unterschied beobachten, wenn Sie den Modus wechseln.

Codeblöcke:



Verwenden Sie den ersten Block in bedingten Anweisungen. Verwenden Sie den zweiten Block, wenn Sie den Wert auf dem Bildschirm anzeigen oder in einer Variablen speichern möchten.

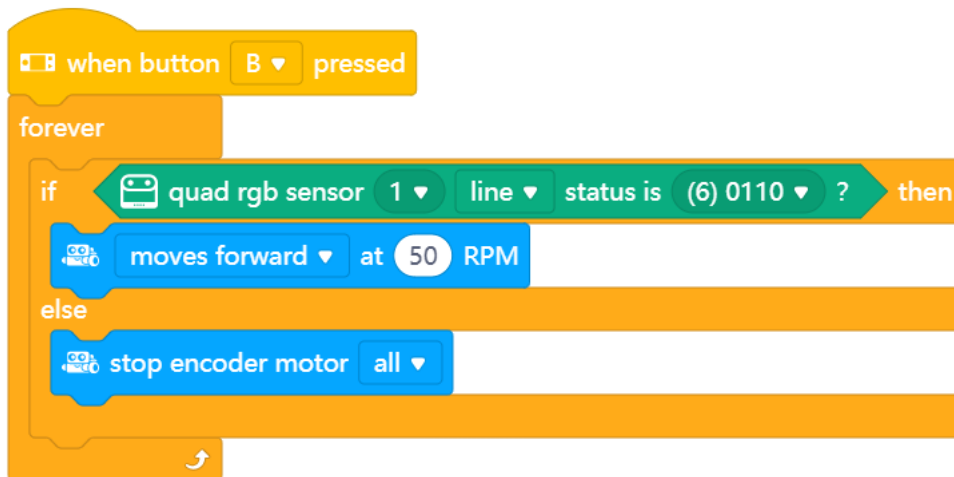
Diese Blöcke folgen genau der gleichen Logik wie die vorherigen, außer dass sie einen größeren Erfassungsbereich haben und Kreuzungen identifizieren können, während sie gleichzeitig eine Linienverfolgung durchführen. Vergessen Sie nicht, dass die Blöcke den Linienstatus prüfen. Wenn also einer der vier Sensoren eine (dunkle) Linie identifiziert, wird die entsprechende Binärziffer auf "1" (logisch wahr) gesetzt, aber die LEDs gehen aus (keine Reflexion auf diesem Sensor vom Hintergrund).

Mit nun 4 Einzelsensoren erhöht sich der Umfang auf 16 verschiedene Status. Die folgende Tabelle zeigt die wichtigsten Status an:

Binäres Muster	Bedeutung	Status (Dezimal)
0000	alle 4 Sensoren auf weißem Hintergrund (keine Spur)	0
0010	nur R1 auf schwarzer Linie	2
0100	nur L1 auf schwarzer Linie	4
0110	innere Sensoren R1 und L1 auf schwarzer Linie	6
0111	scharfe Rechtskurve (L1 und R1 auf schwarzer Linie, Abzweigung nach rechts; L-förmige Abzweigung nach rechts)	7
1110	scharfe Linkskurve (L1 und R1 auf schwarzer Linie, Abzweigung nach links; L-förmige Abzweigung nach links)	14
1111	T-förmige Abzweigung (L1 und R1 auf der schwarzen Linie, Abzweigung nach rechts und links)	15

Eine X-förmige Kreuzung kann nur erkannt werden, indem man über die T-Kreuzung fährt und prüft, ob die mittlere Linie weiterläuft.

Das folgende Beispielprogramm lässt den Roboter so lange fahren, wie er sich auf einer schwarzen Linie mit den inneren beiden Sensoren befindet. Er hält an, wenn er von der Strecke abkommt, eine Kreuzung erkannt wird oder die Linie plötzlich endet:

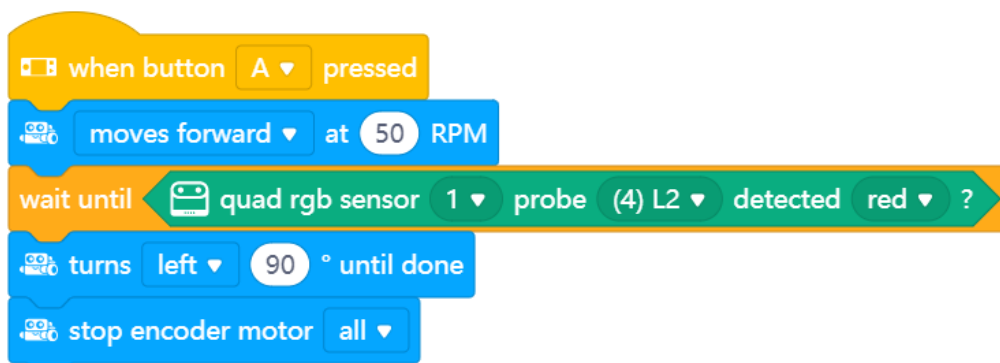


Codeblock:



Mit diesem Codeblock wählen Sie einen der Quad-RGB-Sensoren (L2, L1, R1, R2) aus und legen fest, ob er eine Linie, einen Hintergrund, Weiß, Schwarz oder eine andere der 5 Farben (Rot, Grün, Blau, Gelb, Cyan, Lila) erkennen soll. Für die Erkennung und Verfolgung von Linien werden die vorherigen Blöcke empfohlen, da sie mehr als einen Sensor gleichzeitig überprüfen (schneller, genauer, da sich der Roboter zwischen den Messungen nicht bewegt). Sie können diesen Block aber auch verwenden, um z. B. eine rote Farbmarkierung auf der linken Seite des Roboters zu erkennen - während der Roboter einer Linie folgt.

Wenn Sie im folgenden Programmierbeispiel die B-Taste drücken, fährt der mBot2 geradeaus (unter der Annahme, dass er einer Linie folgt) und macht eine 90°-Drehung, wenn eine rote Markierung auf der linken Seite der (imaginären) Strecke erkannt wird.



Codeblock:
 quad rgb sensor 1 ▼ deviation (-100~100)

Dieser Codeblock misst, wie weit der Roboter von einer schwarzen Spur abweicht. Sie können ihn für fortgeschrittene und gleichmäßige Linienverfolgung verwenden. Der vorherige Codeblock für die Linienverfolgung verwendet eine dreifache Unterscheidung: auf der Spur, links von der Spur oder rechts von der Spur (plus zusätzliche Erkennung von Kreuzungen in einigen Fällen). Aber im Alltag beim Fahrradfahren würden Sie die Lenkbewegung genauer an die Kurve anpassen. Dieser Block ermöglicht eine proportionale Linienverfolgung: je mehr der Roboter von der Spur abweicht, desto mehr wird er in die Gegenrichtung lenken. Wenn der Versatz von der Spur 0 ist, ist auch die Lenkung 0 (gerade).

In Schritt 3 werden wir besprechen, wie wir diesen Block verwenden können, um einer Linie "gleichmäßig" zu folgen, indem wir einen proportionalen Linienverfolger programmieren. Das Beispielprogramm unten zeigt die Abweichung auf dem Bildschirm an, Sie können also zunächst versuchen, den Roboter von Hand auf der Bahn zu bewegen:



2. Einige Programmierbeispiele des Quad-RGB-Sensors nachbilden und testen

In der obigen Tabelle ist jedem Codeblock des Quad-RGB-Sensors ein Programmierbeispiel zugeordnet. Sie werden diese Programmierbeispiele in mBlock nachbilden und testen. Denken Sie sich für ein Programmierbeispiel eine eigene Änderung des Codes aus.

3. Wie kann der mBot2 einer Linie folgen?

Die Grundlage der Aufgabe in "Schritt 3" dieser Lektion ist, dass der mBot2 als Reisebus der Strecke auf einer Karte folgen soll. Dazu müssen Sie die bisher erläuterten und getesteten Code-Beispiele verwenden und diese entsprechend modifizieren.

Verwenden Sie zunächst diesen Codierblock:

 quad rgb sensor 1 ▼ L1, R1's line ▼ status is (3) 11 ▼ ?

Wenn sich der Roboter auf der Strecke befindet (sowohl L1 als auch R1 erkennen eine Linie, Codewert3), soll der Roboter geradeaus fahren. Wenn nur einer der beiden inneren Sensoren eine Linie erkennt, dann soll er entsprechend nach links oder rechts abbiegen.

Verwenden Sie diese Fahrblöcke (es wird nur das Rechtsfahren gezeigt):



Sie starten einfach die Motoren nach Bedarf, und der nächste Codeblock wird sofort ausgeführt. Da Sie nicht wissen, wie weit der Roboter weg ist oder wann die nächste Kurve kommt, können die Fahrbefehle keine Fahrzeit oder Strecke enthalten (gleiches gilt für das Wenden). Bei der nächsten Iteration einer Programmschleife werden die neuen Sensordaten geprüft und die Fahrt entsprechend ausgewählt.

Wenn Sie die mitgelieferte Karte verwenden, stellen Sie sicher, dass Sie den Sensor wie beschrieben auf die hellste Farbe (gelber Bereich auf der Strecke) kalibrieren.

3. Ausprobieren (25 Min.)

Schritt 3: Ihn Ausprobieren

Sie haben bereits viel über den Quad-RGB-Sensor des mBot2 gelernt. Sie werden nun Ihr eigenes Computerprogramm in mBlock unter Verwendung des Quad-RGB-Sensors erstellen. In dieser Aufgabe werden Sie den mBot2 in einen Reisebus verwandeln. Der Reisebus fährt eine Tour durch die Sehenswürdigkeiten einer Stadt. In der Box des mBot2 befindet sich eine Karte mit einer Strecke. Der Reisebus folgt der schwarzen Linie auf der Karte. Im vorherigen Teil sollten Sie bereits einen einfachen, einer Linie folgenden Roboter programmiert haben, jetzt ist es an der Zeit, weitere Erweiterungen hinzuzufügen.

Um das Programm verständlicher zu machen, können Sie Ihre eigenen Codierblöcke "bauen". Der gesamte linienverfolgende Teil des Programms kann zu einem einfachen Befehl, z. B. "einfache_linie", zusammengefasst werden. Mit dem rot/rosafarbenen "Meine Blöcke" im Code-Menü können

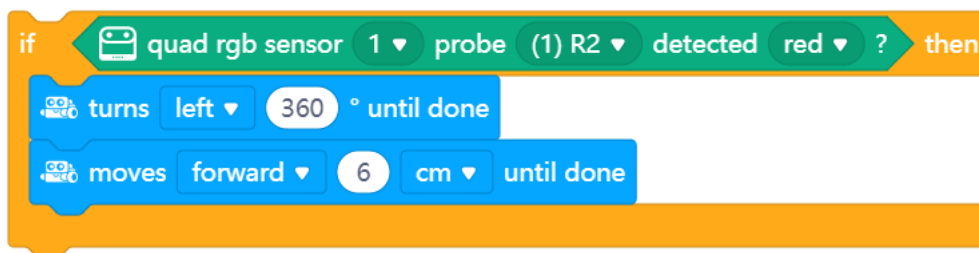


Sie einen benutzerdefinierten Block definieren und Ihren Code auf eine angenehme Weise strukturieren. Das folgende Beispielprogramm verwendet benutzerdefinierte Blöcke.

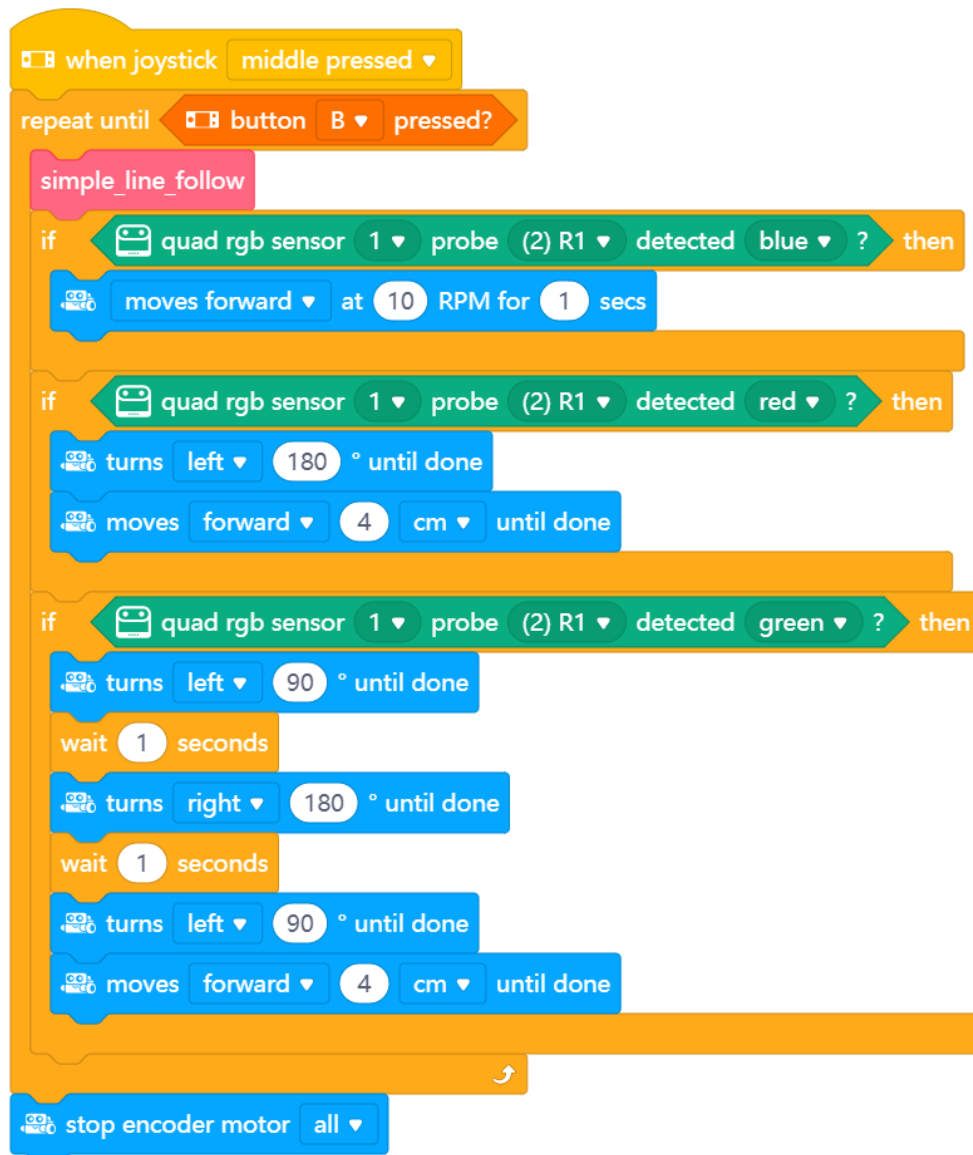
Der Reisebus folgt nicht nur der Strecke, sondern führt an einigen Sehenswürdigkeiten eine Aktion aus. Die Sehenswürdigkeiten sind durch einen farbigen Bereich (rot, gelb, grün und blau) gekennzeichnet.

Sie können sich die Aktionen ausdenken, die der Fahrer ausführen soll. Wollen Sie den Bus in Paris herumfahren lassen? Dann wäre es vielleicht lustig, die französische Nationalhymne zu spielen. Oder die Glocken des Big Ben, wenn der Bus durch London fährt.

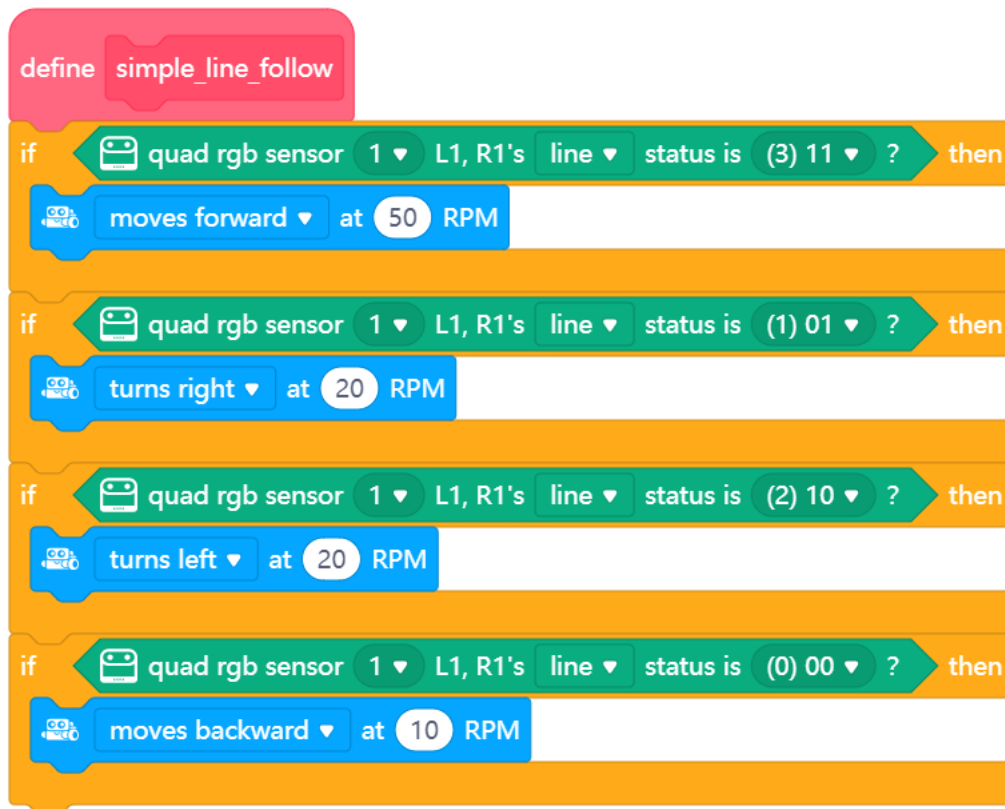
Natürlich können Sie es auch ein wenig einfacher halten. Im Beispiel unten lassen Sie den Reisebus an der roten Markierung umdrehen und an anderen farbigen Markierungen verschiedene Aktivitäten durchführen.



Verwenden Sie das Wissen, das Sie in "Schritt 2" dieser Lektion erworben haben. Natürlich können Sie auch selbst viel mit den verschiedenen Programmiermöglichkeiten in mBlock experimentieren. Kommen Sie nicht weiter? Dann können Sie die Programmierbeispiele unten verwenden. Sie können sie kopieren und an Ihre eigenen Wünsche und Vorlieben anpassen.



Dieses Programm startet, wenn Sie auf B drücken und es stoppt, wenn Sie auf A drücken. Es führt eine einfache Linienverfolgung mit eigenem "Mein Blöcke"-Code durch und prüft dann, ob der R1-Sensor entweder blau (sehr langsam fahren), rot (umdrehen) oder grün (auf beide Seiten schauen) erkennt. Um sicherzustellen, dass die Farbe nicht erneut erkannt wird, nachdem die Aktivitäten ausgeführt wurden, fährt der Roboter jedes Mal 4 cm nach vorne. Der Block "einfache_linie" ist wie folgt definiert:



Erweiterte Version:

Im vorherigen Schritt wurde die Abweichung von der Linie eingeführt. Dieser Block meldet, wie weit der Roboter von der Spur abweicht (-100 bis +100; 0 für perfekt in der Mitte) und kann zur Berechnung der Lenkung des Roboters verwendet werden. Dabei handelt es sich um eine proportionale Linienverfolgung - je mehr der Roboter von der Spur abweicht, desto mehr muss er dagegen steuern. Wenn der Roboter rechts von der Spur ist, ist die Abweichung positiv (der Roboter muss nach links fahren); bei negativen Werten ist der Roboter links von der Spur (er muss nach rechts fahren).

Um dies zu realisieren, benötigen Sie eine Grundgeschwindigkeit - das ist die Geschwindigkeit für die Vorwärtsbewegung auf einer geraden Strecke. Um nach links oder rechts zu lenken, muss sich eines der Räder schneller drehen als das andere - je größer die Differenz, desto stärker wird gelenkt.

Um die Lenkung zu berechnen, müssen wir die Abweichung zur Grundgeschwindigkeit des einen Motors addieren und vom anderen subtrahieren. Um einen Finderabgleich zu ermöglichen, wird mit einem Faktor eingestellt, wie stark sich die Abweichung auf die Lenkung auswirkt, da der

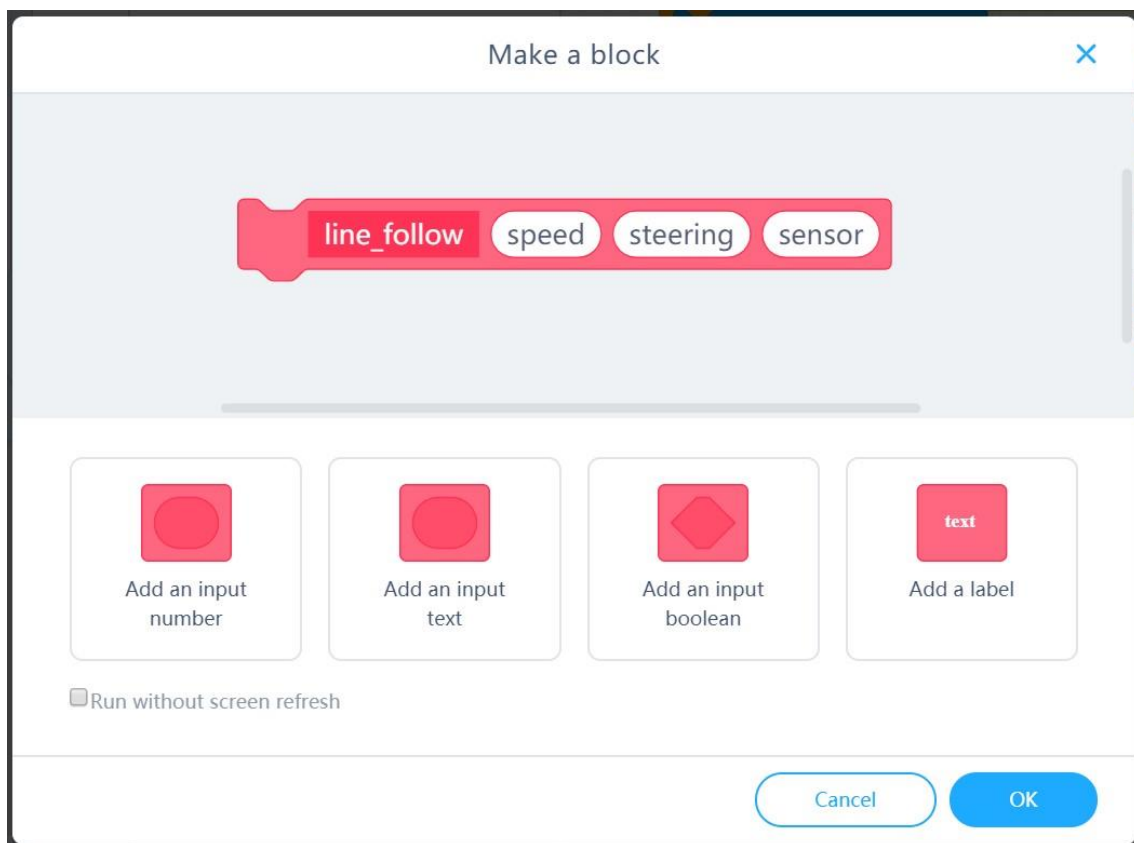
Bereich für die Abweichung selbst zu groß ist, um ihn direkt zur Steuerung der Motoren zu verwenden.

Aber zu welchem Motor muss der errechnete Wert addiert oder subtrahiert werden? Eine positive Abweichung bedeutet, dass der Roboter auf der rechten Seite aus der Spur ist und nach links abbiegen muss. Das bedeutet, dass das rechte Rad schneller drehen muss - wir addieren die skalierte Abweichung zum rechten Rad und subtrahieren sie daher vom linken.

speed + steering * sensor

speed - steering * sensor

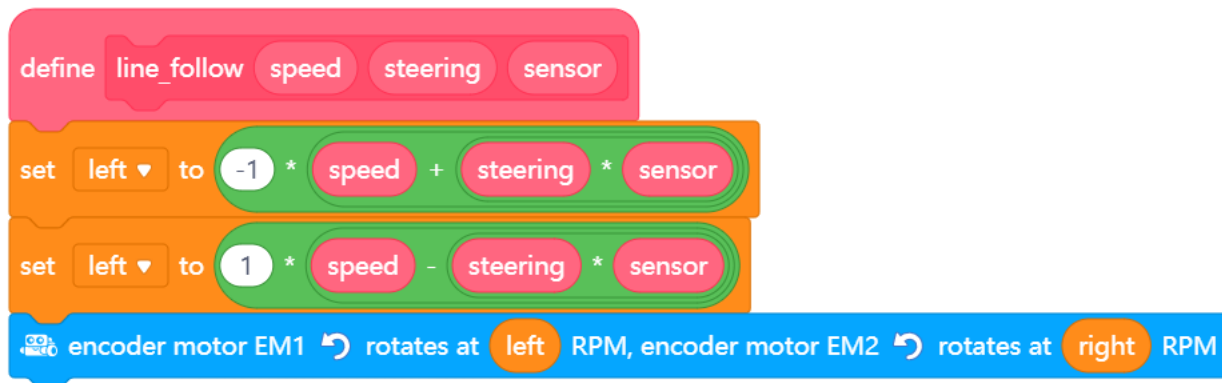
Beim Einrichten der "Meine Blöcke" können Sie Inputs hinzufügen und diese mit einem aussagekräftigen Namen versehen:



Das Einzige, was noch zu beachten ist, ist, dass der Bewegungsblock (siehe unten) negative Werte für den rechten Motor und positive für den linken Motor benötigt, um vorwärts zu fahren (das liegt daran, dass sie einander gegenüber montiert sind).

encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at 50 RPM

Um die Berechnung abzuschließen, müssen die Werte für den rechten Motor mit -1 multipliziert werden. Dies ist die Definition des proportionalen Linienfolgeblocks:



In der Hauptschleife eines Programms verwenden Sie diesen Block einfach mit einem Befehl:

line_follow 50 0.6 quad rgb sensor 1 ▼ deviation (-100~100)

Dieser ruft das Unterprogramm mit einer Grundgeschwindigkeit von 50 und einem Lenkfaktor von 0,6 auf und sendet die aktuelle Abweichung vom Sensorwert mit. Mit den Werten können Sie ein wenig experimentieren - wenn der Roboter "schwankt" (schnelles Wechseln von Links- auf Rechtsabbiegen usw.), ist der Lenkfaktor zu hoch, wenn er auch in leichten Kurven aus der Spur geht, ist er zu niedrig.

Bei dieser Aufgabe ist es hilfreich, den folgenden Schritt-für-Schritt-Plan zu verwenden. Haben Sie eine Idee, was Sie machen wollen? Wenn ja, besprechen Sie zunächst mit Ihrem Lehrer, ob dies machbar ist.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none"> In welche Richtung soll sich der mBot2 bewegen?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none"> Was benötigen Sie zusätzlich zum mBot2?
Schritt 3: Welche Codeblöcke benötigen Sie, damit sich der mBot2 bewegt?	<ul style="list-style-type: none"> Wie können Sie den mBot2 an jeder Ecke die Richtung ändern lassen? Welche Codeblöcke benötigen Sie dafür?



	<ul style="list-style-type: none">• Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML)• Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe
Schritt 4: Test und Implementierung	<ul style="list-style-type: none">• Ist die erste Version fertig? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf• Arbeiten Sie an den Verbesserungspunkten, bis Ihr mBot2 fehlerfrei über Ihre Strecke läuft• Ist es Ihnen gelungen? Filmen Sie das Endergebnis und fragen Sie Ihren Lehrer, ob Sie es auf sozialen Medien mit dem Hashtag #mBot2inclass posten dürfen

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Haben Sie es geschafft, den mBot2 als Reisebus für die Strecke zu programmieren?

In dieser Lektion haben Sie den Quad-RGB-Sensor kennengelernt und erfahren, wo Sie ihn im Alltag begegnen könnten. Sie wissen, wie Sie den Quad-RGB-Sensor des mBot2 so programmieren können, dass der Roboter einer Linie folgt. Sie haben auch gelernt, wie Sie den mBot2 so programmieren können, dass er eine Route basierend auf verschiedenen Farben fährt. Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was ist Ihnen Ihrer Meinung nach gut gelungen?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?

Lektion 6: Vorsichtiges Fahren

Thema: STEAM

Klasse(n): 5. Klasse und aufwärts

Dauer: 45 Minuten

Schwierigkeitsgrad: Anfänger

★ Lektionsziele

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Die Codeblöcke für den Gyroskop-Beschleunigungsmesser in CyberPi zu erkennen und zu verwenden.
- Ein eigenes Computerprogramm in mBlock zu erstellen, um die Änderungen an der Neigung des mBot2 zu messen und seine Bewegungen entsprechend anzupassen

★ Überblick

Ein Gyroskop misst Kippbewegungen, genauer gesagt die Geschwindigkeit von Dreh-/Kippbewegungen. Ein Beschleunigungssensor misst die Änderung der Geschwindigkeit. Beide Sensoren liefern unterschiedliche Informationen über die Position im Raum.

Kombiniert man beide Datentypen, lassen sich die Position und die Bewegung von Fahrzeugen, Robotern oder auch Raumsonden berechnen. Auf holprigen Straßen könnten Roboter zum Beispiel langsamer fahren, um ein Umkippen zu verhindern.



⚙️ Schwerpunkt

Am Ende der Lektion werden die Schüler wissen:

- Wie Gyroskope und Beschleunigungssensoren funktionieren und wie ihre Messungen kombiniert werden können
- Wie man einen Roboter dazu bringt, auf plötzliche Veränderungen zu reagieren und verunfallte oder holprige Straßen zu erkennen

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle

Lehrplan

Diese Lektion besteht aus vier Schritten und dauert insgesamt 45 Minuten.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none">• Gyroskope und Beschleunigungssensoren im Alltag• Welche Kippbewegungen kann der mBot2 erkennen?
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none">• Die verschiedenen Programmierblöcke von Gyroskop und Beschleunigungssensor kennenlernen.• Einige Programmierbeispiele dieser Sensoren nachbilden und testen.
25 Minuten	3. Ausprobieren <ul style="list-style-type: none">• Ihr eigenes Programm für den Roboter schreiben
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none">• Was haben Sie in dieser Lektion gelernt?• Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zum Zwecke späterer Diskussion.• Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass• Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Gyroskope und Beschleunigungssensoren im Alltag
2. Welche Kippbewegungen kann der mBot2 erkennen?

1. Gyroskope und Beschleunigungssensoren im Alltag

Wie erkennen Geräte wie Smartphones ihre Position (aufrecht halten, Display nach oben/unten gerichtet)? Wie lässt sich feststellen, ob ein Aufprall heftig genug ist, um Airbags auszulösen? Die Abschätzung der aktuellen Position ohne eine externe präzise Referenz ist seit Jahrhunderten eine Herausforderung auf See bei bewölktem Himmel - und sie ist immer noch eine sehr wichtige Aufgabe für Haushaltsgegenstände, Roboter und sogar Raumsonden. Sie alle nutzen mindestens zwei verschiedene Sensoren, Gyroskope und Beschleunigungsmesser. Ein Gyroskop arbeitet mit einem Kristall, der unter elektrischer Spannung steht. Dieser wandelt die Richtung der Kraft der Kippbewegung in ein elektrisches Signal um. Gyroskope messen die Drehgeschwindigkeit in Grad pro Sekunde und können mit Hilfe der Mathematik auch den Drehwinkel abschätzen - letzteres ist aber nicht so genau: Gyroskope "driften" mit der Zeit. Beschleunigungssensoren melden die Änderung der Position (wie Zu- oder Abnahme der Geschwindigkeit) - und in Bezug auf die Erdanziehungskraft melden sie die Fallbeschleunigung (im Vergleich zum freien Fall). Beide Arten von Sensoren können auf jeder Raumachse messen und melden entweder die Rotationsgeschwindigkeit oder die Beschleunigung auf der X-, Y- und Z-Achse.

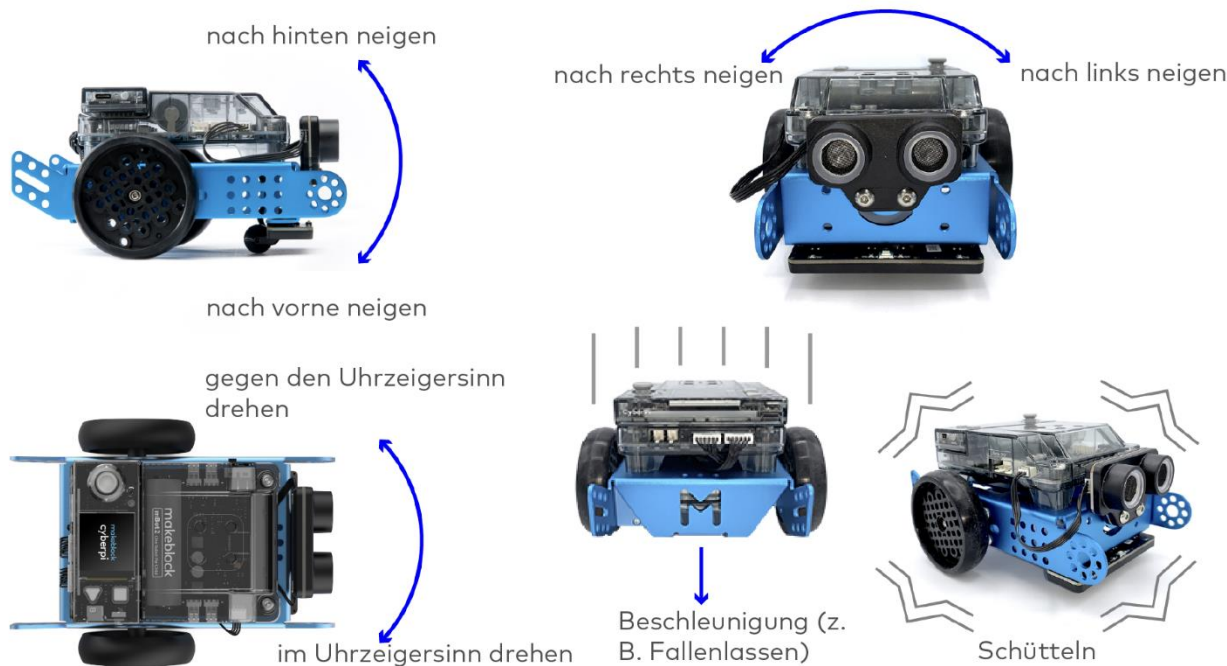
Durch die Kombination der Daten von Gyroskop und Beschleunigungssensor können Richtung und Stärke der Bewegungen berechnet werden (in der Realität ist dies nicht superpräzise, es ist also eher eine sehr gute Schätzung). In der Robotik werden diese Sensoren verwendet, um Fahrzeuge in unebenem Gelände (oder im Flug) zu stabilisieren und ihnen zu helfen, ihre programmierte Bahn beizubehalten. Beim mBot2 sind diese Sensoren in einem kleinen Chip auf dem CyberPi integriert.

Daher werden wir den kombinierten Sensor als "Bewegungssensor" bezeichnen.

2. Welche Kippbewegungen kann der mBot2 erkennen?

Bevor Sie lernen, im mBlock5 mit dem Bewegungssensor zu programmieren, ist es sinnvoll zu wissen, welche Kippbewegungen der mBot2 messen kann. Sie können die Messungen der Kippbewegungen auf dem CyberPi-Display anzeigen. In Lektion 2 haben Sie bereits gelernt, wie man das macht.

Der mBot2 kann die folgenden Kippbewegungen erkennen:



2. Praxisorientiert (10 Min.)

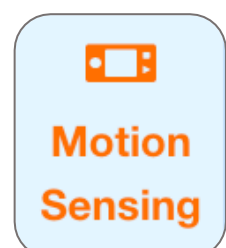
Schritt 2: Praxisorientiert

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Die verschiedenen Codeblöcke des Bewegungssensors kennenlernen
2. Einige Programmierbeispiele der Bewegungssensoren nachbilden und testen

1. Die verschiedenen Codeblöcke des Bewegungssensors kennenlernen

In mBlock gibt es mehrere Codeblöcke für den Bewegungssensor, die Sie in Ihren Programmen verwenden können. Diese Codeblöcke finden Sie in der Kategorie "Bewegungssensorik" des Blockfelds in mBlock. Diese Codeblöcke sind orange.

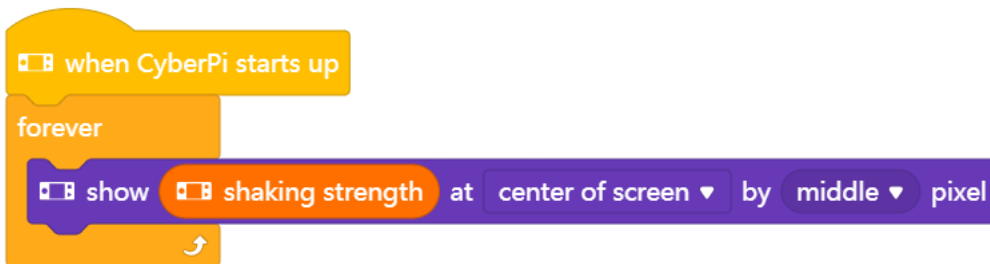


In der folgenden Tabelle finden Sie eine ausführliche Erklärung der Code-Blöcke zusammen mit einem Beispiel-Programmcode zum Testen.

Codeblock:

shaking strength

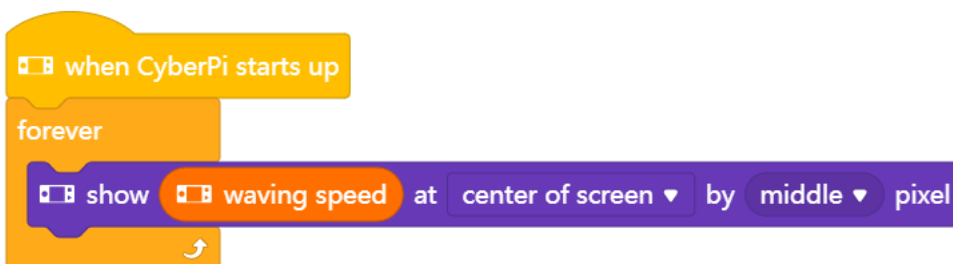
Dieser Codeblock meldet, wie stark der mBot2 erschüttert wird. Bei einem Wert von "0" wird der mBot2 überhaupt nicht erschüttert. Bei einem Wert von "100" wird eine sehr starke Erschütterungsbewegung erkannt. Im folgenden Programmierbeispiel wird die Stärke des Rüttelns des mBot2 auf dem Display angezeigt. Wenn der mBot2 aufhört zu rütteln, ist die Anzeige leer.



Codeblock:

waving speed

Dieser Codeblock dient zur Erfassung der Schwenkgeschwindigkeit. Die Schwenkgeschwindigkeit reicht von 0 bis 100. Diese Einheiten stellen keine Geschwindigkeit in m/s oder in Grad/s dar, sie sind nur eine einfache Bezugsgröße. Im folgenden Programmierbeispiel wird die Schwenkgeschwindigkeit des mBot2 auf dem Display angezeigt. Wenn der mBot2 aufhört zu schwingen, ist die Anzeige leer.



Codeblock:

 tilted forward ▾ angle (°)

Jede Neigung führt eine Bewegung in einem bestimmten Winkel in eine bestimmte Richtung aus. Bei der Angabe eines Winkels werden die Achsen X, Y und Z verwendet, um die Richtung zu bezeichnen. Anstelle dieser mathematischen Begriffe werden sie auch als Roll-, Nick- und Scherwinkel bezeichnet. Diese Begriffe stammen aus der Navigation von Flugzeugen. Roll ist die Achse von vorne nach hinten, Nick bezieht sich auf Auf-/Abwärtsdrehungen (Achse von links nach rechts) und Scherwinkel ist die Achse von oben nach unten, die für die Ausrichtung nach links/rechts zuständig ist.

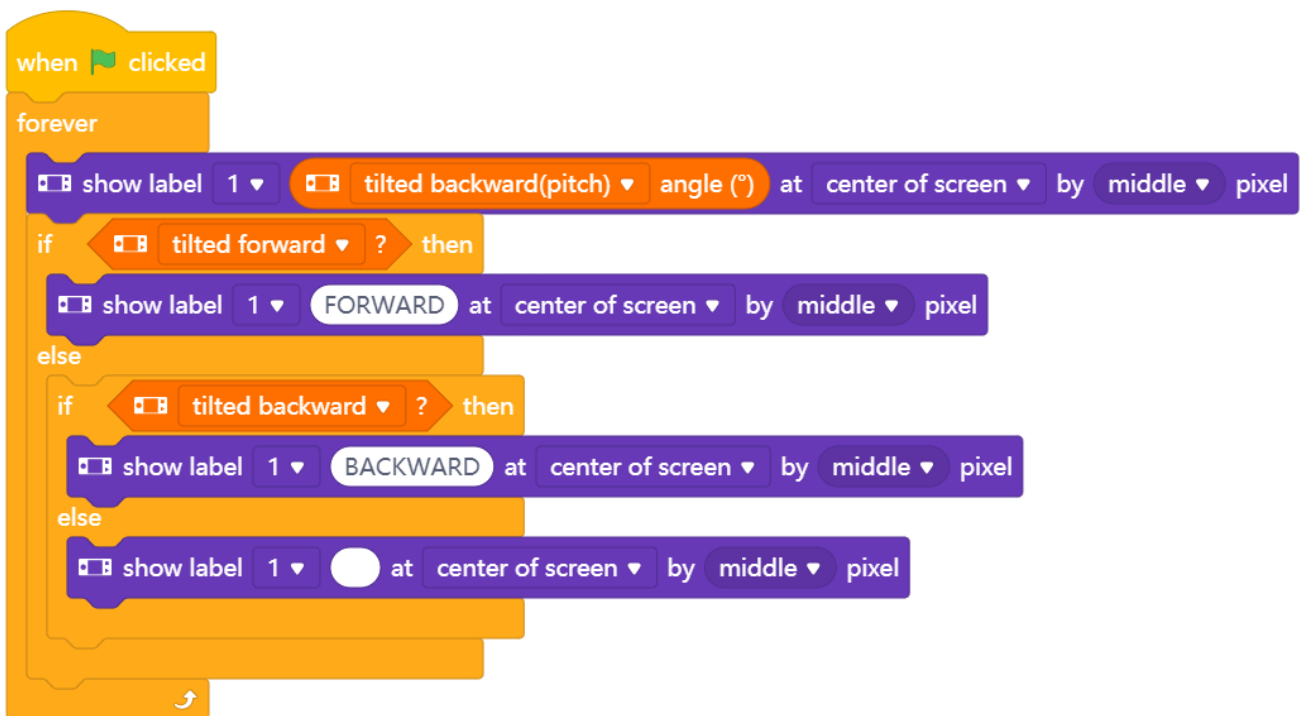
Sehen Sie sich die Abbildung unten an.



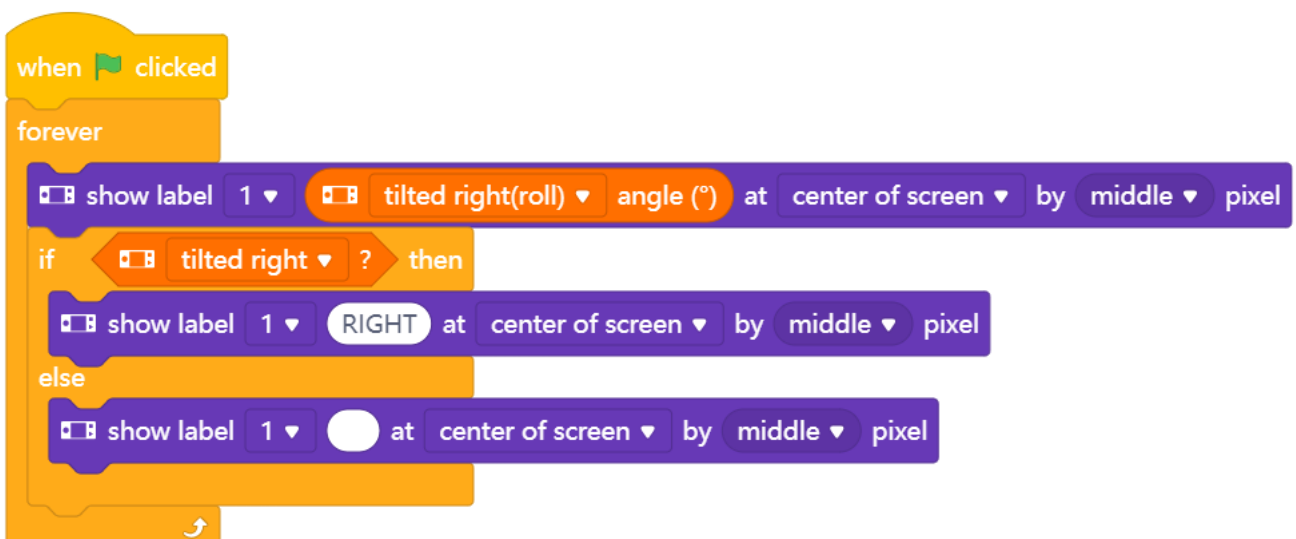
Winkel werden in Grad angegeben:

- Rollwinkel: -180° bis $+180^{\circ}$
- Scherwinkel: -180° bis $+180^{\circ}$
- Nickwinkel: -90° bis $+90^{\circ}$

Im folgenden Programm wird der Nickwinkel in der obersten Zeile angezeigt und wenn eine Kippbewegung nach vorne oder hinten erkannt wird, erscheint in der letzten Zeile das Wort "vorwärts" oder "rückwärts". Können Sie den Schwellenwert für die Erkennung erkennen?



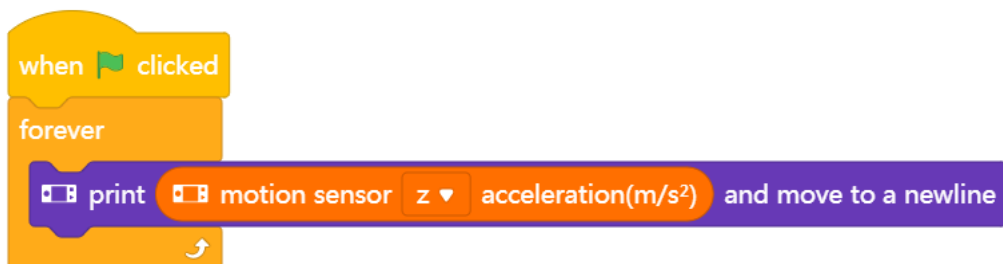
Versuchen Sie, mit den verschiedenen Einstellungen aus dem Dropdown-Menü zu experimentieren. Jetzt haben Sie die Kippbewegung auf der Nick/X-Achse analysiert. Wenn Sie auf tilt_left/right umschalten, denken Sie daran, auch die boolesch gemeldeten Blöcke umzuschalten:



Codeblock:

 motion sensor  acceleration(m/s²)

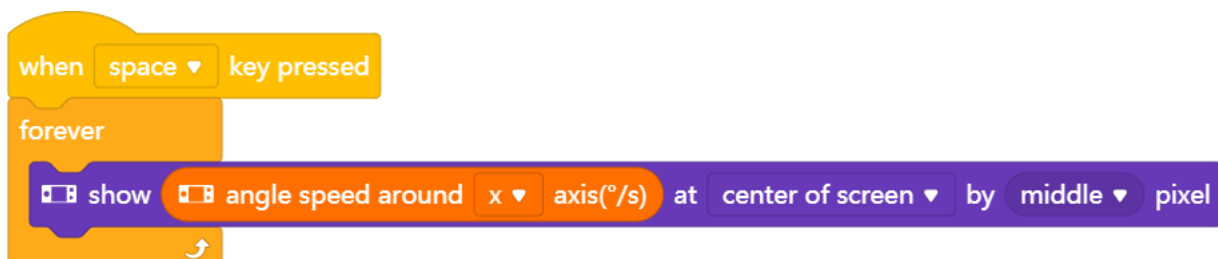
Bisher haben wir uns Kippbewegungen angesehen, aber die Bewegungssensoren enthalten einen weiteren Sensor, den Beschleunigungssensor. Er misst Änderungen der Geschwindigkeit des mBot2 im Laufe der Zeit. Das können langsame Erhöhungen der Fahrgeschwindigkeit ebenso sein wie ein plötzlicher Stopp oder sogar ein Seitenaufprall durch ein anderes Fahrzeug. Da diese Messungen auf der X-, Y- und Z-Achse des Roboters beruhen, können Sie einen Aufprall erkennen, wenn Sie eine schnelle Änderung der Beschleunigung feststellen. Der Beschleunigungsmesser zeigt die Beschleunigung in m/s² an. Im folgenden Programmierbeispiel wird die Beschleunigung des mBot2 auf der Z-Achse im Display angezeigt. Bewegen Sie das mBot2 von Hand und überprüfen Sie die Anzeige. Welche Werte zeigt es an, wenn der Roboter gegen Ihre Hand oder gegen ein Buch "kracht"? Können Sie das Programm ändern und einen Seitenaufprall simulieren?



Codeblock:

 angle speed around  axis(°/s)

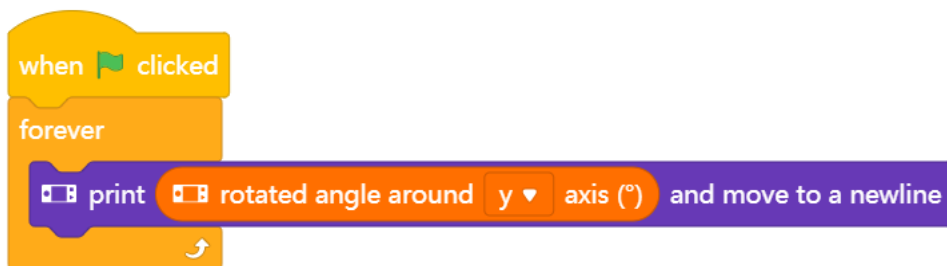
Dieser Codeblock zeigt die Geschwindigkeit an, mit der sich der mBot2 um eine bestimmte Achse dreht. Dies wird auch als Winkelgeschwindigkeit bezeichnet. Die Winkelgeschwindigkeit wird in Grad pro Sekunde (°/s) angezeigt. Im folgenden Programmierbeispiel wird die Winkelgeschwindigkeit des mBot2 um die X-Achse auf dem Display angezeigt.



Codeblock:

Dieser Codeblock zeigt den Winkel an, mit dem sich der mBot2 um eine bestimmte Achse dreht. Der Winkel wird in Grad (°) angezeigt und wird dabei relativ gemessen. Die Ausgangsposition wird festgelegt, wenn der mBot2 eingeschaltet wird. Diese Daten werden aus der Drehgeschwindigkeit berechnet, die das Gyroskop meldet. Sie kann mit der Zeit drifteten und ist nicht so stabil - versuchen Sie, den mBot2 mit dem unten stehenden Programm seitlich zu kippen und überprüfen Sie die Werte. Gehen sie in den Ausgangszustand zurück?

Im untenstehenden Programmierbeispiel wird der Winkel, in dem sich der mBot2 um die Y-Achse dreht, auf dem Display angezeigt.



Es gibt zwei wichtige Aspekte: Dieser Block wird Drehbewegungen verfolgen, die größer als eine Umdrehung sind. Verwenden Sie diesen Block, um zu zählen, wie oft sich der Roboter gedreht hat. Betrachtet man die Z-Achse, so erhöht eine Drehung im Uhrzeigersinn die von diesem Block gemeldeten Werte:



Während der gedrehte Winkel um die Z-Achse abnimmt:



Nach 3 Drehungen um 360° meldet der letzte Block "-1080°", der vorherige Block zeigt aber nur "0" an (zurück zur Ausgangsposition, Mehrfachdrehungen werden ignoriert). Wenn dann 4 mal gegen den Uhrzeigersinn gedreht wird (vorher noch einmal), ist der "rotate_clockwise"-Winkel wieder bei 0°, während "rotated_angle_around_z axis" nun 360° anzeigt (alle Angaben ohne Gyroskop-Drift).

Wenn Sie nur kleine Drehbewegungen berücksichtigen müssen, bieten für die X- und Y-Achse die zuvor besprochenen "gekippter (___) Winkel"-Blöcke eine zuverlässigere Messung.

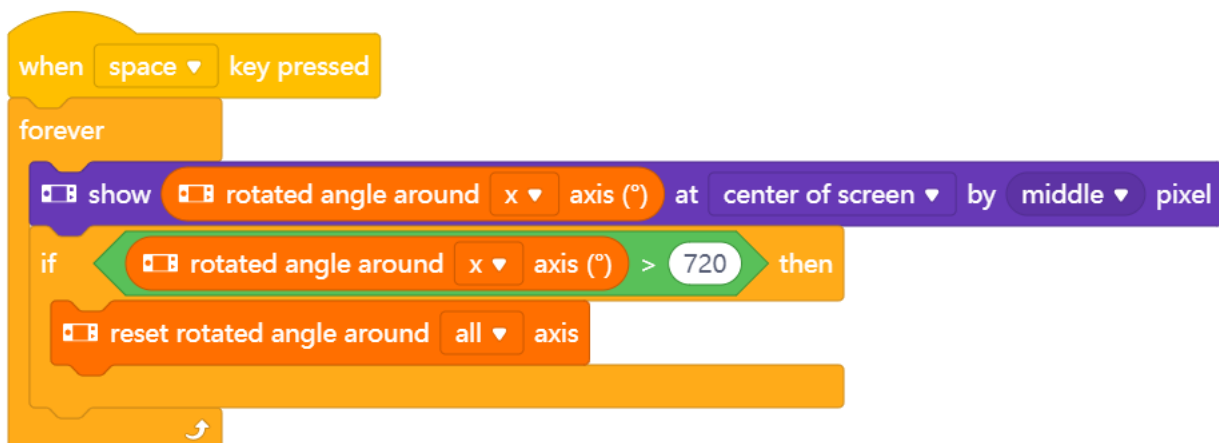
Es gibt eine Ausnahme: Drehungen um die Gier/Z-Achse können nur mit Hilfe des Gyroskops gemessen werden. Wenn Sie die Grade messen möchten, um die sich der mBot2 nach links/rechts dreht, müssen Sie mit diesen Daten arbeiten.

Codeblock:

 reset rotated angle around all axis

Wenn Sie den Winkel bestimmen möchten, den der mBot2 macht, ist es einfacher, die Änderung vom Wert 0 aus zu messen. Dieser Codeblock setzt den gemessenen Wert eines Winkels auf den Startwert 0 zurück und definiert damit eine neue Startposition für zukünftige Drehungen.

Im folgenden Programmierbeispiel wird der Winkel, um den sich der mBot2 um die z-Achse dreht, auf dem Display angezeigt. Wenn der Winkel über 720° liegt, wird der gemessene Wert des Winkels auf 0 gesetzt und es wird erneut gemessen.



```
when space key pressed
  forever
    show rotated angle around x axis (°) at center of screen by middle pixel
    if rotated angle around x axis (°) > 720 then
      reset rotated angle around all axis
```

Vergessen Sie nicht, dass es zwei Blöcke gibt, um die Drehung zu messen: einer kümmert sich nur um volle Umdrehungen, der andere zählt weiter. Dieser Rücksetzblock setzt alle (oder ausgewählte) "rotated_around_axis"-Blöcke auf Null. Er hat keine Auswirkung auf den "rotate clockwise"/"rotate counterclockwise"-Block. Dieser hat einen eigenen Rücksetzblock:

 reset yaw angle

Das Setzen auf Null gilt nur für Drehungen, die allein vom Gyrosensor gemeldet werden. Der gekippte ___ Winkel meldet immer von einer perfekt nivellierten Ebene senkrecht zum Erdmittelpunkt und kann nicht beliebig auf Null gesetzt werden.

2. Einige Programmierbeispiele der Bewegungssensoren nachbilden und testen

In der obigen Tabelle ist für jeden Codeblock des Bewegungssensors ein Programmierbeispiel angegeben. Sie werden diese Programmierbeispiele in mBlock nachbilden und testen.

Der Bewegungssensor am CyberPi registriert die Kippbewegungen des mBot2. Beim Testen der obigen Programmierbeispiele; Sie benötigen daher eigentlich nur den CyberPi. Falls es bequemer ist, entfernen Sie den CyberPi vom mBot2 und führen Sie die Kippbewegungen mit dem CyberPi beim Testen selbst aus. Sie können die Programmierbeispiele im Live-Modus testen. Da sich der Akku im mBot2-Shield befindet, müssen Sie den CyberPi über das USB-C-Kabel mit dem Computer verbinden, um die Programme auszuführen.

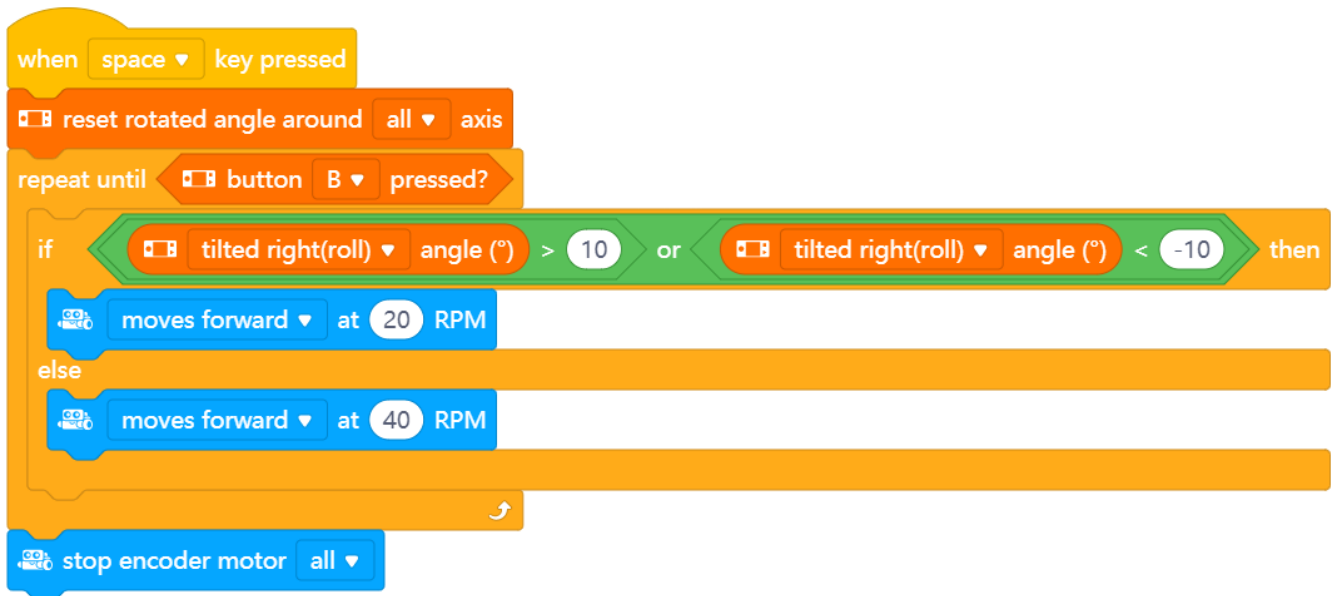
3. Ausprobieren (25 Min.)

Schritt 3: Ausprobieren

Sie haben bereits viel über den Bewegungssensor des mBot2 gelernt. Sie werden nun Ihr eigenes Computerprogramm in mBlock erstellen, das diesen Sensor verwendet.

In dieser Aufgabe werden Sie eine holprige Strecke für den mBot2 erstellen. Sie können diese Strecke z. B. aus (kleinen) Steinen oder Knetmasse auf Karton herstellen. Die Idee ist, dass der mBot2 über die Strecke fahren soll. Wenn der mBot2 (stark) durchgerüttelt wird, muss er langsamer fahren. Sie können auch einen Gegenstand auf den mBot2 legen, der während der Fahrt nicht herunterfallen darf.

Knifflige Aufgabe, nicht wahr? Zum Glück müssen Sie nicht alles selbst erfinden! Unten sehen Sie ein Programmierbeispiel, das Sie erweitern können. In diesem Programmierbeispiel wird der mBot2 langsamer, wenn er zu große Kippbewegungen macht.



Verwenden Sie das Wissen, das Sie in „Schritt 2“ dieser Lektion erworben haben. Natürlich können Sie auch selbst viel mit den verschiedenen Programmiermöglichkeiten in mBlock experimentieren.

Bei den Überlegungen zu dieser Aufgabe ist es hilfreich, den folgenden Schritt-für-Schritt-Plan zu verwenden.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none"> • Welche Strecke soll der mBot2 fahren? • Wann soll der mBot2 langsamer werden? • Möchten Sie, dass die Sensoren des mBot2 auch Daten auf dem Display anzeigen?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none"> • Was benötigen Sie zusätzlich zum mBot2?
Schritt 3: Welche Codeblöcke benötigen Sie, um den mBot2-Antrieb herzustellen?	<ul style="list-style-type: none"> • Wie werden Sie den mBot2-Antrieb herstellen? • Welche Codeblöcke werden Sie verwenden? • Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML) • Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe

<p>Schritt 4: Wie wollen Sie die Daten der Sensoren auf dem Display anzeigen?</p>	<ul style="list-style-type: none"> • Wie wollen Sie die Daten auf dem Display anzeigen? • Welche Codeblöcke werden Sie verwenden? • Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML) • Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe
<p>Schritt 5: Test und Implementierung</p>	<ul style="list-style-type: none"> • Ist die erste Version fertig? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf. • Arbeiten Sie an den Verbesserungspunkten, bis Ihrer mBot2 genau das tut, was Sie sich vorgestellt haben • Ist es Ihnen gelungen? Filmen Sie das Endergebnis und fragen Sie Ihren Lehrer, ob Sie es auf sozialen Medien mit dem Hashtag #mBot2inclass posten dürfen

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Wie ist diese Aufgabe gelaufen? Ist das Objekt während der Fahrt ordentlich auf dem mBot2 geblieben?

In dieser Lektion haben Sie Sensoren zur Messung der Bewegung um Achsen und der Beschleunigung kennengelernt, das Gyroskop und den Beschleunigungsmesser. Sie sind in einer einzigen kleinen Komponente integriert, die auf dem CyberPi platziert ist. Diese kombinierten Sensoren werden verwendet, um die Position eines Roboters im 3D-Raum zu bestimmen. Man nennt sie Inertial Measurement Units, IMU. Sie haben gelernt, wo man ihnen im Alltag begegnet. Sie wissen, welche Kippbewegungen und Bewegungsänderungen der mBot2 messen kann und wie Sie verschiedene Codeblöcke verwenden können, um diese Messungen in Ihren Programmen mit mBot2 zu nutzen.

Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was ist Ihnen Ihrer Meinung nach gut gelungen?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?

Lektion 7: Ein Netzwerkspiel

Thema: STEAM

Klasse(n): 5. Klasse und aufwärts

Dauer: 45 Minuten

Schwierigkeitsgrad: Anfänger

★ Lektionsziele

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Mehrere mBot2 drahtlos miteinander kommunizieren zu lassen, ohne dass ein WIFI-Zugangspunkt erforderlich ist
- Ihr eigenes Computerprogramm in mBlock zu verfassen, um mehrere mBot2 zu steuern
- Eine Zufallsauswahl in einem Computerprogramm anzuwenden

★ Überblick

LAN ist die Abkürzung für Local Area Network. Der mBot2 mit seinem eingebauten WIFI-Modul baut automatisch eine Netzwerkverbindung auf, sobald sich mehrere mBot2 oder CyberPi im selben Raum im Umkreis von 30 Metern befinden. Über diese Netzwerkverbindung können der mBot2 und der CyberPi miteinander kommunizieren und Befehle ausführen.

🔗 Schwerpunkt

Am Ende der Lektion werden die Schüler wissen:

- Was eine LAN-Verbindung ist
- Wie man diese Verbindung einrichtet und wo man sie einsetzt
- Wie die Zufallsauswahl beim Programmieren funktioniert

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Der mBot2 mit einem CyberPi (mehrere mBot2 und/oder CyberPi, wenn möglich, für den Schritt des Ausprobierens)
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle
- Farbkarten (blau ist empfehlenswert)

Lehrplan

Diese Lektion besteht aus vier Schritten und dauert insgesamt 45 Minuten.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none">• Eine LAN-Verbindung im Alltag• So richten Sie eine LAN-Verbindung mit mBot2 und CyberPi ein
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none">• Die verschiedenen Codeblöcke zum Einrichten einer LAN-Verbindung kennenlernen• Programmierbeispiele für die LAN-Verbindung nachbilden und testen
25 Minuten	Abschnitt 3 - Ausprobieren <ul style="list-style-type: none">• Sein eigenes Spiel erstellen
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none">• Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zum Zwecke späterer Diskussion.• Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass• Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Eine LAN-Verbindung im Alltag
2. So richten Sie eine LAN-Verbindung mit mBot2 und CyberPi ein

1. Eine LAN-Verbindung im Alltag

LAN steht für Local Area Network und beschreibt ein Netzwerk von Computern und Peripheriegeräten mit einer bestimmten räumlichen Reichweite, meist innerhalb eines Gebäudes. Das Netzwerk ermöglicht den Computern den Datenaustausch untereinander innerhalb dieses Netzwerks und (nicht zwingend) mit anderen Netzwerken wie dem Internet. Dazu benötigen sie eine Art von Verbindung (drahtgebunden per Kabel oder drahtlos/WIFI per Funk) und eine gemeinsame "Sprache" (Protokoll genannt). Normalerweise verbinden sich die einzelnen Computer mit einem Verwaltungsgerät, das den Datenverkehr schaltet und leitet ("Router"). In einem WIFI-Netzwerk ist dies ein dedizierter Access Point oder der Internet-Router bietet einen eingebauten n Access Point.

Fallen Ihnen einige Beispiele ein, bei denen ein LAN-Netzwerk verwendet wird? In dieser Lektion sind der mBot2 und der CyberPi die verschiedenen Geräte, die über eine LAN-Verbindung miteinander kommunizieren.

2. Wie Sie eine LAN-Verbindung mit mBot2 und CyberPi einrichten

Um eine LAN-Verbindung zwischen mBot2 und CyberPi einzurichten, benötigen Sie nicht viel. Der mBot2 und der CyberPi (bzw. der mBot2 untereinander) werden automatisch drahtlos miteinander kommunizieren können, eine Verbindung zu einem Wi-Fi-Zugangspunkt oder Router ist dafür nicht erforderlich.

Wird ein Router verwendet? Dann müssen alle mBot2 und CyberPi auf den gleichen Kanal des Routers gelegt werden. Dies muss in den Einstellungen des Routers eingestellt werden.

2. Praxisorientiert (10 Min.)

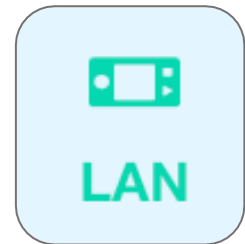
Schritt 2: Praxisorientiert

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Die verschiedenen Codeblöcke für die LAN-Kommunikation kennenlernen
2. Einige Beispiele der LAN-Kommunikation nachbilden und testweise programmieren

1. Die verschiedenen Codeblöcke für die LAN-Kommunikation kennenlernen

In mBlock gibt es mehrere Codeblöcke, die Sie zum Einrichten einer LAN-Verbindung verwenden können. Sie finden diese Codeblöcke in der Kategorie "LAN" des Blockfeldes in mBlock. Diese Codeblöcke sind grün.



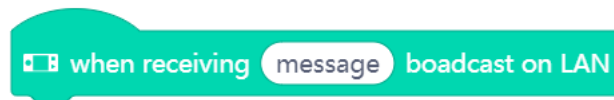
Die folgende Tabelle zeigt ein Beispiel für diese Codeblöcke zum Einrichten einer LAN-Verbindung.

Codeblock:



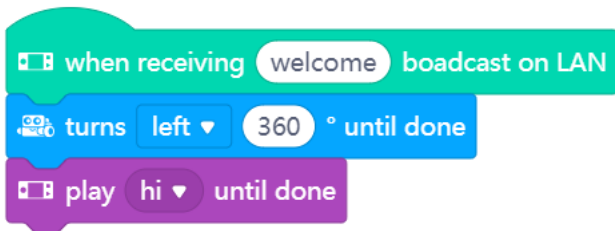
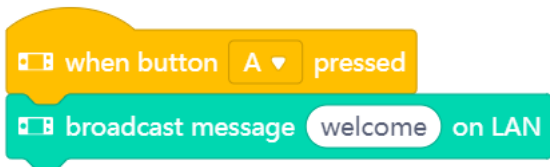
Mit diesem Codeblock können Sie eine Nachricht an alle anderen mBot2 und CyberPi im selben Netzwerk senden. Sie können die Nachricht benennen (wie ein Thema).

Codeblock:



Anstatt ständig auf einen Empfang von Nachrichten zu prüfen, gibt es einen Event-Block. Der an diesen Block angehängte Code wird nur einmal ausgeführt, wenn eine Nachricht mit dem angegebenen Thema empfangen wird.

Im folgenden Programmierbeispiel gibt es zwei mBot2 und/oder CyberPi, die über eine LAN-Verbindung unter Verwendung dieser beiden Codeblöcke miteinander kommunizieren. Das erste Programm ist die "Sendeeinheit", die für den ersten mBot2 bzw. CyberPi bestimmt ist. Das zweite Programm ist der Empfänger-Code für den zweiten mBot2 oder CyberPi. Sie können beide Code-Teile in beide CyberPi/mBot2 hochladen, da der zugrunde liegende Code dafür sorgt, dass die Sendeeinheit ihre eigene Nachricht nicht empfängt.



Nachdem Sie die Programmiercodes hochgeladen haben, müssen Sie das mBot2 und/oder den CyberPi neu starten.

Wenn Sie nun die Taste A des ersten mBot2 oder CyberPi drücken, empfängt das zweite mBot2 oder CyberPi die gesendete Nachricht (z. B. "Willkommen"). Was passiert, wenn Sie am zweiten CyberPi oder mBot2 die Taste B drücken? Was müssten Sie ändern, um verschiedene Nachrichten zurückzusenden?

Codeblock:



Mit diesem Codeblock können Sie eine Nachricht (mit einem bestimmten Thema wie zuvor) und einem bestimmten Wert (Zahl oder Text) an alle anderen mBot2 und CyberPi im Netzwerk senden.

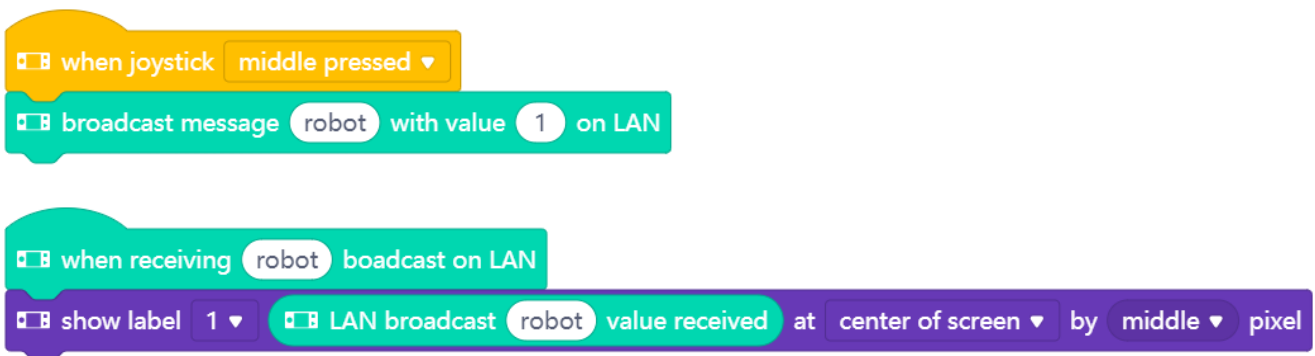
Codeblock:

Für den empfangenden Code können Sie denselben Ereignisblock verwenden und das Thema angeben, auf das er genau wie im vorherigen Beispiel reagieren soll. Um den Wert (Text oder Zahl) zu nutzen, gibt es einen eigenen Reporter-Block:



Im folgenden Programmierbeispiel gibt es zwei mBot2 und/oder CyberPi, die über eine LAN-Verbindung unter Verwendung dieser beiden Codeblöcke miteinander kommunizieren. Das erste Programm ist die "Sendeeinheit", die für den ersten mBot2 bzw. CyberPi bestimmt ist. Das zweite Programm ist der Empfänger-Code für den zweiten mBot2 oder CyberPi. Sie

können beide Code-Teile in beide CyberPi/mBot2 hochladen, da der zugrunde liegende Code dafür sorgt, dass die Sendeeinheit ihre eigene Nachricht nicht empfängt.



Nachdem Sie die Programmiercodes hochgeladen haben, müssen Sie das mBot2 und/oder den CyberPi neu starten. Wenn Sie nun den Joystick am ersten mBot2 oder CyberPi drücken, erhält der zweite mBot2 oder CyberPi die gesendete Nachricht (Wert 1).

2. Einige Beispiele der LAN-Kommunikation nachbilden und testweise programmieren

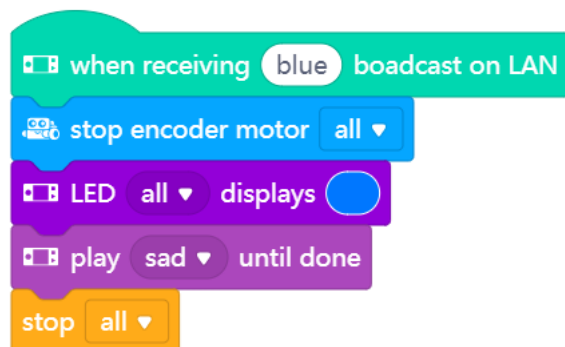
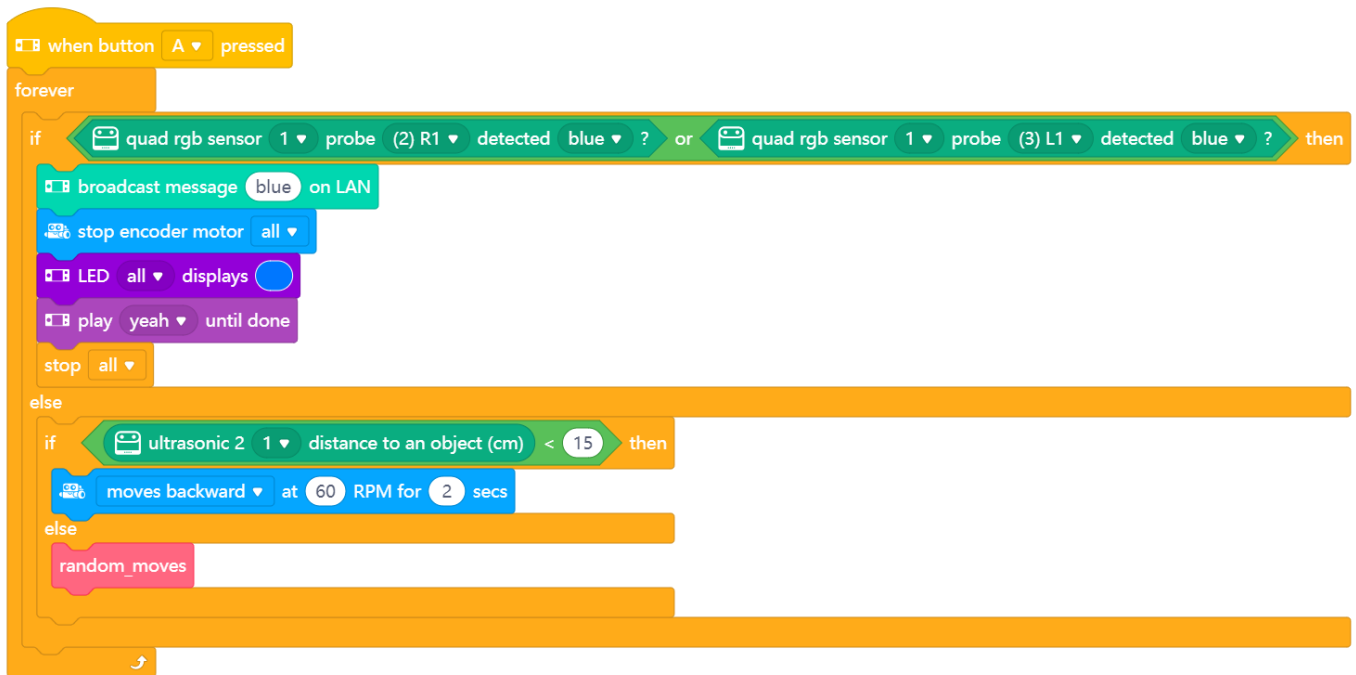
In der obigen Tabelle wird jeder Codeblock der LAN-Verbindung von einem Programmierbeispiel begleitet. Sie werden diese Programmierbeispiele in mBlock nachbilden und testen. Denken Sie sich für ein Programmierbeispiel eine eigene Erweiterung aus.

3. Ausprobieren (25 Min.)

Schritt 3: Ausprobieren

Sie haben bereits viel über die LAN-Verbindung des mBot2 gelernt. Sie werden nun Ihr eigenes Computerprogramm in mBlock unter Verwendung der LAN-Kommunikation erstellen können. In dieser Aufgabe werden Sie selbst ein Spiel programmieren. Es heißt "Auf der Suche nach ... der Farbe Blau". In diesem Spiel sucht eine Gruppe von mBot2 auf dem Boden nach der Farbe Blau. Wenn einer der mBot2 die Farbe findet, informiert er die anderen über eine LAN-Nachricht und gewinnt das Spiel.

Klingt ziemlich kompliziert! Zum Glück müssen Sie nicht alles selbst programmieren. Im Folgenden finden Sie ein Programmierbeispiel. Das gleiche Programm sollte auf alle mBot2 hochgeladen werden, damit sie alle den gleichen Anweisungen folgen.

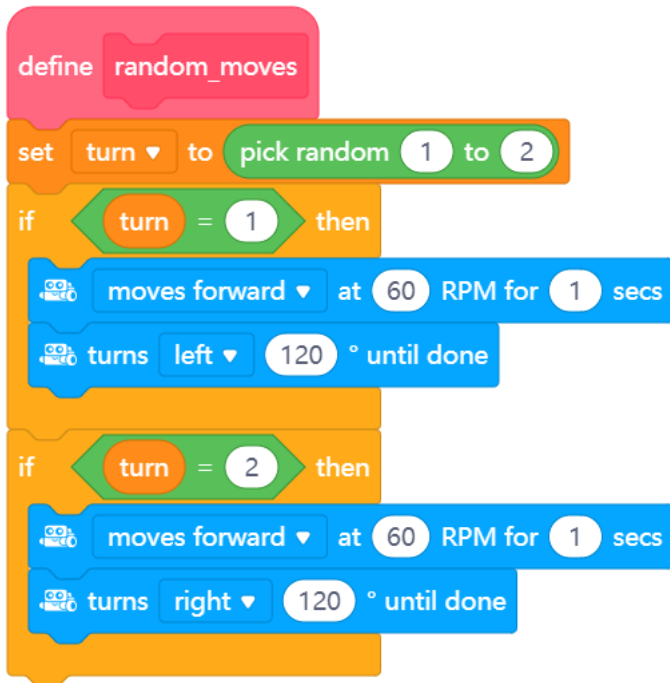


Um den mBot2 nach dem Zufallsprinzip fahren zu lassen, können Sie den folgenden Codeblock verwenden:

pick random 1 to 10

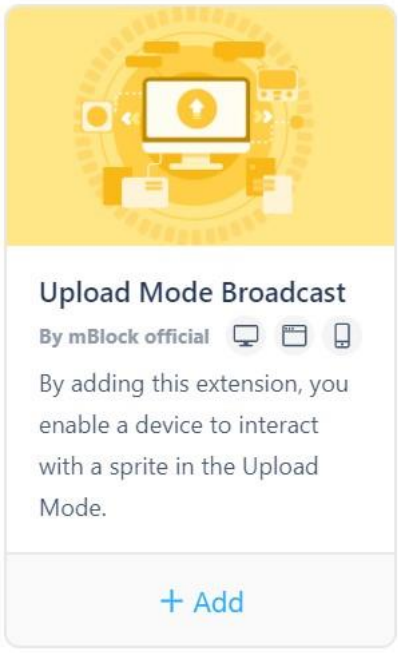
Dieser Codeblock wird verwendet, um nach dem Zufallsprinzip einen Wert innerhalb eines bestimmten Bereichs auszuwählen. Jedem Wert des Bereichs kann eine bestimmte Anweisung zugewiesen werden. Auf diese Weise kann der mBot2 auf nichtlineare Weise aus einer Reihe von Anweisungen auswählen. Um Ihr Programm übersichtlicher zu gestalten, können Sie die

Blöcke für diese Anweisung abtrennen und unter einen benutzerdefinierten Block legen, den Sie dann in Ihrem Hauptprogramm verwenden können:

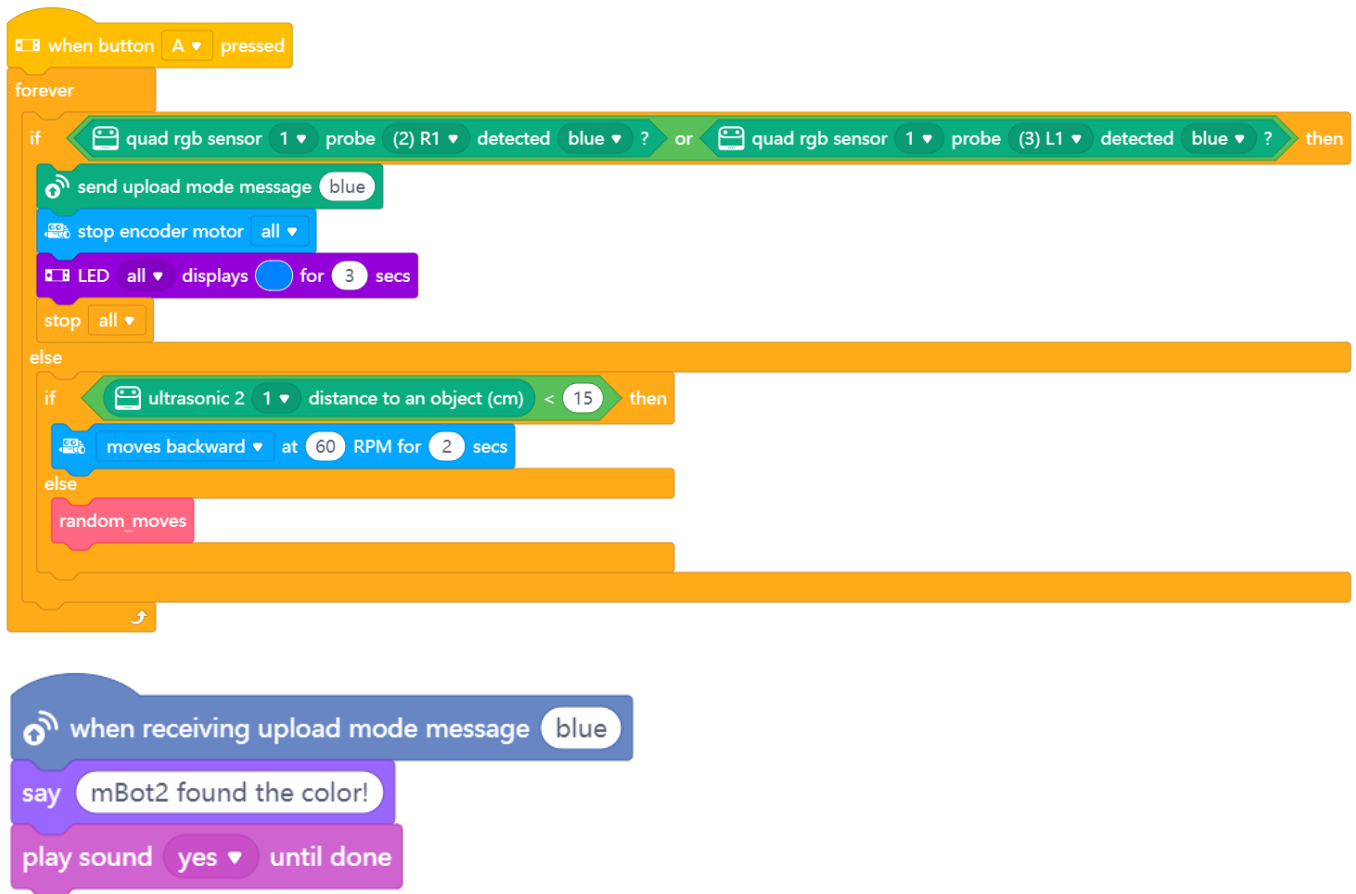


Sie haben nur einen mBot2? Dann können Sie die Kommunikation zwischen zwei Geräten simulieren, indem Sie ein Sprite in den mBlock Phase einfügen. Die Phase ist der Bereich in der linken oberen Ecke des Bildschirms des mBlocks.

Sie müssen eine neue Erweiterung zu den Codeblöcken für das Sprite und den mBot2 hinzufügen, die "Hochlade-Modus Broadcast" genannt wird. Sie ermöglicht das Senden und Empfangen von Nachrichten zwischen dem Sprite und dem Roboter, während sich der Roboter im Hochlade-Modus befindet, ähnlich den Nachrichten, die über LAN-Kommunikation gesendet werden.

	Erweiterung	Hinzufügen
Für das Sprite		<ul style="list-style-type: none"> • Klicken Sie auf „Sprites“ • Klicken Sie auf „Erweiterungen“ • Klicken Sie auf „Sprite-Erweiterungen“ • Suchen Sie die Erweiterung mit der Bezeichnung "Hochlade-Modus Broadcast" und fügen Sie sie hinzu
Für mBot2		<ul style="list-style-type: none"> • Klicken Sie auf "Geräte" • Klicken Sie auf „Erweiterungen“ • Klicken Sie auf "Geräteerweiterungen" • Suchen Sie die Erweiterung mit der Bezeichnung "Hochlade-Modus Broadcast" und fügen Sie sie hinzu (sie befindet sich möglicherweise nicht auf der ersten Seite der Erweiterungsbibliothek)

In diesem Programmierbeispiel sendet der mBot2, wenn er die Farbe gefunden hat, eine Nachricht an das Panda-Sprite in der Phase von mBlock, und der Panda sagt, dass der mBot2 die Farbe gefunden hat.



Sie können die obigen Programmierbeispiele kopieren, aber Sie können sich auch eine eigene Erweiterung ausdenken. Es ist dann sinnvoll, den folgenden Schritt-für-Schritt-Plan zu verwenden. Haben Sie eine Vorstellung davon, was Sie machen wollen? Wenn ja, besprechen Sie zunächst mit Ihrem Lehrer, ob dies machbar ist.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none"> Was wollen Sie programmieren?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none"> Welche Materialien benötigen Sie dafür?
Schritt 3: Welche Codeblöcke benötigen Sie dafür?	<ul style="list-style-type: none"> Wie wollen Sie den mBot2 fahren und kommunizieren lassen? Welche Codeblöcke werden Sie verwenden? Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML)

	<ul style="list-style-type: none"> • Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe.
Schritt 4: Test und Implementierung	<ul style="list-style-type: none"> • Ist die erste Version schon einsatzbereit? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf • Arbeiten Sie an den Verbesserungspunkten, bis der mBot2 genau das tut, was Sie sich vorgestellt haben. • Ist es Ihnen gelungen? Filmen Sie das Endergebnis und fragen Sie Ihren Lehrer, ob Sie es auf sozialen Medien mit dem Hashtag #mbot2inclass posten dürfen

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Ist es Ihnen gelungen, das Spiel zu programmieren?

In dieser Lektion haben Sie gelernt, was eine LAN-Verbindung ist und wo Sie ihr im Alltag begegnen können. Sie wissen, wie Sie eine LAN-Verbindung zwischen verschiedenen Geräten einrichten können. Außerdem haben Sie gelernt, wie Sie mit dem mBot2 und dem CyberPi ein eigenes LAN-Netzwerk erstellen können. Des Weiteren wissen Sie, wie Sie einem Computerprogramm eine Zufallsauswahl hinzufügen können.

Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was hat Ihrer Meinung nach gut funktioniert?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?

Lektion 8:

mBot2 zu Ihren Diensten

Thema: STEAM

Klasse(n): 5. Klasse und aufwärts

Dauer: 45 Minuten

Schwierigkeitsgrad: Fortgeschrittener Anfänger

★ Lektionsziele

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Den mBot2 mittels Spracherkennung und WIFI-Verbindung Aufgaben ausführen zu lassen
- Spracherkennung mit Variablen für Dialoge zu kombinieren
- Nachrichten über Cloud Messaging zu senden
- Boolesche Aussagen mit Sensordaten zu verwenden.

★ Überblick

Spracherkennung ist die Technologie, die das gesprochene Wort (aber auch ganze Sätze) wieder in Text umwandelt. Es ist eine recht rechenintensive Aufgabe, auch nur ein einziges Wort allein durch die Variation der Frequenz zu erkennen. Und um von praktischem Nutzen zu sein, sollte die Spracherkennung unabhängig von einer individuellen Stimme funktionieren: eine hohe oder tiefe Stimme, langsames oder schnelles Sprechen sollten keinen Unterschied machen. Spracherkennung ist ein Teil der Benutzerinteraktion in der Unterhaltungselektronik (wie Smartphones, TVs, Smart Home), aber auch für Assistenzsysteme. In dieser Aufgabe werden Sie einen kurzen Dialog mit dem mBot2 programmieren, der als Serviceroboter fungiert. Sie können diese Aufgabe später erweitern, um einen Benutzerdialog auch für andere Aufgaben zu simulieren.

🔗 Schwerpunkt

Am Ende der Lektion werden die Schüler wissen:

- Was Spracherkennung ist und wie man sie anwendet
- Wie man eine Internetverbindung über WIFI herstellt
- Wie man Spracherkennung einsetzt und einen Dialog mit dem Benutzer aufbaut
- Wie man Nachrichten an entfernte Computer über Cloud Messaging sendet

- Wie man Variablen verwendet, um eine Phase in einem Prozess zu beschreiben (z. B. Bestellung, Benutzerinteraktion)

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook), oder ein Tablet mit installierter mBlock-App
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle

Lehrplan

Diese Lektion besteht aus vier Schritten und dauert insgesamt 45 Minuten.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none"> • Spracherkennung im Alltag • Die Spracherkennung mit mBot2 kennenlernen
10 Minuten	2. Praxisorientiert <ul style="list-style-type: none"> • Eine WIFI-Verbindung einrichten • Spracherkennung programmieren und anwenden • Cloud-Messaging
25 Minuten	3. Ausprobieren <ul style="list-style-type: none"> • Den mBot2 zu einem Roboter-Kellner programmieren
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none"> • Showtime: Zeigen Sie, was Sie mit Ihrem Roboter gemacht haben, in einem lustigen, kurzen Film zum Zwecke späterer Diskussion. • Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass • Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus zwei Teilabschnitten:

1. Spracherkennung im Alltag
2. Die Spracherkennung mit dem mBot2 kennenlernen

1. Spracherkennung im Alltag

Spracherkennung ist die Technologie, die Wörter und Sätze anhand des Klangs der menschlichen Stimme identifizieren kann und diese in Text zurückübersetzt. Dies ist eine sehr rechenintensive Aufgabe, da alles, was die Algorithmen betrachten können, eine Änderung der Frequenz über die Zeit ist. Aber auch das ist von Mensch zu Mensch unterschiedlich - man denke nur an tiefe oder hohe Stimme, schnelles Sprechen oder einen Dialekt. Selbst die Stimmung kann den "Ton" verändern, nach dem derselbe Satz klingt. Die jahrzehntelange Forschung in Informatik und Linguistik lässt diese komplexen Aufgaben immer mehr in den Alltag einfließen. Manche Erkennungssysteme erfordern eine Art "Training", bei dem der Benutzer bestimmte Texte liest, die dem Erkennungsalgorithmus bekannt sind - so wird ein Computermodell an den individuellen Tonfall angepasst. Andere erkennen nur ganz bestimmte Wörter - sie werden meist in Benutzerdialogsystemen eingesetzt. Einige Systeme können Texte unabhängig vom Kontext erkennen und übersetzen.

Der CyberPi und der mBot2 können die Spracherkennung nutzen, ohne dass eine komplexe Codierung notwendig ist. Da der Output aber "nur" Text ist, muss das Programm diesen analysieren und weiterverarbeiten.

Spracherkennung wird in der Unterhaltungselektronik wie Fernsehern, Smartphones oder Smart Homes eingesetzt, um dem Benutzer zusätzlichen Komfort zu bieten - um nach einer Verbindung zu fragen, einen Termin in den Kalender einzutragen oder die Raumtemperatur und das Licht durch bloßes Sagen ändern zu lassen.

Neben dem reinen Komfort kann die Spracherkennung auch eine große Hilfe für behinderte Menschen sein oder zusätzliche Sicherheit bieten, z. B. kann ein Fahrer in einem Auto eine Navigationsänderung aufgrund eines bevorstehenden Staus nur per Sprache freigeben, anstatt mit Knöpfen oder einer bildschirmbasierten Schnittstelle zu interagieren.

In dieser Aktivität werden Sie einige der Funktionen der Spracherkennung nutzen.

2. Die Spracherkennung mit dem mBot2 kennenlernen

Der mBot2 bietet eine Spracherkennung, die auf Blockcodierung basiert - es ist so einfach wie das Starten der Erkennung für eine bestimmte Zeit (z. B. 2, 3 oder 5 Sekunden) mit einem einzelnen Block und das Abrufen des erkannten Textes als Zeichenkette durch einen anderen Block. Die Codierungsblöcke werden weiter unten erklärt.

Diese Spracherkennung kann viele verschiedene Sprachen verstehen und ist nicht auf eine kleine Menge von vordefinierten Wörtern beschränkt. Daher ist es eine sehr rechenintensive Aufgabe - sie ist so anspruchsvoll, dass der mBot2 oder der CyberPi sie nicht selbst bewältigen können... sie müssen über das Internet um Unterstützung bitten. Deshalb muss zunächst eine Internetverbindung hergestellt werden, und ein Benutzerkonto wird ebenfalls benötigt. Durch das Einrichten eines Benutzerkontos wird ein "Schlüssel" generiert, mit dem man sich bei den verwendeten Services anmelden kann (zur Datennutzung und zum Datenschutz siehe unten). Die Spracherkennung nimmt den Ton auf, wie Sie dies in der 3. Aktivität getan haben, und sendet diese Aufnahme dann zur Verarbeitung an ein sehr leistungsfähiges Netzwerk von Computern. Diese kümmern sich um die Übersetzung zurück in Text und senden den Text zurück an den CyberPi/mBot2. Das Netzwerk von Computern wird üblicherweise als "die Cloud" bezeichnet. Es handelt sich dabei nicht mehr um einen bestimmten Computer, sondern um einen riesigen Cluster von ihnen, der die Rechenaufgaben zwischen ihnen verteilt. Man kann das Internet nutzen, um Daten zwischen mehreren CyberPi's in verschiedenen Netzwerken und auch an verschiedenen Orten (wie in der Schule und zu Hause) zu übertragen. Auch dieser digitale Service wird in der Cloud verarbeitet.

Hinweis zur Datennutzung und zum Datenschutz:

Die Spracherkennung erfordert eine Internetverbindung für die Übertragung des aufgezeichneten Audiomaterials und die Rücksendung des transkribierten Textes. Die aufgezeichneten Audiodaten werden auf Servern in Europa, der Microsoft Azure Cloud, verarbeitet, aber nicht dauerhaft gespeichert. Für die Nutzung der Services wird ein "Schlüssel" benötigt, der beim Anlegen eines Benutzerkontos auf Makeblock.com automatisch generiert und von den Codierblöcken intern verwendet wird. Es ist nicht nötig, diesen Schlüssel irgendwo manuell einzugeben.

Außer einer Email und einem selbstgewählten Passwort sind keine weiteren persönlichen Informationen (wie Namen, Bilder, Geschlecht...) erforderlich.

Die Internetverbindung erfordert einen WIFI-Zugang über SSID (das ist der Name des Funknetzwerks) und Passwort - zertifikatsbasierte Anmeldungen werden nicht unterstützt. Die

SSID und das Passwort werden mit dem Programm im Klartext gespeichert, so dass es sinnvoll ist, einen temporären Hotspot zu verwenden. Idealerweise können Sie ein Smartphone mit einer Datenflatrate verwenden.

2. Praxisorientiert (10 Min.)

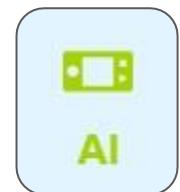
Schritt 2: Praxisorientiert:

In diesem Schritt machen Sie sich mit den verschiedenen Codeblöcken der KI- und IoT-Funktionen vertraut. Dieser Schritt besteht aus drei Teilabschnitten:

4. Eine WIFI-Verbindung einrichten
5. Spracherkennung programmieren und anwenden
6. Messaging und Kommunikation in LAN und Cloud

1. Eine WIFI-Verbindung einrichten

Öffnen Sie mBlock und verbinden Sie den CyberPi im Reiter "Geräte". Wechseln Sie in den Upload-Modus. Suchen Sie nach dem Verbinden in den Codeblöcken die Blockkategorie AI (Artificial Intelligence/Künstliche Intelligenz) und klicken Sie sie an. Sie sehen nun die verfügbaren Funktionen einschließlich der Spracherkennung (wenn die Blöcke ausgegraut sind, heißt es, dass Sie vergessen haben, in den Upload-Modus zu wechseln...).



Codeblock:



Dieser Block startet Verbindungsversuche mit dem bereitgestellten drahtlosen Netzwerk. Sie benötigen einen Router oder einen Zugangspunkt, der es Geräten ermöglicht, sich mit dem Netzwerk zu verbinden und auf das Internet zuzugreifen. Einige Schulen lassen möglicherweise nur bekannte Geräte zu - jeder Netzwerkadapter (kabelgebunden oder drahtlos) hat eine eindeutige Nummer, die MAC-Adresse. Wenn diese Nummer bei diesen gesicherten Netzwerken unbekannt ist, verweigern sie die Verbindung, selbst wenn das Passwort korrekt ist.

Der einfachste Weg, dies zu umgehen, ist dann ein temporärer Hotspot, z. B. durch ein Smartphone mit einer Datenflatrate, die nur für den CyberPi/mBot2 in einer Unterrichtsstunde genutzt wird (prüfen Sie die rechtlichen Bestimmungen Ihrer Schule). Alle Blöcke, die diesem Codeblock folgen, werden sofort ausgeführt, es wird also nicht auf einen Verbindungsaufbau gewartet ("non-blocking"). Sie müssen mit dem nächsten Block selbst eine Warteroutine codieren:

Codeblock:

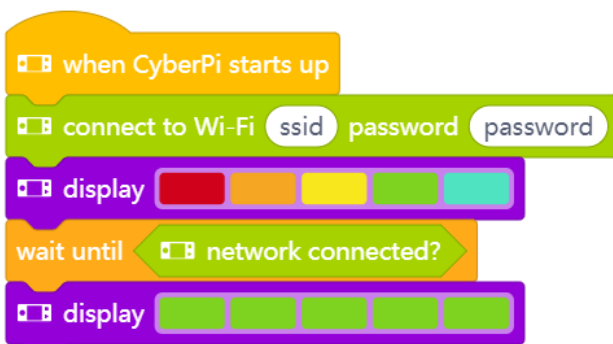


Dieser boolesche Block prüft, ob die Verbindung zum Netzwerk hergestellt werden konnte - wenn ja, meldet er ein True. Sie können ihn zusammen mit einem Wait-Block verwenden:



Diese Kombination stoppt alle weiteren Code-Ausführungen im selben Thread, bis eine Verbindung erfolgreich hergestellt wurde. Vielleicht möchten Sie dies später benutzerfreundlicher gestalten, indem Sie über Verbindungsversuche informieren, eine erfolgreiche Verbindung melden oder sogar eine Zeitüberschreitung hinzufügen (z. B. einen Neustart, wenn 30 Sek. lang keine Verbindung hergestellt werden kann).

Um die Spracherkennung zu nutzen, müssen Sie den CyberPi mit dem Internet verbinden. Nachfolgend sehen Sie das Programmierbeispiel, das Sie verwenden werden, um den CyberPi mit dem Internet verbinden zu lassen. Sobald der CyberPi mit dem Internet verbunden ist, sehen Sie dies auf dem Display und die LEDs leuchten grün.





Diese Beispiele verwenden einen ereignisbasierten Ansatz, d. h. ein Thread im Programm versucht beim Start, die Internetverbindung zu aktivieren. Die anderen Threads starten die Erkennung. Aber vorerst wird nicht geprüft, ob die Internetverbindung hergestellt werden kann, bevor es weitergeht - und es wird dem Benutzer auch nicht angezeigt, was passiert, außer, dass es erfolgreich war. Vielleicht möchten Sie dies später beim Ausprobieren eigener Ideen ändern...

Achten Sie darauf, dass Sie den Namen (SSID) und das Passwort für das WIFI-Netzwerk korrekt eingeben (die Abbildung gibt ein Beispiel!). Bei beiden wird zwischen Groß- und Kleinschreibung unterschieden. Verbinden Sie den mBot2 mit einem USB-Kabel mit Ihrem Computer und wechseln Sie in den Upload-Modus. Erstellen Sie den Programmiercode von oben und versuchen Sie, ob der Code funktioniert. Ein Tipp ist, dies mit einem Hotspot von Ihrem Mobiltelefon aus zu tun, da ein Schulnetzwerk oft nur bekannte Geräte zulässt (White-Listed MAC-Adressen). Achten Sie darauf, dass der Datentarif die Internetnutzung abdeckt! Verbinden Sie nun den CyberPi mit Ihrem Computer.

Sie haben nun den mBot2 mit dem Internet verbunden. Sie können nun mit dem mBot2 über die Cloud kommunizieren. Dies ist das, was man als Internet der Dinge (IoT) bezeichnet. Wenn Sie einen separaten CyberPi verwenden, um den Status anzuzeigen (z. B. die an die Küche gesendete Bestellung oder den Liefervorgang), muss dieser mit demselben WIFI-Netzwerk verbunden sein. Damit soll sichergestellt werden, dass beide Geräte (mBot2-Cyberpi und eigenständiger CyberPi) denselben Kanal im WIFI-Netzwerk verwenden. Der Router/Access Point entscheidet über die Kanäle. Damit ist sichergestellt, dass alle Geräte auf dem gleichen Kanal für den Datenaustausch sind. Sie können sich auch auf die LAN-Kommunikation in Lektion 7 beziehen.

2. Spracherkennung programmieren und anwenden

Sie werden nun die Spracherkennung programmieren und testen. Sie werden den Programmiercode, in dem Sie den mBot2 mit dem WIFI verbinden lassen, um einen Code für die Spracherkennung ergänzen.

Codeblock:



Dieser Codeblock wird im Folgenden in einem Beispiel verwendet - er macht das Gegenteil von Spracherkennung: Mit diesem Block können Sie den CyberPi/mBot2 einen bereitgestellten Text laut vorlesen lassen. Dabei wird der Text an den Cloud-Service

übertragen, und die synthetisierte Audiodatei wird zurückgeschickt. Es gibt eine automatische Erkennung der verwendeten Sprache, so dass die gesprochene Sprache die gleiche sein sollte, die der Text verwendet.

Codeblock:

Die Spracherkennung ist in 12 verschiedenen Sprachen verfügbar. Wenn Ihre Muttersprache nicht unterstützt wird, können Sie vielleicht Englisch verwenden. Dieser Codeblock startet eine Audioaufzeichnung für die bestimmte Zeit und sendet sie zur Erkennung an den Cloud-Service. Für die Ergebnisse benötigen Sie den folgenden Codeblock:

Codeblock:

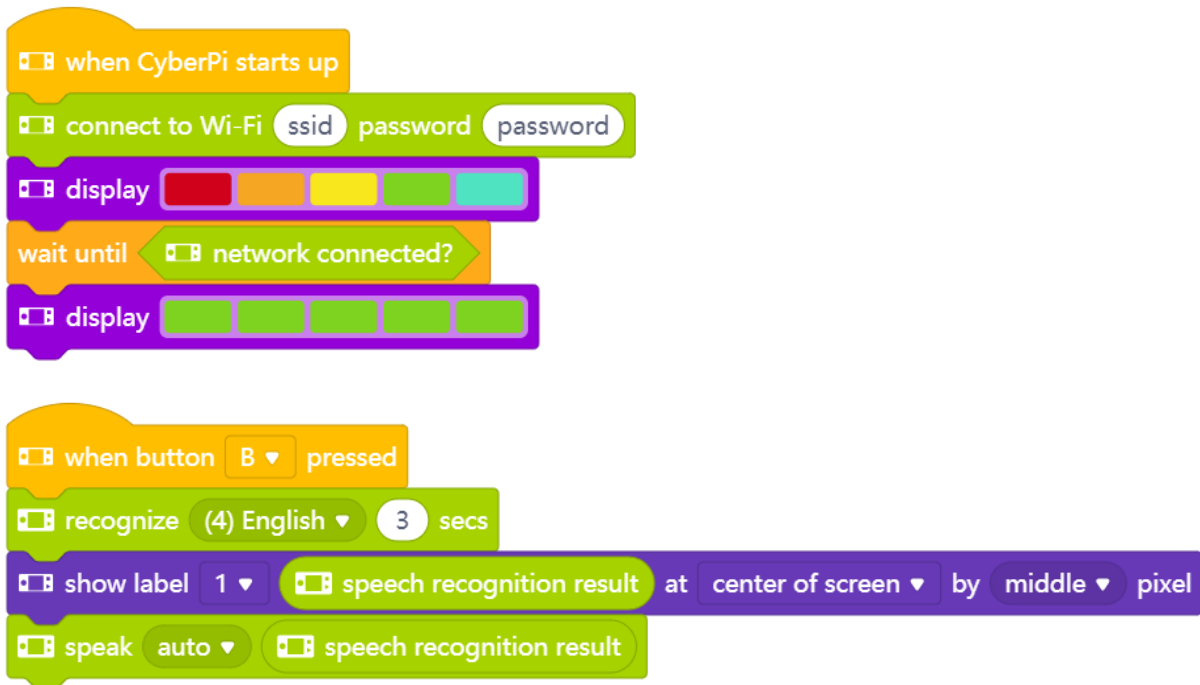
Auf die Ergebnisse der Spracherkennung kann über diesen Reporter-Block zugegriffen werden. Sie können das Ergebnis in einer Variablen speichern, falls Sie es wiederverwenden möchten, oder es auf dem Bildschirm anzeigen oder mit einer Analyse weiterverarbeiten (z. B. "enthält der Text das Wort 'ja'?").

Codeblock:

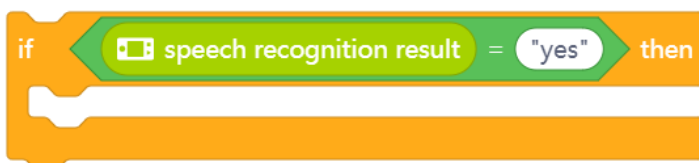
Die KI-Funktionen ermöglichen sogar die sprachliche Übersetzung von Texten. Dieser Codeblock ist ein Reporterblock, mit dem Sie den bereitgestellten Text in eine (unterstützte) Sprache Ihrer Wahl übersetzen können. Sie könnten diesen Block zusammen mit den vorherigen und dem Wissen, das Sie aus den vorherigen Aktivitäten gewonnen haben, verwenden und einen sprachgesteuerten Übersetzer aufbauen: Drücken Sie eine Taste und übersetzen Sie 5 Sekunden Sprache in eine Fremdsprache...Sie könnten sogar verschiedene ausprobieren und Ihre Fremdsprachenkenntnisse testen. 😊

Im Programmierbeispiel unten ist der mBot2 so programmiert, dass er beim Drücken der Taste B die Sprache für 3 Sekunden erkennt. Das Ergebnis wird auf dem CyberPi-Display angezeigt und die Erkennung wird ausgesprochen. Erstellen Sie den untenstehenden Programmiercode nach und testen Sie ihn aus. Überprüfen Sie, was Sie gesagt haben und was "verstanden" und zurückübersetzt wurde.

Funktioniert es? Dann schauen Sie, was Sie sonst noch mit der Spracherkennung machen können!



Sie können die Spracherkennung auch zur Sprachsteuerung von mBot2 verwenden. Sie können bestimmte Anweisungen bestimmten Spracherkennungsergebnissen zuordnen, aber achten Sie darauf, wie spezifisch Sie mit den Ergebnissen sind. Zum Beispiel könnten Sie die Anweisung so gestalten, dass nur "ja" erkannt wird, wenn die Taste B gedrückt wird. Was passiert aber, wenn Sie "auf jeden Fall ja" oder "ja natürlich" sagen? Dann würde ein direkter Abgleich mit...



... nicht funktionieren... das Erkennungsergebnis wäre anders (String "ja" ist nicht gleich String "auf jeden Fall ja"). Um eine breitere Menge von Antworten zuzulassen, die alle "ja" bedeuten, können Sie einfach prüfen, ob die Antwort-Zeichenkette das Wort "ja" enthält:



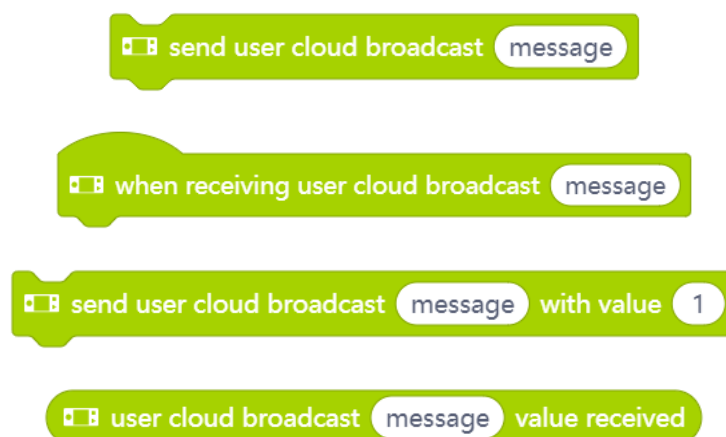
3. Messaging und Kommunikation in LAN und Cloud

Sie können das Ergebnis der Spracherkennung auch verwenden, um mit mehreren mBot2 oder CyberPi zu kommunizieren. In Lektion 7 haben Sie bereits die LAN-Verbindung kennengelernt und erfahren, wie Sie diese einrichten.

Neben dem Senden einer Nachricht über das LAN, was voraussetzt, dass die Geräte "lokal" verbunden sind, können Sie einen weiteren Cloud-Service verwenden, um Nachrichten an Geräte an entfernten Standorten zu senden - sofern diese eine Internetverbindung haben. Diese Blöcke befinden sich in der Blockkategorie IoT (Internet of Things).



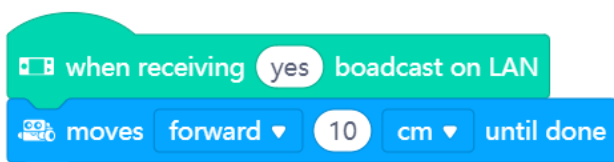
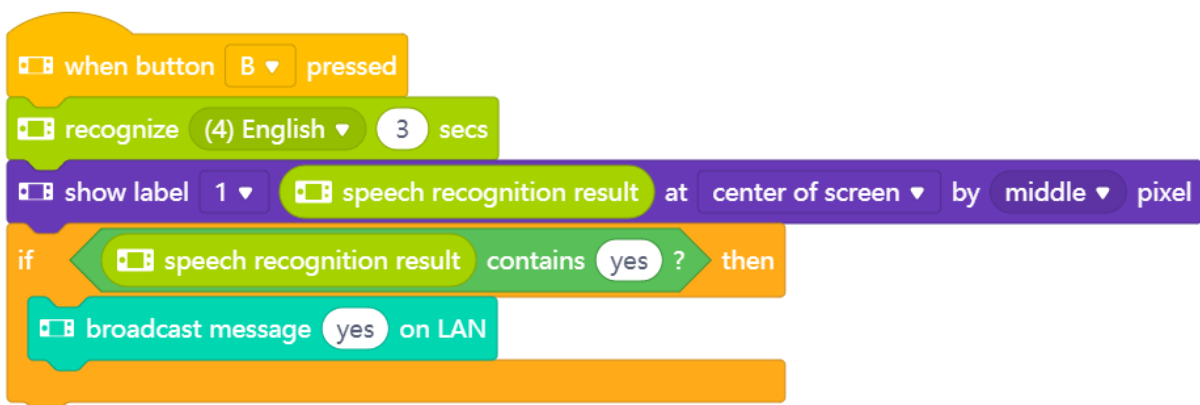
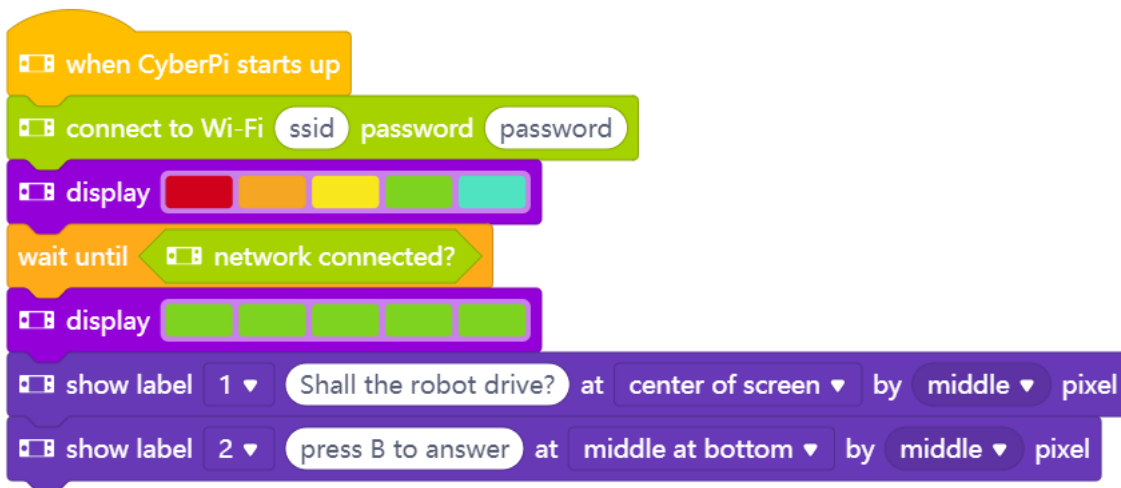
Diese Blöcke arbeiten identisch mit denen, die die LAN-Kommunikation abwickeln:



Um sicherzustellen, dass die Remote-Seite nur mit den gewünschten Geräten kommunizieren kann, benötigen Sie ein Makeblock-Benutzerkonto. Dadurch wird ein "Schlüssel" für die Cloud-Kommunikation erstellt - dieser Schlüssel ist in den Codierblöcken versteckt und nur für einen

Benutzer bestimmt. Um 2 Remote-Geräte über das Internet kommunizieren zu lassen, müssen beide von demselben Benutzerkonto aus programmiert werden.

Die Cloud-Broadcast-Blöcke haben den Vorteil, dass sie weltweit kommunizieren können, aber sie sind nicht für eine schnelle Kommunikation geeignet - es kann ein paar Sekunden dauern, bis die Nachricht ankommt. Die LAN-Broadcasts hingegen sind schnell, funktionieren aber nur lokal. Unten sehen Sie ein Programmierbeispiel, das das oben Gesagte zusammenfasst. Erstellen Sie das Programm zunächst neu und testen Sie, ob alles mit der Erkennung und der Kommunikation zwischen dem CyberPi und einem anderen mBot2 funktioniert. Für diese Übung müssen Sie den gleichen Programmcode im Upload-Modus auf beide Geräte hochladen.



Nachdem Sie sichergestellt haben, dass alles gut funktioniert, können Sie das Beispiel ändern und erweitern - vielleicht Richtungsbefehle? Denken Sie an Lektion1 und die Übung, um den mBot2 manuell durch ein Labyrinth zu lenken... können Sie das Gleiche mit Sprachbefehlen machen?

3. Ausprobieren [25 Minuten]

Schritt 3: Ausprobieren

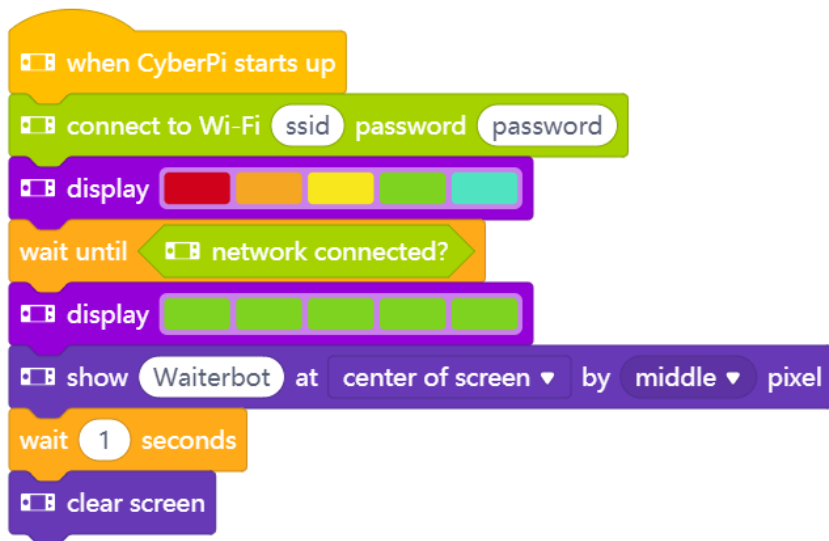
Jetzt wissen Sie, wie Sie eine WIFI-Verbindung herstellen, die Spracherkennung programmieren und Cloud- oder LAN-Messaging einsetzen können. Es ist nun an der Zeit, selbst anzupacken. Sie werden einen höflichen Serviceroboter programmieren - einen Roboterkellner. In einigen Ländern gibt es bereits Restaurants, in denen ein Roboter die Bestellung aufnimmt und das Essen aus der Küche bringt. Schauen Sie sich das folgende Video an:

<https://www.youtube.com/watch?v=FFCPKmLAZb4>

Der Roboter-Kellner, den Sie bauen werden, muss eine Reihe von Dingen können:

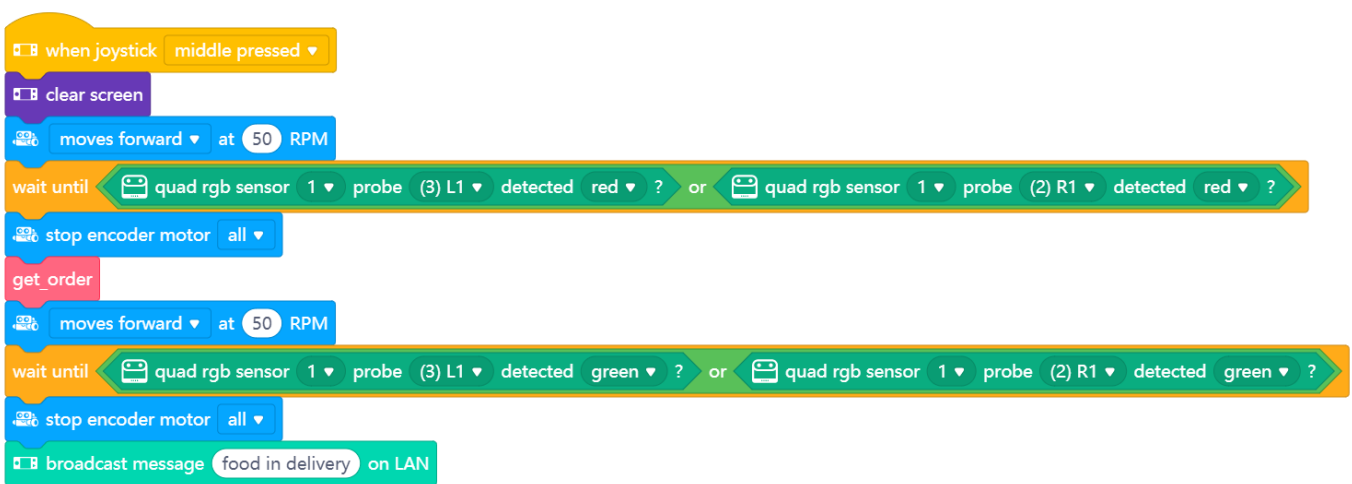
- Entlang einer geraden Linie fahren
- An einem roten Bereich (Ihrem Tisch) anhalten, eine Bestellung aufnehmen und die Bestellung an die Küche senden
- Zusätzliche Option: wenn ein zweiter mBot2 oder CyberPi zur Verfügung steht, dann die Bestellungen an die Küche anzeigen (damit diese bereits mit der Zubereitung der Speisen beginnen kann)
- Weiterfahren, bis der mBot2 auf eine grüne Fläche (die Küche) auftrifft und anhält, um das Essen aus der Küche zu holen
- Optional: wenn ein zweiter mBot2 oder CyberPi verfügbar ist, dann geben Sie eine Benachrichtigung, dass das Essen geliefert wird

Die Startsequenz ist fast identisch mit den vorherigen Beispielen - denken Sie daran, die richtige SSID und das Passwort einzugeben:



Der nächste Codeblock wird im gleichen Skriptfeld unter dem vorherigen Beispiel programmiert. In diesem Codeblock weisen Sie den mBot2 an, loszufahren und dann an einer roten Markierung anzuhalten, um die Bestellung aufzunehmen. Danach fährt der mBot2 weiter zu einer grünen Markierung. Die Farbmarkierungen zeigen den Tisch und die Küche an. Um den Start zu erleichtern, gehen wir davon aus, dass sie in einer geraden Linie liegen. Wenn alles funktioniert, können Sie dies weiter ausbauen und das Wissen aus den vorherigen Lektionen mit einbeziehen und den Roboter dazu bringen, einer Linie zum Tisch zu folgen (die mitgelieferte Karte hat rote und grüne Markierungen auf der Strecke, die zu diesem Zweck verwendet werden können). In Lektion 5 wird alles über das Folgen einer Linie erklärt...

Als zusätzliche Ergänzung zur Spracherkennung, um bei der geraden Linie zwischen Tisch und Küche zu bleiben, können Sie die Bestellung per Cloud- oder LAN-Messaging an einen anderen mBot2 oder CyberPi zur Anzeige senden.



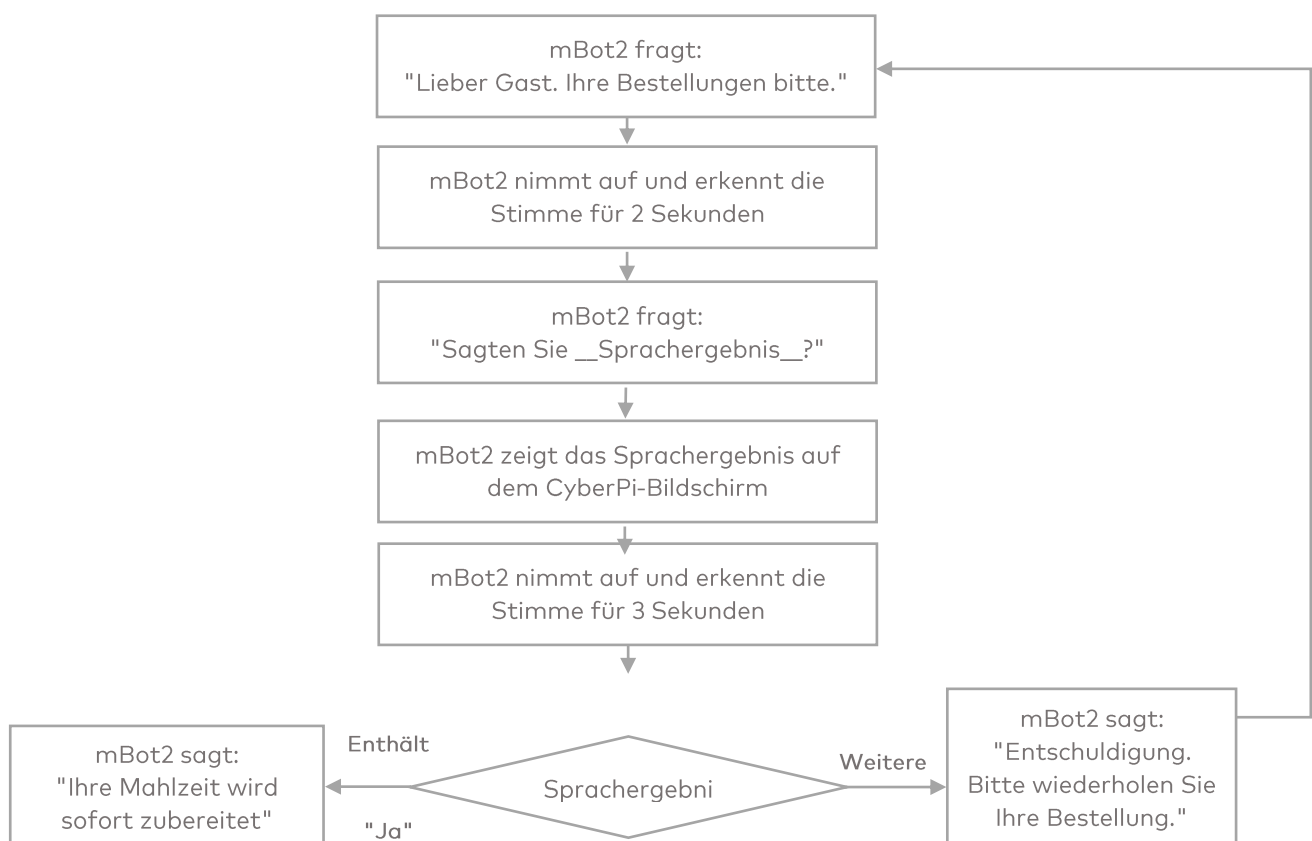
Dazu müssen Sie selbst den Block "get_order" in "Meine Blöcke" erstellen. Das hilft, den Überblick über das Hauptprogramm zu behalten - es geht also nur darum, farbige Marker zu fahren und dazwischen die Bestellung zu bekommen.

Um zu definieren, was "get_order" eigentlich bedeutet, lassen Sie uns zunächst an einen Bestellvorgang denken:

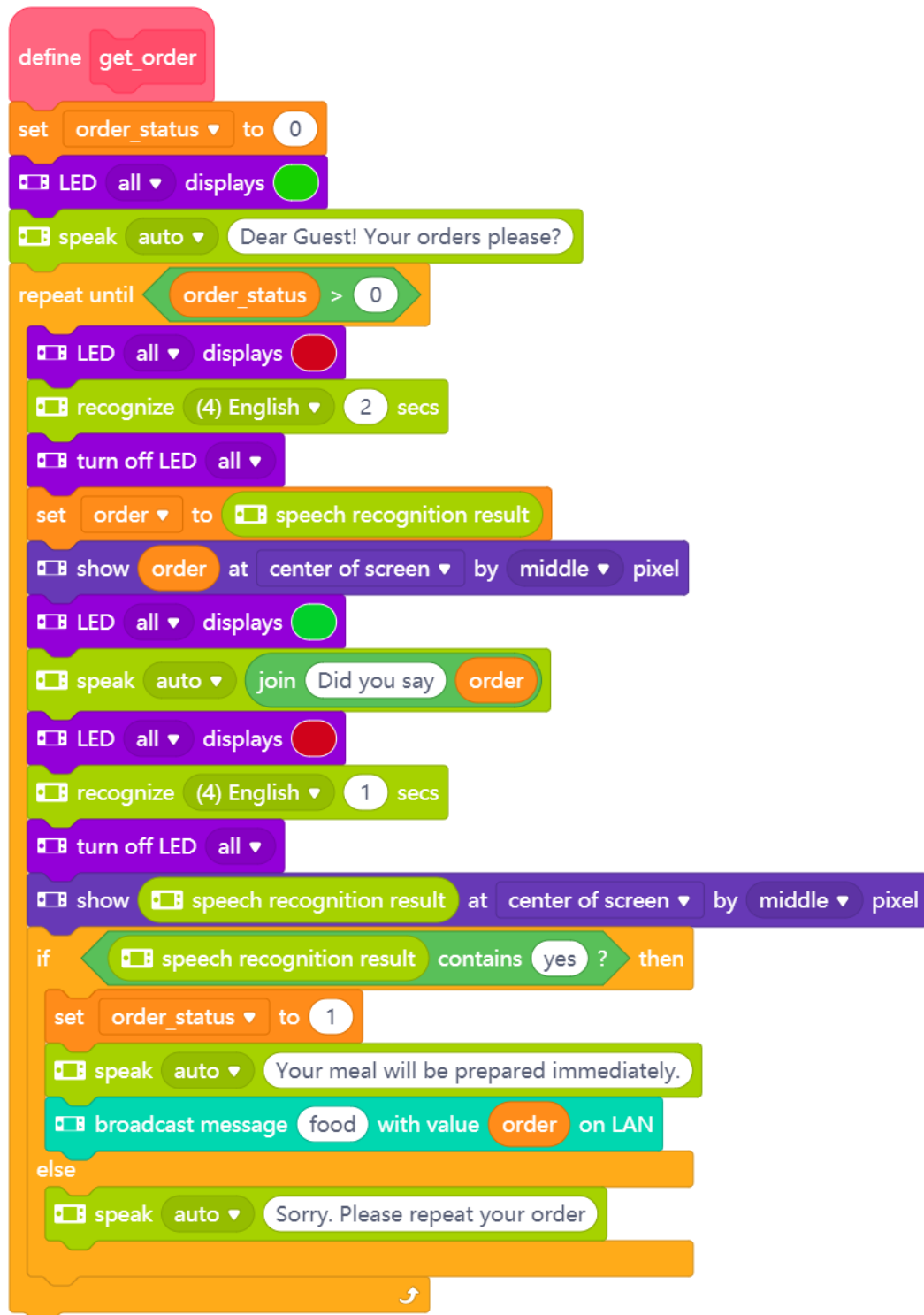
1. Nach der Bestellung fragen
2. Die Antwort erkennen
3. Zurückprüfen, ob die erkannte Bestellung korrekt ist
4. Wenn ja, fortfahren (Bestellung an die Küche schicken und die Bestellsequenz beenden)

Wenn nein, erneut fragen.

Die Überprüfung ist eine sinnvolle Sache, um Fehlbestellungen zu vermeiden. Um diese Schritte strukturiert zu übersetzen, können Sie ein Flussdiagramm wie das folgende verwenden:



Von hier aus ist es viel einfacher, die notwendigen Codierblöcke zu erstellen: Sie beginnen mit dem Codeblock "Bestellung holen (get order)". Diesen Codeblock legen Sie wiederum in „Meine Blöcke“ an.



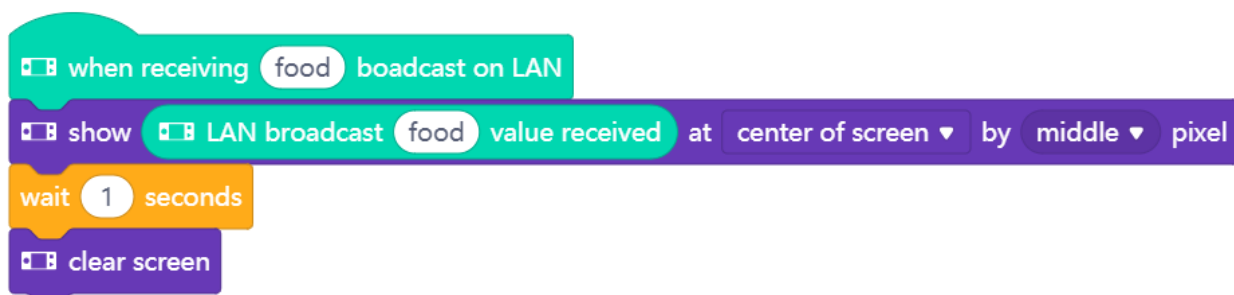
Außerdem müssen Sie zwei Variablen anlegen. In der einen Variablen wird das Erkennungsergebnis der Bestellung gespeichert, so dass dieses, wenn alles richtig verstanden wurde, an die Küche gesendet werden kann (Achtung: Rückfrage mit ja/nein - die Antwort darauf würde das vorherige Erkennungsergebnis im Report-Block überschreiben; deshalb müssen wir die erkannte Bestellung in einer Variablen speichern).

Die zweite Variable verfolgt den gesamten Prozess: Ist die Bestellung vollständig (also richtig verstanden)? Wenn nicht, beginnt der Bestellvorgang wieder von vorne. Diese zweite Verwendung der Variable berichtet über Prozessdaten (Stadium).

Sie können das Wissen aus den vorangegangenen Lektionen zur Unterstützung des Programms verwenden, z. B. LEDs aufleuchten lassen, um dem Benutzer zu verdeutlichen, dass der Roboter gerade spricht (grün) oder zuhört (rot), oder Sie können den Bildschirm des Kellner-mBot2 verwenden, um Daten anzuzeigen. Verwenden Sie dies an Kontrollpunkten in Ihrem Programm, um zu überprüfen, ob Ihr Code funktioniert, und um den Benutzern ein Feedback zu geben.

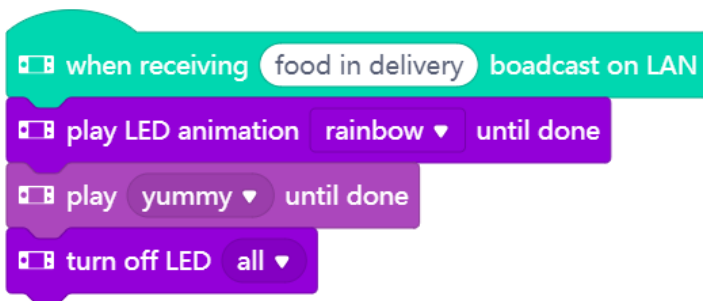
Als zusätzliche Ergänzung können Sie Cloud- oder LAN-Messaging verwenden, um Daten an einen anderen mBot2 zu senden. Für eine bessere Lernerfahrung gibt es in dieser Lektion Beispiele, wie Sie bestimmte Aufgaben lösen können. Sie können sich auf diese beziehen, wenn Sie Probleme mit Ihrem eigenen Code haben oder sich Ideen holen, wie Sie mit einer bestimmten Situation umgehen können. Im Allgemeinen ist es besser, einen Entwurf zum Laufen zu bringen und den Code in Iterationen zu verfeinern, als zu versuchen, alles auf einmal fertig und perfekt programmiert zu haben. Beginnen Sie mit der Fahrsequenz und anstatt den "get_order"-Block parat zu haben, machen Sie dort einfach einen Piepton, um zu zeigen, dass alles so weit funktioniert. Dann arbeiten Sie am Bestellvorgang, lassen aber das Fahren außen vor. Auf diese Weise können Sie schneller iterieren, ohne dass der Roboter jedes Mal zwischen den Markierungen hin- und herfahren muss, wenn Sie Ihren Code ausprobieren wollen. Als letzte Schritte kombinieren Sie sie, um die gegebene Aufgabe zu lösen.

In den obigen Programmierbeispielen haben Sie als zusätzliche Option eine Nachricht an einen zweiten mBot2 gesendet. Zum Beispiel haben Sie im "Bestellvorgang" gesagt, dass der mBot2, sobald der Kunde die Bestellung bestätigt, eine Nachricht über die Cloud oder LAN-Messaging an einen zweiten mBot2 oder CyberPi (zur Anzeige der eingehenden Bestellungen in der Küche) sendet. Nun müssen Sie den zweiten mBot2 so programmieren, dass er die Nachricht empfängt.

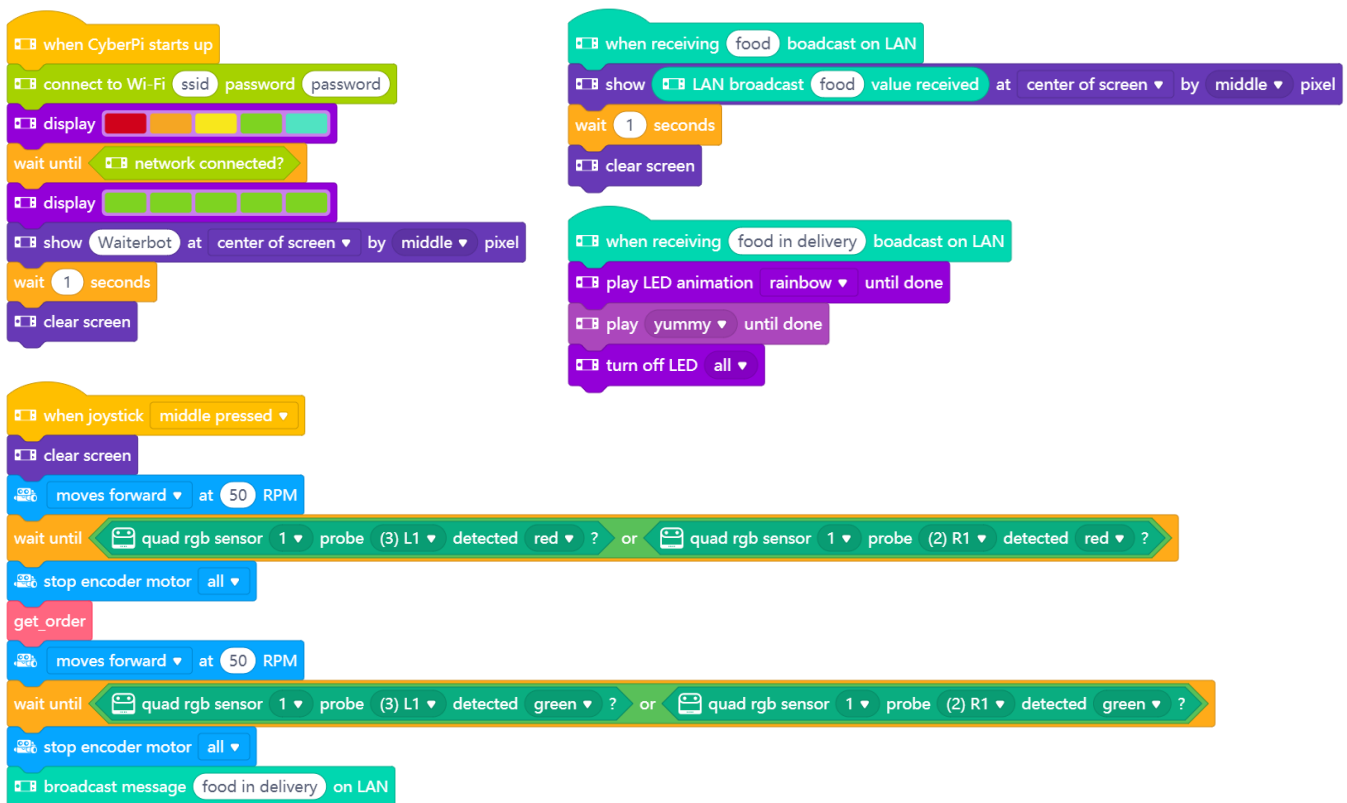


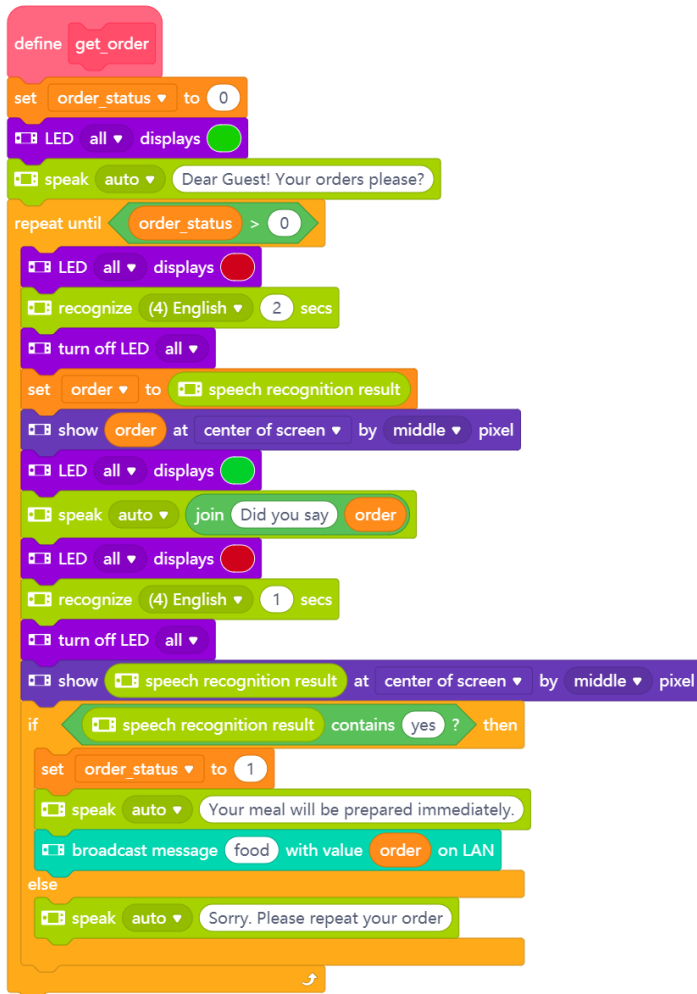
Im folgenden Programmierbeispiel empfängt die Küche die Bestellung und zeigt sie an. Erstellen Sie das folgende Programmierbeispiel im gleichen Skript. Legen Sie dieses neben das erste Programmierbeispiel. Auf diese Weise halten Sie das Skript übersichtlich.

Im Programmierbeispiel "Steuerung eines Roboters" haben Sie programmiert, dass nach dem Erkennen des grünen Bereichs eine Cloud-Nachricht an den Kunden gesendet wird, um ihn zu informieren, dass seine Bestellung kommt. Auch dies ist im gleichen Skriptfeld neben den anderen Blöcken programmiert - Sie könnten einen weiteren CyberPi (als Smart Device auf dem Tisch des Kunden) verwenden, um über den Bestellstatus zu informieren:



Wenn Sie alles programmiert haben, sieht Ihr Skriptfeld so aus:





Sie müssen diesen Code auf alle mBot2s und/oder CyberPi's hochladen, die Sie verwenden möchten. Es kann derselbe Code sein, da sie automatisch über die LAN- oder Cloud-Verbindung verbunden sind - und derjenige, der die Nachricht sendet, wird nicht darauf reagieren.

Zeichnen Sie eine gerade Linie, auf der der mBot2 fahren soll, und stellen Sie sicher, dass eine breite rote und grüne Markierung vorhanden ist. Laden Sie den Code hoch und testen Sie, ob der Roboter-Kellner die Bestellungen der Kunden bearbeitet.

Natürlich gibt es immer Raum für Verbesserungen. Passen Sie den Code dort an, wo Sie denken, dass er besser oder ansprechender sein könnte. Vielleicht beginnen Sie mit einem Linienfolger, der die Bestellungen entgegennehmen kann? Oder Sie erweitern diese Übung weiter und haben einen eigenen Kellner mBot2, der die Bestellungen entgegennimmt und einen Lieferservice mBot2, der das Essen bringt? Sie können mit anderen Teams in Ihrer Klasse zusammenarbeiten und deren Roboter integrieren...

Diskutieren Sie über mögliche Erweiterungen wie diese in der Klasse, wenn Sie den bisherigen Fortschritt besprechen, und erarbeiten Sie einen Anwendungsfall mit mehreren Robotern - es muss kein Restaurant sein...

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Haben Sie es geschafft, den Kellner-Bot zu programmieren?

In dieser Lektion haben Sie etwas über Spracherkennung und Sprachsynthese gelernt und wie Sie diese über einen Cloud-Service anwenden können. Sie haben es geschafft, den mBot2 mit dem Internet zu verbinden und so Daten lokal oder sogar aus der Ferne zu senden.

Das Erstellen eines Dialogs (z. B. für die Bestellung eines Gerichts) war Teil der Übung, und Sie haben gelernt, die Textantwort nach bestimmten Auslösewörtern zu filtern, um ein "natürlicheres" Gespräch zu führen (und nicht zu erwarten, dass der Benutzer nur mit bestimmten Antworten antwortet).

Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was hat Ihrer Meinung nach gut funktioniert?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?

Lektion 9: mBot2 in freier Wildbahn

Thema: STEAM

Klasse(n): 5. Klasse und aufwärts

Dauer: 45 Minuten

Schwierigkeitsgrad: Anfänger

★ Lektionsziele

Am Ende dieser Lektion werden die Schüler in der Lage sein:

- Maschinelles Lernen in die Programmierung des mBot2 zu integrieren
- Mehrere technische Funktionen des mBot2 zusammenarbeiten zu lassen
- Eine Kommunikation zwischen dem Roboter und dem Computer herzustellen, obwohl der Roboter unabhängig im Upload-Modus für die Interaktion zwischen physischer Hardware (Roboter) und Sprite-Stadium der mBlock-Software läuft

★ Überblick

Maschinelles Lernen ist eine schwache Form der Künstlichen Intelligenz (KI). Künstliche Intelligenz bezieht sich auf Geräte oder Maschinen, die die menschliche Intelligenz nachahmen. Basierend auf gesammelten Daten ermöglicht maschinelles Lernen Computer- oder Robotersystemen, Aktionen auf der Grundlage der Datenhistorie und der Wahrscheinlichkeit zu wählen, ohne dass dafür explizit ein Algorithmus formuliert werden muss.

🔗 Schwerpunkt

Am Ende der Lektion werden die Schüler wissen:

- Was maschinelles Lernen ist und wie man es anwendet,
- Wie man mBot2 abhängig von den Bedingungen arbeiten lässt,
- Wie die verschiedenen technischen Funktionen von mBot2 in einem Ökosystem zusammenarbeiten,
- Wie man ein Kommunikationsprotokoll zwischen dem Roboter, der selbstständig Programme ausführt, und dem Computer einrichtet.

Checkliste vor der Lektion

Was brauchen Sie?

- PC oder Laptop (mit USB-Anschluss) mit installierter mBlock-Software, der Web-Version (auch für ChromeBook)
- Den mBot2 mit einem CyberPi
- Ein USB-C-Kabel oder Makeblock Bluetooth-Dongle
- Webcam, an Ihrem Laptop oder extern
- Drucker (optional) oder Papier, Stifte und Marker zum Zeichnen oder Smartphone zum Anzeigen von Bildern

Lehrplan

Diese Lektion besteht aus vier Schritten und dauert insgesamt 45 Minuten.

Dauer	Inhalt
5 Minuten	1. Aufwärmen <ul style="list-style-type: none">• Maschinelles Lernen im Alltag• Maschinelles Lernen kennenlernen
25 Minuten	2. Praxisorientiert <ul style="list-style-type: none">• Maschinelles Lernen anwenden• Maschinelles Lernen in Kombination mit dem mBot2 in mBlock• Einen Kommunikationsprotokoll von innerhalb laufender Programme aufbauen
10 Minuten	3. Ausprobieren <ul style="list-style-type: none">• Letzte Aufgabe: das Ökosystem einer Maus
5 Minuten	4. Zusammenfassung <ul style="list-style-type: none">• Showtime: zeigen Sie Ihr Ökosystem in einem lustigen, kurzen Film zwecks späterer Diskussion.• Wenn Ihr Lehrer es erlaubt, teilen Sie das Endergebnis auf sozialen Medien mit dem Hashtag #mBot2inclass• Reflexion: Worauf sind Sie am meisten stolz? Was würdet ihr gerne an eurem Roboter verbessern?

1. Aufwärmen (5 Min.)

Schritt 1: Aufwärmen

Dieser Schritt besteht aus drei Teilabschnitten:

1. Maschinelles Lernen im Alltag
2. Die Maschinelles Lernen-Erweiterung in mBlock kennenlernen
3. Maschinelles Lernen anwenden

1. Maschinelles Lernen im Alltag

Maschinelles Lernen funktioniert auf statistischer Basis: Dem Algorithmus werden Trainingsdaten zur Verfügung gestellt, wie z. B. Bilder von verschiedenen Objekten plus deren Beschreibungen. Die Lernphase versucht dann, diese Bilder bestmöglich zu unterscheiden und jedes Bild der richtigen Kategorie zuzuordnen. Nach dieser Phase werden die gelernten Muster gegen neue Datensätze getestet, und das Gelernte kann vertieft werden. Der Unterschied zu "klassischen" Sortierprogrammen besteht darin, dass es im Vorfeld keine klaren Hinweise gibt, wie zwischen den verschiedenen Kategorien zu entscheiden ist, so wie man zwischen Farben unterscheiden kann.

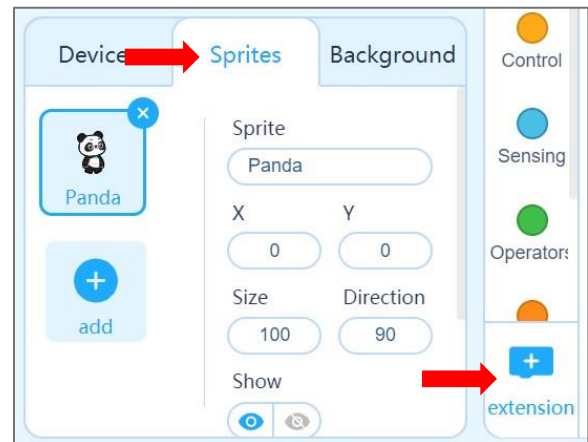
Durch das "Lernen aus vergangenen Daten" kann sich das Computerprogramm besser an veränderte Situationen anpassen. Denken Sie an die Gesichtserkennung auf Ihrem Telefon: die ersten Male nach der Einrichtung ist es für das Telefon schwieriger, Sie korrekt zu identifizieren, aber mit der Zeit "lernt" der Algorithmus, Ihr Gesicht auch in unterschiedlichen Umgebungen zu erkennen. Maschinelles Lernen wird in großem Umfang bei der Bildklassifizierung (z. B. in sozialen Medien) eingesetzt, um automatisch zu analysieren, welche hochgeladenen Bilder eine Katze, einen Hund, Möbel, Menschen usw. zeigen.

Maschinelles Lernen wird als "schwache künstliche Intelligenz" betrachtet, da das Computerprogramm sich seiner selbst nicht bewusst ist und auch nicht "denken" kann, wie wir es tun - aber sein aktuelles Verhalten wird durch gesammelte und analysierte Daten beeinflusst.

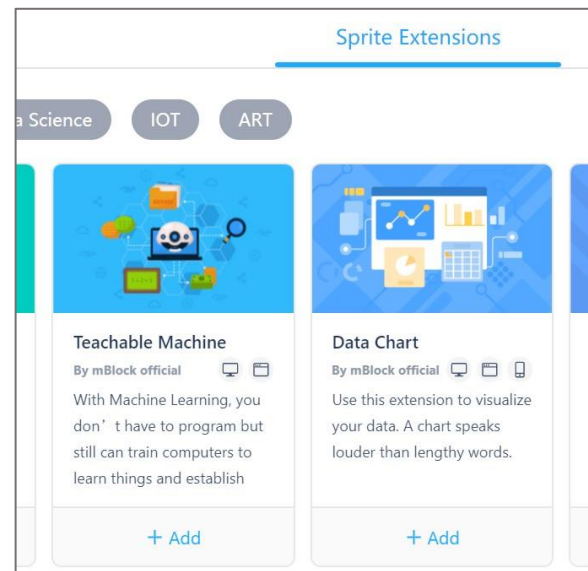
2. Die Maschinelles Lernen-Erweiterung in mBlock kennenlernen

In mBlock gibt es ein Tool für maschinelles Lernen. Sie müssen dieses Tool aus der Erweiterungsbibliothek hinzufügen. Gehen Sie wie folgt vor:

- Klicken Sie auf "Sprites".
- Klicken Sie auf "Erweiterungen"
(unten in der Mitte des
Bildschirms).



- Sie gelangen nun in die
Erweiterungsbibliothek. Klicken Sie
oben auf dem Bildschirm auf
"Sprite-Erweiterungen".
- Suchen Sie die Erweiterung
"Teachable Machine" und fügen Sie
sie hinzu.



- Die Erweiterung wird nun zu den
Codeblöcken hinzugefügt. Sie
erkennen die Erweiterung an der
Abkürzung TM in grün.
- Stellen Sie sicher, dass Sie eine
funktionierende Webcam haben.
- Klicken Sie auf die Schaltfläche "Trainingsmodell".



Training model

Die Erweiterung besteht aus einem Trainingsmodell, in dem Sie die Unterscheidung von mit einer Webcam aufgenommenen Bildsätzen "lernen" können. Das Grundmodell, das Sie sehen, besteht aus 3 Kategorien. Wenn Sie mehr wollen, klicken Sie auf "Neues Modell erstellen" und wählen Sie die Anzahl der Kategorien.

Nützlicher Tipp:

- Denken Sie vorher über die Anzahl der Kategorien und die Erkennung nach. Sie können später keine weitere Kategorie hinzufügen, Sie müssten wieder von vorne anfangen.
- Wählen Sie 1 zusätzliche Kategorie, die als "Hintergrund" verwendet werden soll. Wenn Sie also 2 verschiedene Tiere anhand von Bildern, die Sie vor die Webcam halten, erkennen wollen, wählen Sie 3 Kategorien: Die dritte wird die sein, in der kein Bild eines Tieres gezeigt wird. Ansonsten "zwingen" Sie das Modell, sich z. B. zwischen Katze und Hund zu entscheiden, auch wenn die Webcam den leeren Hintergrund zeigt.
- Der Hintergrund sollte generell einfach und statisch sein (nicht: Klassenzimmer mit Schülern, Webcam schaut auf die Schüler usw.) und einen deutlichen Kontrast zu den zu erkennenden Objekten aufweisen.

Die Bilder werden nur am Computer ausgewertet und das Ergebnis kann gespeichert werden. Die Bilder selbst werden nicht aufbewahrt! Wenn Sie also die Programmdatei in mBlock speichern und später wieder öffnen, sehen Sie nicht die Originalbilder, an denen das Modell trainiert wurde. Sie können immer neue Bilder zu bestehenden Kategorien hinzufügen, aber keine einzelnen löschen (nur alle Bilder und Learnings einer Kategorie).

Sie können die Kategorien direkt mit Bildern "anlernen", die Sie mit der Kamera aufgenommen haben. Für diese Übung werden Sie die standardmäßigen 3 Kategorien verwenden:

1. Sie können mit dem mBot2 beginnen. Legen Sie ihn auf den Tisch und richten Sie die Kamera so aus, dass Sie einen guten Blick darauf haben - oder halten Sie ihn fest in der Hand, wenn Sie einen Laptop mit eingebauter Kamera verwenden. Klicken Sie dann auf die Schaltfläche "Lernen". Halten Sie die Taste gedrückt, bis die Kamera mindestens 20 Probeaufnahmen aufgenommen hat. Oberhalb des Mosaiks der Proben sehen Sie die Anzahl der für diese Kategorie gespeicherten Proben. Geben Sie dieser Kategorie einen Namen, z. B. "mBot2".
2. Wählen Sie nun etwas anderes, z. B. ein Notebook, halten Sie es vor die Kamera und klicken Sie auf die Schaltfläche "Lernen". Halten Sie die Taste gedrückt, bis die Kamera

mindestens 20 Probeaufnahmen aufgenommen hat. Geben Sie dieser Kategorie einen Namen, z. B. "Notizbuch".

3. Nehmen Sie nun den "Hintergrund" auf. Dieser sollte immer dann verwendet werden, wenn keine weiteren Objekte vor der Kamera zu sehen sind. Klicken Sie auf die Schaltfläche "Lernen" und halten Sie sie gedrückt, bis die Kamera mindestens 20 Probeaufnahmen aufgenommen hat. Geben Sie dieser Kategorie einen anderen Namen, z. B. "Hintergrund".

The screenshot displays a machine learning interface with three categories, each represented by a card. Each card includes a sample image grid, a name input field, a progress bar, and a 'Learn' button.

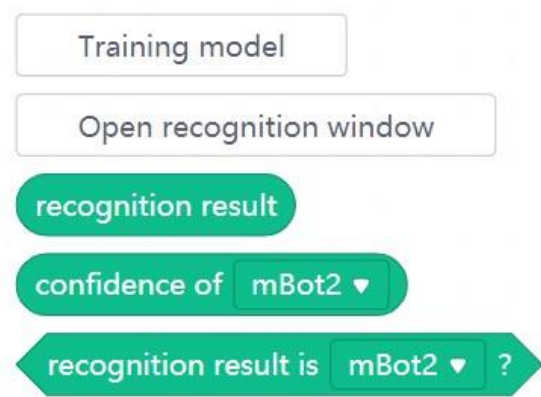
- Category 1:** 36 Samples, name 'mBot2'. The sample image grid shows a blue robot (mBot2) with a white 'X' in the top right corner. The progress bar is empty.
- Category 2:** 41 Samples, name 'notebook'. The sample image grid shows a notebook with a white 'X' in the top right corner. The progress bar is 40.0% full.
- Category 3:** 20 Samples, name 'nothing'. The sample image grid shows a dark background with a white 'X' in the top right corner. The progress bar is empty.

Sie haben dem Maschinellen Lernen nun verschiedene Objekte (mBot2, Notebook und Hintergrund) "angelernt", die der Algorithmus nun im Kamerabild erkennen wird. Für jede Kategorie sehen Sie einen Balken mit einer Prozentangabe. Dieser stellt das Vertrauen dar, das der Algorithmus in das Erkennungsergebnis hat. Sie können den Prozentsatz des Vertrauens erhöhen, indem Sie die Anzahl der Probeaufnahmen erhöhen und einige leichte

Variationen durchführen, wie z. B. den Winkel des Objekts ändern. Dies hilft der Kamera, die Objekte noch besser und schneller zu erkennen.

Sie können die Erkennung live testen, ohne dass Sie zu diesem Zeitpunkt eine Codierung vornehmen müssen. Fügen Sie so lange Bilder hinzu, bis die Erkennung schnell und genau ist (nicht zwischen den Ergebnissen wechselt und einen hohen Prozentsatz > 90 % anzeigt). Sind Sie zufrieden? Dann klicken Sie auf "Das Modell verwenden". Sie kehren nun zum Bildschirm für die Codierung auf Sprite-Ebene zurück.

Es gibt jetzt drei neue Codeblöcke. Ein Block meldet das Erkennungsergebnis (so wie Sie es eingegeben haben), der zweite ist die Vertrauenswürdigkeit für eine bestimmte Kategorie und der dritte ist der boolesche Block für bedingte Anweisungen. Hier können Sie die Kategorie über ein Dropdown-Menü auswählen. Wenn Sie auf einen der Codeblöcke klicken, wird ein neues Fenster angezeigt. Dies ist das Erkennungsfenster, in dem das Kamerabild und das Erkennungsergebnis angezeigt werden. Jetzt funktioniert Ihre Webcam als intelligenter Sensor, der verschiedene Objekte oder Bilder erkennt.



2. Praxisorientiert (25 Min.)

Schritt 2: Praxisorientiert

Dieser Schritt besteht aus drei Teilabschnitten:

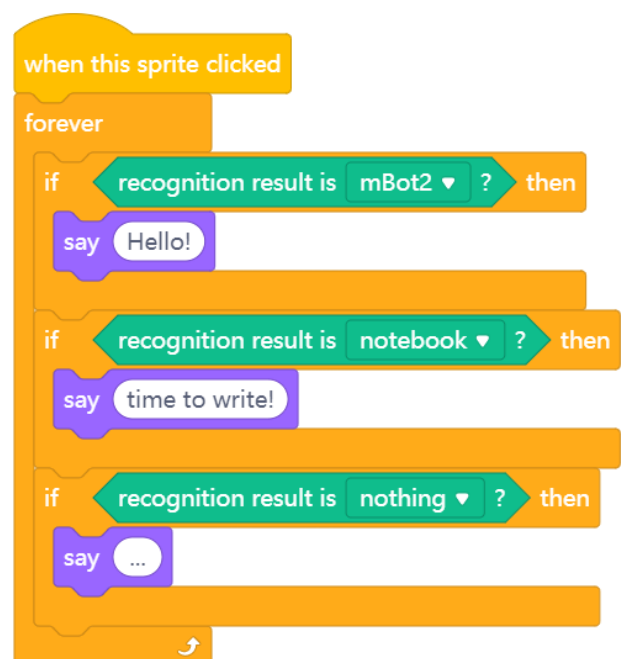
1. Maschinelles Lernen anwenden
2. Maschinelles Lernen in Kombination mit dem mBot2
3. Eine LAN-Verbindung herstellen

1. Maschinelles Lernen anwenden

Sie haben zwei verschiedene Objekte, den mBot2 und ein Notizbuch, sowie den Hintergrund des Machine Learning-Modells "eingelernt". Nun können Sie den Output der Erkennung wie jeden anderen Sensor-Block in Ihrem Programmcode verwenden. Sie können bedingte Anweisungen verwenden, um zwischen den Objekten zu unterscheiden und jedem Ergebnis separat eine bestimmte Aktion zuzuweisen.

Sie werden nun versuchen, das Sprite (in diesem Fall den Panda) zum Sprechen zu bringen, wenn die Webcam ein gelerntes Objekt erkennt.

Auf der rechten Seite sehen Sie ein Programmierbeispiel. Reproduzieren Sie das Programmierbeispiel und schauen Sie, was der Panda macht, wenn die Webcam Ihre gelernten Objekte erkennt. War er erfolgreich? Dann erforschen Sie, welche anderen Möglichkeiten es gibt. Sind Sie in der Lage, den Panda springen oder laufen zu lassen?

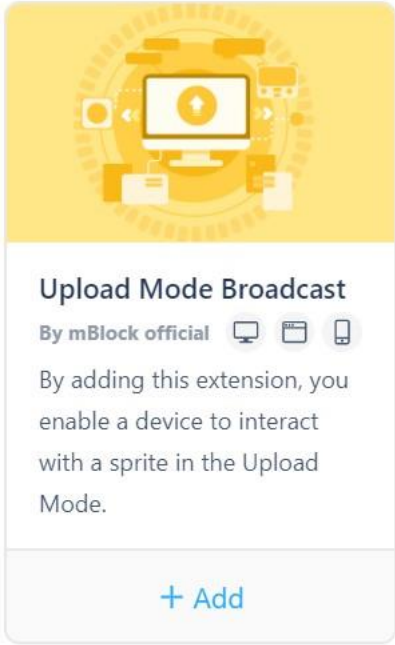


2. Maschinelles Lernen in Kombination mit dem mBot2

Sie wissen nun, wie Sie die Erweiterung für maschinelles Lernen zur Bilderkennung nutzen und damit ein Sprite, z. B. den Panda, steuern können. Sie werden nun lernen, wie Sie vom Computer aus mit dem mBot2 kommunizieren, der sein eigenes Programm ausführt. Dazu muss der mBot2 mit dem Computer "verbunden" bleiben - dies geschieht über ein USB-Kabel oder den Makeblock Bluetooth-Dongle. Der Dongle erlaubt es dem Roboter, sich unabhängig vom Computer zu bewegen; wenn Sie ein USB-Kabel verwenden, stellen Sie sicher, dass Sie nur kleine Bewegungen mit dem mBot2 durchführen, wie kleine Drehungen und Strecken.

Wenn diese Verbindung hergestellt ist, kann das Sprite (in diesem Fall der Panda) Daten an den mBot2 senden. Der mBot2 empfängt diese Daten und das auf dem Roboter laufende Programm kann darauf reagieren.

Sie müssen eine neue Erweiterung zu den Codeblöcken für das Sprite und den mBot2 hinzufügen, die "Hochlade-Modus Broadcast" genannt wird.

	Erweiterung	Hinzufügen
Für das Sprite		<ul style="list-style-type: none"> • Klicken Sie auf „Sprites“ • Klicken Sie auf „Erweiterungen“ • Klicken Sie auf „Sprite-Erweiterungen“ • Suchen Sie die Erweiterung mit der Bezeichnung "Hochlade-Modus Broadcast" und fügen Sie sie hinzu
Für mBot2		<ul style="list-style-type: none"> • Klicken Sie auf "Geräte" • Klicken Sie auf „Erweiterungen“ • Klicken Sie auf "Geräteerweiterungen" • Suchen Sie die Erweiterung mit der Bezeichnung "Hochlade-Modus Broadcast" und fügen Sie sie hinzu (sie befindet sich möglicherweise nicht auf der ersten Seite der Erweiterungsbibliothek)

Auf diese Weise wird die Erweiterung sowohl für den mBot2 als auch für das Sprite hinzugefügt. In mBlock erscheint sie in dunkelblau. Klicken Sie auf die neue Erweiterung und sehen Sie die verschiedenen Codeblöcke, die mit ihr verbunden sind. Im Grunde fügt die Erweiterung ein Kommunikationsprotokoll hinzu. Es ermöglicht Ihnen, Nachrichten zwischen dem Sprite und dem Roboter zu senden und zu empfangen, während sich der Roboter im Upload-Modus befindet.

3. Die Verbindung mit dem mBot2 herstellen

Sie haben nun alle notwendigen Erweiterungen installiert, damit die Hard- und Software miteinander kommunizieren können. Nun werden Sie den mBot2 so programmieren, dass er

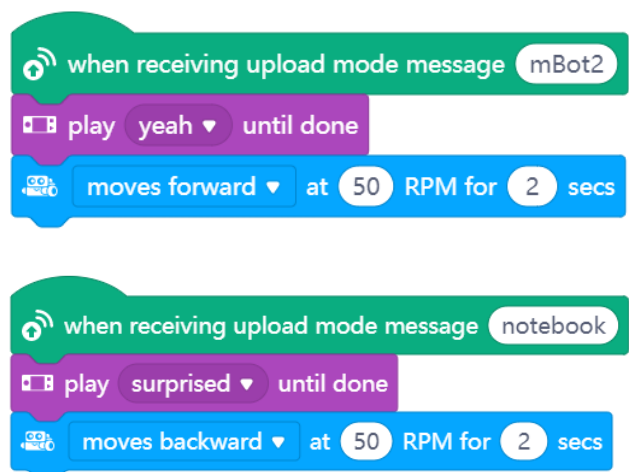
durch maschinelles Lernen gesteuert wird. Die Idee ist, dass der mBot2 Aktionen aufgrund von Erkennungsergebnissen ausführt. Eine ähnliche Aufgabe haben Sie zuvor nur für das Sprite (mBot2, Notizbuch und Hintergrund) programmiert.

Nun müssen die Erkennungsergebnisse über die Verbindung zwischen dem mBot2 und dem Computer (entweder per Kabel oder drahtlos per Bluetooth-Dongle) gesendet und empfangen werden. Der Empfangscode verwendet Ereignisse, so dass der entsprechende Code ausgeführt wird, sobald die Nachricht empfangen wird. Mit Ereignissen muss nicht in einer Schleife nach einem Sensor oder einer Nachricht gesucht werden, was den Code effizienter und leichter verständlich macht.

Sie müssen zwei Computerprogramme erstellen - eines für das Gerät (mBot2) und eines für das Sprite, das das Add-on "Machine Learning" verwendet:

Sprite-Programm

mBot2-Programm



Stellen Sie nach der Neuprogrammierung der obigen Codeblöcke sicher, dass Ihr Computer mit dem mBot2 verbunden bleibt, wenn Sie ein Kabel verwenden (trennen Sie das Kabel nicht ab). Idealerweise sind sie über Bluetooth mit dem Makeblock Bluetooth-Dongle verbunden, da dies dem mBot2 mehr Flexibilität beim Herumfahren gibt.

Wenn Sie den Dongle verwenden:



Installieren Sie den Dongle auf dem Computer und schalten Sie den mBot2 ein. Sie müssen das mBot2 mit dem Bluetooth-Dongle koppeln. Dies tun Sie, indem Sie auf die entsprechende Schaltfläche auf dem Dongle klicken. Die LED beginnt zu blinken und der Dongle versucht, sich mit dem nächstgelegenen Makeblock-Controller (wie mBot2 und andere) zu verbinden. In einer Klassenzimmersituation sollten Sie sicherstellen, dass dies teamweise geschieht, um die entsprechende Kopplung zwischen Dongle und mBot2 zu gewährleisten. Sobald die Kopplung erfolgt ist, werden Sie einen Piepton vom mBot2 hören. Das bedeutet, dass der Computer und der mBot2 eine drahtlose Verbindung hergestellt haben.

Der Makeblock-Bluetooth-Dongle merkt sich das zuletzt gekoppelte Gerät und stellt die Verbindung sofort wieder her.

Falls Sie den Bluetooth-Dongle von Makeblock nicht haben, können Sie die Kompatibilität des Bluetooth in Ihrem Computer über den folgenden Link überprüfen:

<https://www.yuque.com/makeblock-help-center-en/cyberpi/bluetooth-compatibility>

Schalten Sie nun mBlock in den Upload-Modus und verbinden Sie den mBot2 mit mBlock.

Klicken Sie auf die Taste "Verbinden" und laden Sie das Computerprogramm hoch. Sie können nun mit dem Testen beginnen!

3. Ausprobieren (25 Min.)

Schritt 3: Ausprobieren

Sie haben das Wissen und die Fähigkeiten, um den mBot2 in jeder Situation einzusetzen. Jetzt werden Sie mit dem Ökosystem einer kleinen Maus arbeiten. Eine Maus lebt in der Natur, wo sie nach Nahrung und Wasser suchen muss. Es gibt auch viele Gefahren, wie zum Beispiel Raubtiere, die sich von Mäusen ernähren.

In dieser Aufgabe werden Sie den mBot2 so programmieren, dass er sich wie eine kleine Maus verhält, die nach Nahrung sucht und vor Raubtieren flieht.

Suchen Sie ein Bild eines fliegenden Adlers und etwas Getreide. Das Getreide steht für das Futter der kleinen Maus. Der Adler ist der Jäger. Sie können diese Bilder ausdrucken oder mit



Ihrem Smartphone auf die Webcam übertragen. Sie können die Bilder für Futter und Jäger auch von Hand zeichnen, wenn Sie möchten. Verwenden Sie einfach die gleiche Methode, die Sie für das "Anlernen" der Bilder an Ihren Algorithmus mit der maschinellen Lernerweiterung und für das Testen des Programms gewählt haben.

Programmieren Sie den mBot2 so, dass sich der mBot2 vorwärts bewegt, wenn Sie der Kamera das Getreide zeigen (wenn Sie das Kabel verwenden, machen Sie eine kleine Bewegung), aber wenn Sie das Bild des Adlers zeigen, bewegt er sich rückwärts.

Sie können auch noch mehr Codes hinzufügen: Wenn die kleine Maus das Futter sieht, sagt sie "Lecker" und wenn sie den Adler sieht, können wir ihren Herzschlag hören.

Beim Erstellen des Computerprogramms für diese Aufgabe ist es hilfreich, den folgenden Schritt-für-Schritt-Plan zu verwenden.

	Erklärung
Schritt 1: Was möchten Sie tun?	<ul style="list-style-type: none">• Was wollen Sie programmieren?
Schritt 2: Was brauchen Sie?	<ul style="list-style-type: none">• Was benötigen Sie zusätzlich zum mBot2?
Schritt 3: Welche Codeblöcke benötigen Sie dafür?	<ul style="list-style-type: none">• Wie bringen Sie den mBot2 dazu, sich zu bewegen?• Welche Codeblöcke werden Sie verwenden?• Beschreiben Sie kurz, wie Ihr Programm funktioniert (Pseudocode/Natürliche Sprache, Flussdiagramm oder UML)• Wenn Sie weitere Erklärungen benötigen, können Sie mit Ihren Mitschülern oder dem Lehrer diskutieren oder eine Recherche zu dem Thema durchführen. Zu jedem Codierblock in mBlock gibt es auch eine Hilfe.

Schritt 4: Test und Implementierung	<ul style="list-style-type: none"> • Ist die erste Version schon einsatzbereit? Testen Sie sie! Schreiben Sie während der Testrunde Verbesserungspunkte auf • Arbeiten Sie an den Verbesserungspunkten, bis der mBot2 genau das tut, was Sie sich vorgestellt haben. • Ist es Ihnen gelungen? Filmen Sie das Endergebnis und fragen Sie Ihren Lehrer, ob Sie es auf sozialen Medien mit dem Hashtag #mBot2inclass posten dürfen
-------------------------------------	--

4. Zusammenfassung (5 Min.)

Schritt 4: Zusammenfassung

Konnten Sie das Ökosystem der Maus nachbilden?

In dieser Lektion haben Sie etwas über maschinelles Lernen gelernt und wie man es anwendet. Sie wissen auch, wie Sie eine Kommunikation (und warum vorzugsweise drahtlos) zwischen einem Sprite und dem mBot2 einsetzen und für den Datenaustausch zwischen den beiden nutzen können. Darüber hinaus haben Sie es geschafft, dass verschiedene technische Funktionen des mBot2 zusammenarbeiten.

Nun ist es Zeit für eine kurze Reflexion. Denken Sie alleine nach und diskutieren Sie mit der Gruppe:

- Was ist Ihnen Ihrer Meinung nach gut gelungen?
- Was könnte noch besser sein?
- Welche Teile der Lektion sind Ihnen leicht gefallen und welche fanden Sie eher schwierig?
- Worüber hätten Sie gerne mehr Erklärungen?
- Wer könnte Ihnen dabei helfen?