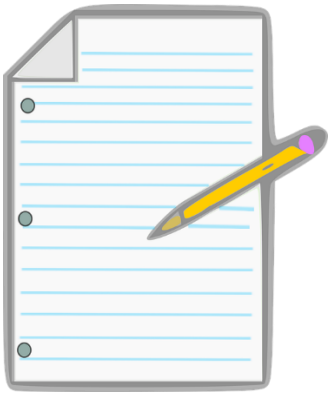


Grafik von PIXABAY CCO PUBLIC DOMAIN

ARBEITSBLÄTTER FÜR DIE ARBEIT MIT DEM ROBOTERKOFFER

AB1_Ein Blick in den Garten
AB2_Was ist ein Roboter?
AB3_Vergleich Mensch - Roboter
AB4_Von Sensoren und Aktoren
AB5_LEDs Farbmischung
AB6_Eine Binäre Geschichte
AB7_Dezimalzahl in Binärzahl umwandeln
AB8_Binäres Zählen
AB9_ Hexadezimalsystem
AB10_Was ist ein Programm
AB11_ Visuelle Programmierung (VPL)
AB12_Prüfe dein Wissen
AB13_Das magische Robotik-Quadrat!

(1) Ozobot AB1 – AB14

(2) InOBot AB1 – AB6

(3) Thymio AB1 – AB5

(4) Micro:bit AB1 – AB2

(5) mBot AB1 – AB9

Danke für das Korrekturlesen an Josef Schwarz!

H. Milchram, Oktober 2021

AB1_Ein Blick in den Garten

Aufgabe A: Beschreibe, was du auf dem Bild siehst.
Wozu sind die verschiedenen Teile erforderlich!

Wie funktioniert ein Mähroboter?

https://bit.ly/2Mf9CrA

Aufgabe B: Beantworte die folgenden Fragen:

1. Welche Arbeit führt der Roboter auf dem Bild aus?

2. Wer hat dies früher gemacht? _____

3. Welches Werkzeug wurde dazu benötigt? _____

4. Wer steuert den Roboter auf dem Bild?

5. Woher weiß er, was er tun soll?

6. Kennst du noch andere Roboter?

7. Welche Arbeiten erledigen diese?

8. Erkläre mit eigenen Worten, was alle Roboter gemeinsam haben.

AB2_Was ist ein Roboter?

Beschreibung eines Roboters:

1. Er arbeitet nach programmierten Anweisungen.
2. Er nimmt die Umgebung mit Sensoren wahr.
3. Er kann verschiedene Tätigkeiten selbständig ausführen.
4. Er wird durch einen Prozessor (Teil vom Computer) gesteuert.

ROBOTER
https://bit.ly/2vNcTEn

Aufgabe A: Kreise bei jedem Bild ein: Ist das ein Roboter ✓ oder ist es kein Roboter ✗ ?
Aufgabe B: Begründe deine Entscheidung mit Hilfe der vier genannten Eigenschaften.

Roboter? ✓ ✗

Roboter? ✓ ✗

Roboter? ✓ ✗

Roboter? ✓ ✗

Roboter? ✓ ✗

Roboter? ✓ ✗

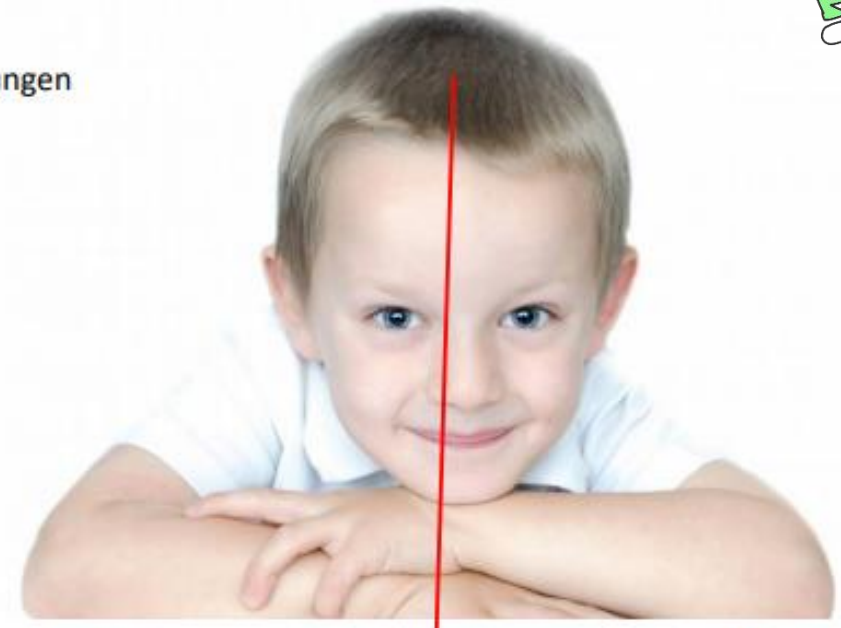
Von spannenden Problemen zu kreativen Lösungen PH Luzern

nö://media
Wir machen Sie DigiFit

H. Milchram, Oktober 2021

AB3_Vergleich Mensch - Roboter

Aufgabe: Verbinde die entsprechenden Bezeichnungen und Bildteile miteinander.



Ohr | Haut | Mund, Hand-zeichen | Nahrung | Gehirn | Auge | Muskeln

Daten verarbeiten	hören	sehen	spüren	sprechen, Zeichen geben	bewegen	Energie nutzen
Prozessor	Mikrofon	Licht-sensoren, Kamera	Tempera-tursensor, Druck-sensoren	Laut-sprecher, Licht-zeichen	Motoren	Akku / Batterie

AB4_Von Sensoren und Aktoren

Ein Sensor ist eine passive Komponente, er erfasst Dinge.

- ➔ Sensoren sind in etwa vergleichbar mit den fünf Sinnen beim Menschen.

Ein **Aktor** ist eine aktive Komponente, er **bewegt** Dinge oder **führt etwas aus**.

- ➔ Aktoren sind zum Beispiel vergleichbar mit dem Gehen, dem Sprechen oder der Mimik beim Menschen.

Aufgabe: Kreuze an, welche der folgenden Bauteile **Sensoren** und welche **Aktoren** sind.

Tipp: Es gibt auch Bauteile, die nichts von beidem sind.

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

❑ Sensor
❑ Aktor

Ein Roboter besteht aus verschiedenen Sensoren und Aktoren.

Von spannenden Problemen zu kreativen Lösungen PH Luzern

nö://media
Wir machen Sie DigiFit

AB5_LEDs - Farbmischung

Am PC oder bei **RGB-LED**s kann eine Farbe durch ihre Anteile an den drei **Primärfarben**

_____, _____ und _____ definiert werden.

Dieses als **RGB-System** bezeichnete Verfahren baut auf der **additiven Farbmischung** auf. Jede Farbe wird durch ihren Anteil an den drei Primärfarben (**Rotwert**, **Grünwert**, **Blauviolettwert**) definiert.

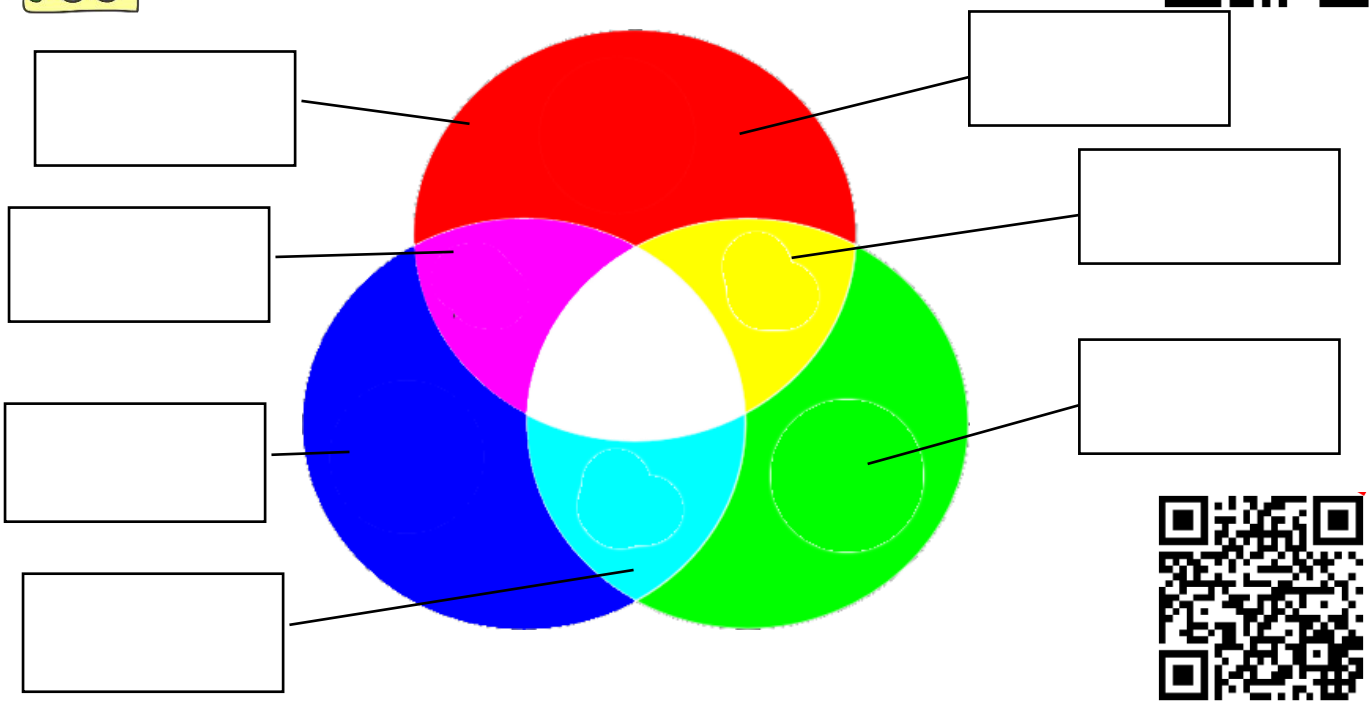
Die drei Farbwerte werden jeweils durch ein **Byte** = 8 Bits dargestellt und können somit die Werte von 0 bis 255 annehmen. Wird eine Grundfarbe für die Darstellung des Farbtons nicht benötigt, beträgt der entsprechende Zahlenwert 0. Da jede der drei Grundfarben in 256 Stufen (0 - 255) dargestellt werden kann, sind insgesamt $256^3 = 256 \times 256 \times 256 = 16\,777\,216$ unterschiedliche Farben darstellbar.

Eigenschaften der Grundfarbe

- ✓ Grundfarben können nicht aus anderen Farben gemischt werden.
- ✓ Alle anderen Lichtfarben werden durch Mischen der drei Grundfarben erzeugt.

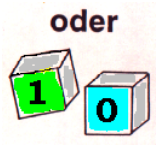
Additiver Farbmischer: http://www.spectrumcolors.de/cor_rgb_demo.php

Wie heißen die 7 Grundfarben?
Gib auch die jeweiligen RGB-Werte an!



Die meisten Farbdrucker verwenden die Farben **CYAN**, **MAGENTA**, **YELLOW**
→**Subtraktive Farbmischung** **http://www.spectrumcolors.de/cor_cmyk_demo.php**

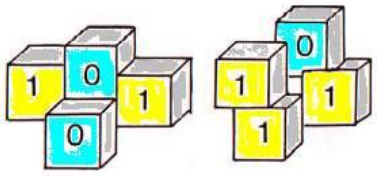
AB6_Eine Binäre Geschichte



Bit = **binary digit**, kleinste Informationseinheit (bestehend aus 0 oder 1)

Im Computer entspricht ein Bit einem Schaltvorgang: **Strom fließt** oder **Stromkreis unterbrochen**.

Ketten von Bits werden zur Darstellung von Buchstaben und größeren Zahlen benutzt.
Mit **8 Bits** = 1 **Byte** können dadurch 2^8 = 256 Zahlen (0 -255) dargestellt werden.



Eine binäre Geschichte

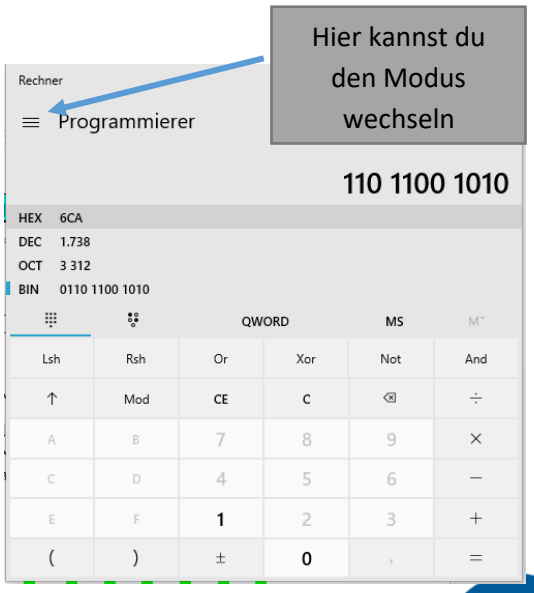
- ✓ I was born in year 110 1110 1010 in Bonn.
- ✓ I became a professional musician at the age of 1011.
- ✓ I lived in Vienna since 111 0000 0010.
- ✓ I wrote my first symphony in 111 0000 1000.
- ✓ The third symphony „EROICA“ was written in 111 0000 1100.
- ✓ I died in 111 0010 0011 I was 11 1000 years old.

My name is _____.

Tipp: Verwende zum Umrechnen der Binärzahlen den **Windows Taschenrechner** im Modus „**Programmierer**“!

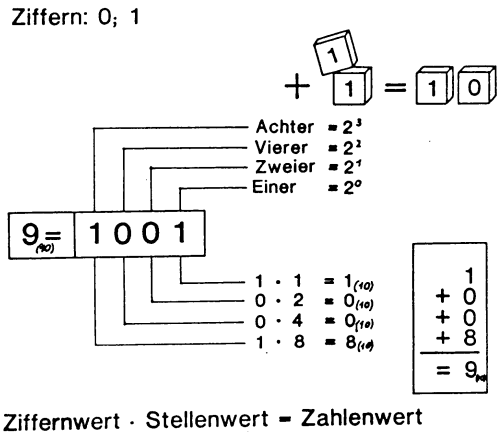
Dein(e) Lehrerin/Lehrer kann dir sicher erklären, wie die Umrechnung funktioniert!

Schreibe die **Dezimalzahlen** (DEC) jeweils über die im Text angegebenen **Binärzahlen** (BIN)!



AB7_Dezimalzahl in Binärzahl umrechnen

Zur Umwandlung einer Dezimalzahl in eine **Binärzahl** wird die Dezimalzahl durch die Basis (beim Dualsystem 2) dividiert. Das Ergebnis der Division wird ohne Berücksichtigung des Restes erneut dividiert. Dieser Vorgang wird so lange wiederholt, bis das Ergebnis Null ist. Die **Divisionsreste** (Überlauf) bilden **von rechts nach links angeschrieben** die gesuchte Binärzahl.



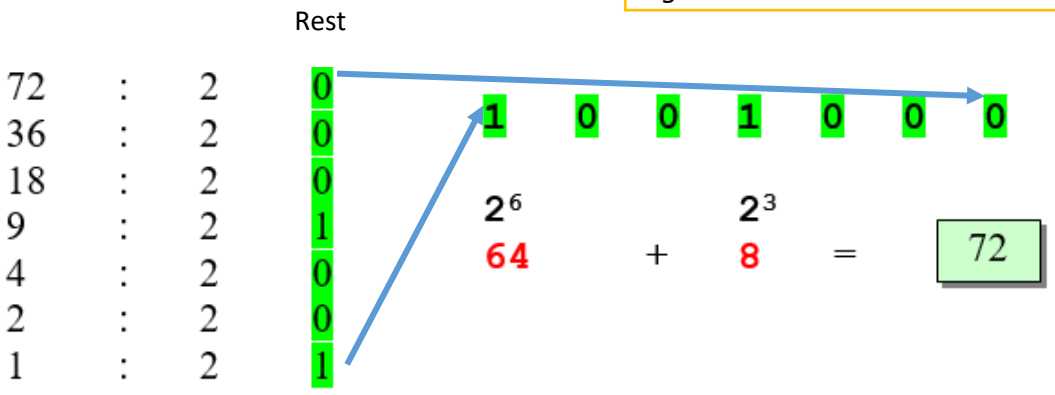
Dieser Vorgang, bei dem die Reste der Division ermittelt werden, wird als **Restwertdivision (Modulo = mod)** bezeichnet!

In den meisten **Programmiersprachen** wird für die **Modulo-Funktion** das Zeichen „**%**“ verwendet.

72 mod 2 = 0
9 mod 2 = 1;
12 mod 7 = 5
16 mod 8 = 0

Binäre Zahlen werden wie dezimale Zahlen erzeugt. Anstelle von 10 Ziffern (0,1,2 ...9) gibt es aber nur 2 Ziffern (0,1) Bei jeder Erweiterung um eine Stelle verdoppelt sich der Stellenwert.

Die einzeln ermittelten Restwerte werden anschließend, beginnend mit dem zuletzt ermittelten Restwert, in einer Reihe angeschrieben.



Wandle die Zahlen 134, 229 ohne Taschenrechner in Binärzahlen (Dualzahlen) um!

H. Milchram, Oktober 2021

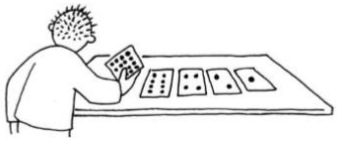
AB8_Binäres Zählen

Daten sind im Computer als eine Folge von Nullen und Einsen gespeichert und werden auch so übermittelt. Wie können wir Wörter und Zahlen darstellen, indem wir nur diese beiden Symbole verwenden?

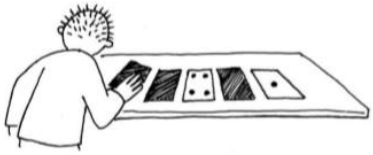
Zählen Lernen wie ein Computer

So, du meinst also, du weißt bereits, wie man zählt? Na dann, hier ist eine neue Art zu zählen! Wusstest du, dass Computer nur Nullen und Einsen kennen? Alles was du auf einem Computer siehst oder von ihm hörst – Wörter, Bilder, Filme, ja sogar Musik, wird mit diesen beiden Zahlen gespeichert. In dieser Aktivität werden wir uns damit befassen, wie man geheime Nachrichten mit Freunden austauscht, genauso wie es der Computer tut.

Aufgabe Nimm die **Kopiervorlage BINÄR-Zahlen** zur Hand und schneide die Karten aus der Vorlage aus. Lege sie danach vor dir auf den Tisch wie dargestellt (die Karte mit 16 Punkten ganz links):



Wichtig: Stelle sicher, dass die Karten genau in derselben Reihenfolge da liegen, wie auf dem Bild.



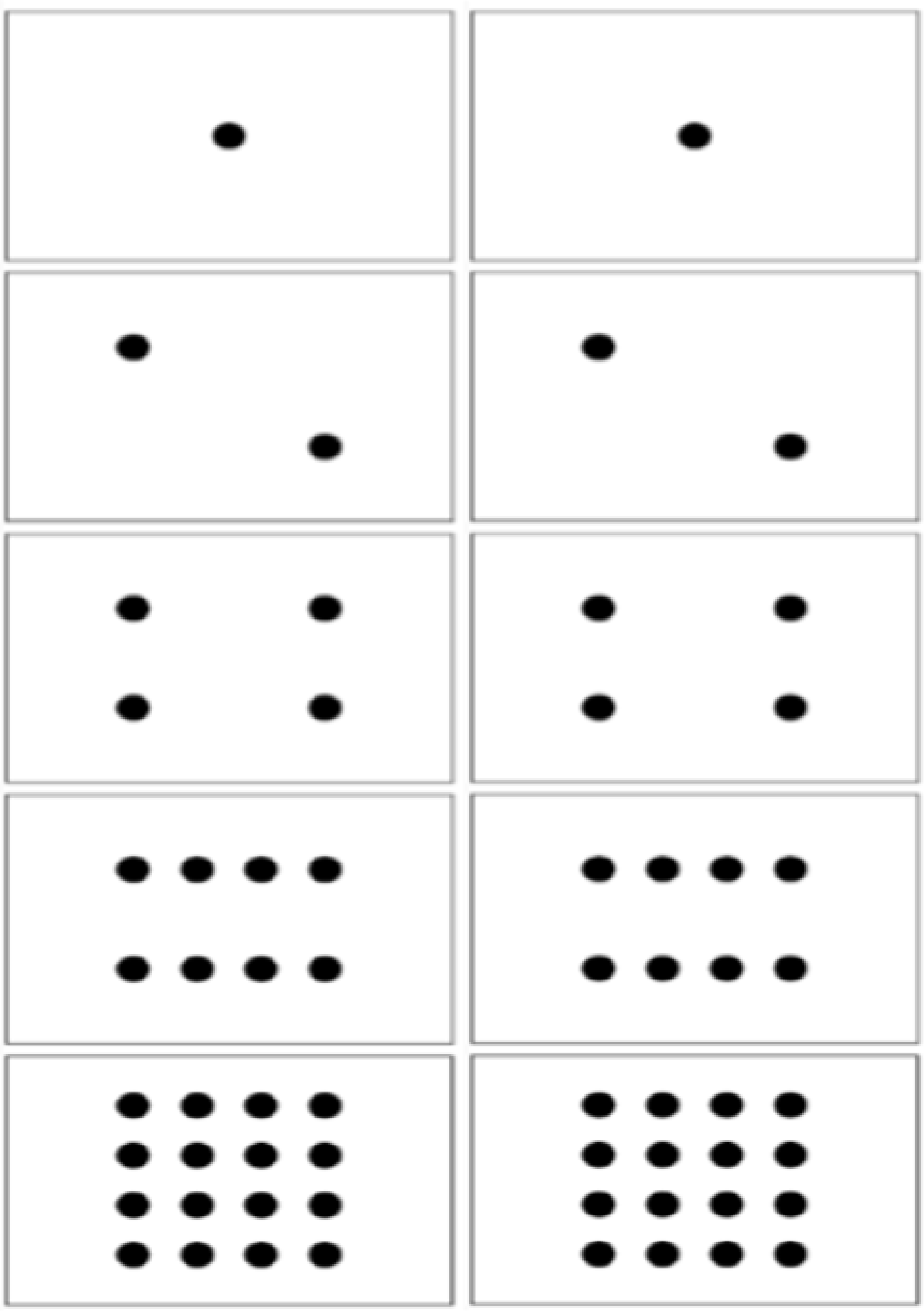
- ✓ Drehe nun die Karten so um, dass nur noch **5** Punkte sichtbar sind. Behalte dabei die Reihenfolge bei! Schreibe die Zahl in Binärschreibweise an: leere Karte = 0, Karte mit Punkten = 1

Dezimal	Binär
5	**0 0 1 0 1**

- ✓ Wie stellt man **3** = _____ **12** = _____ und **19** = _____ dar?
- ✓ Gibt es mehrere Möglichkeiten eine Zahl zu bilden? _____
- ✓ Gib die größte _____ bzw. kleinste Zahl _____ an, die du auf diese Weise mit den 5 Karten bilden kannst?
- ✓ Gibt es Zahlen zwischen der größten und der kleinsten Zahl, die man nicht erzeugen kann? _____

Zusatzaufgabe für Experten:
Versuche der Reihe nach die Zahlen 1, 2, 3, 4 zu bauen. Finde eine logische und zuverlässige Methode, mit welcher du sagen kannst, welche Karten als nächstes gedreht werden müssen, wenn du hochzählst.

Kopiervorlage BINÄR-Zahlen



AB9_Hexadezimalsystem

Beim Binärsystem enthält jedes **Halb-Byte** 16 Kombinationsmöglichkeiten. Die meisten modernen Datenverarbeitungsanlagen verwenden daher heute das **Hexadezimalsystem** (Sedezimalsystem). Dieses System arbeitet mit der Zahl 16 als Basis. Mit diesem System ist es daher möglich, ein Halb-Byte durch ein einziges Zeichen darzustellen.

Dezimal Basis 10	Binär Basis 2	Hexadezimal Basis 16
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

239

2. Halbbyte	1. Halbbyte
1 1 1 0	1 1 1 0
14	15

Keine Buchstaben sondern **SEDEZIMAL - Zeichen**!

Wandle die Hexadezimalzahlen **89**, **2A** und **CD** ohne Taschenrechner in **Dezimalzahlen** um und überprüfe dein Ergebnis mit dem Windows-Rechner! Verwende dazu die obenstehende Tabelle.

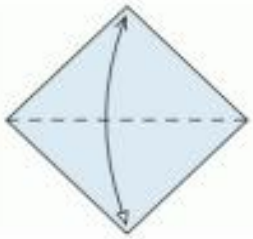
Hexdadezimal	Binär	Dezimal
9D	1001 1101	$2^7+0+0+2^4+2^3+2^2+0+2^0=128+16+8+4+1=$**157**
89		
2A		
CD		

H. Milchram, Oktober 2021

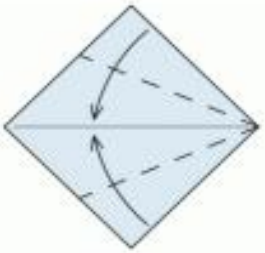
AB10_Was ist ein Programm

Ein **Programm** besteht aus einzelnen **Befehlen**, die in einer **bestimmten Reihenfolge** aneinandergereiht werden. Eine **Aufgabe** wird also in kleine einzelne **Befehle** aufgeteilt: Sie wird **programmiert**. Du kennst das von Kochrezepten oder Bastelanleitungen: Auch sie sind in einzelne Befehle aufgeteilt.

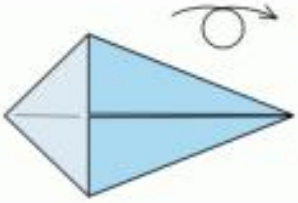
Ein Beispiel für ein **Programm**: Die **Aufgabe** ist es, einen Schwan zu falten.
Führe die **Befehle** 1 bis 9 aus.



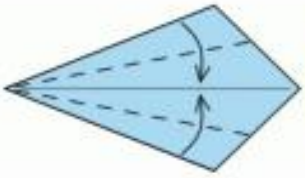
1. Quadrat in der Mitte falten und wieder öffnen.



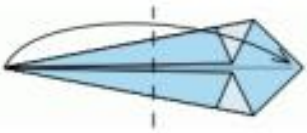
2. Kanten oben rechts und unten rechts in die Mitte falten.



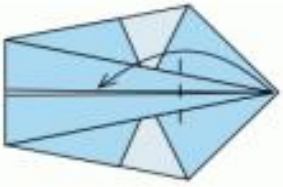
3. Modell wenden.



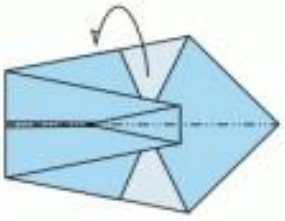
4. Kanten oben links und unten links in die Mitte falten.



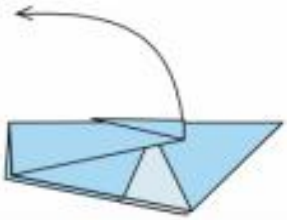
5. Linke Ecke zur rechten Ecke falten.



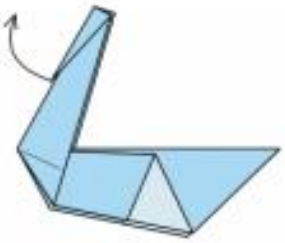
6. Spitze im gestrichelten Bereich nach links falten.



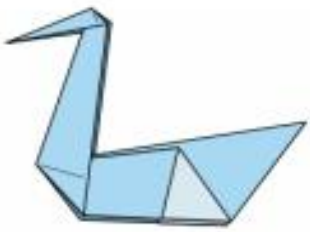
7. Obere Hälfte nach hinten falten.



8. Hals aufrichten und in dieser Position durch Zusammendrücken fixieren.



9. Kopf aufrichten und in dieser Position durch Zusammendrücken fixieren.



10. Fertiger Schwan.

Von spannenden Problemen zu kreativen Lösungen PH Luzern

Kennst du noch andere Beispiele für Programme: _____

_____

AB11_Visuelle Programmierung (VPL)

Visuelle Programmiersprachen benutzen vorgefertigte Programmelemente, meist in Form von Blöcken, die per Drag&Drop wie Puzzleteile zu einem Programm zusammengefügt werden können.

Die Blöcke sind so gestaltet, dass sich nur auch vom logischen Ablauf passende teile aneinanderfügen lassen. Meist erleichtert eine farbliche Kennzeichnung der einzelnen Blöcke die Orientierung bei der Auswahl der richtigen „Puzzleteile“

Zwei der am häufigsten verwendeten blockorientierten, visuellen Lernsprachen sind **Scratch** und **Blockly**. Vom Prinzip her sind VPLs Programmgeneratoren die aus den ausgewählten und zusammengefügten Programmblöcken einen Quellcode in einer professionellen Programmiersprache (Java Skript, Python, …) generieren.

Diese per **Drag&Drop** in **Scratch** zusammengesetzten Programmblöcke generieren im Hintergrund den nebenstehend Quellcode in **Java-Skript**.


```
//When play button pressed
setAllLedsColor("green");
pen("DOWN");
var cmf = 4;
for(var mvz = 0; mvz < cmf; mvz++)
{
  forwardForDistance("02", "10");
  spinLeftByDegrees("02", "90");
}
pen("UP");
setLedColor("00", "red");
playSound("3");
```

Was passiert wenn dieses Programm ausgeführt wird?
Beschreibe den Programmablauf mit eigenen Worten!

H. Milchram, Oktober 2021

AB12_Prüfe dein Wissen

- Welche Bauteile eines Roboters nehmen Informationen aus der Umwelt war?
  - ☐ Bluetooth-Schnittstelle
  - ☐ Motoren
  - ☐ Sensoren
  - ☐ LEDs
- Welcher Bauteil eines Roboters steuert die Motoren und Sensoren?
  - ☐ USB-Interface
  - ☐ Mikroprozessor
  - ☐ Aktoren
  - ☐ Bluetooth Schnittstelle
- Welche Bauteile in einem Roboter wandeln empfangene Signale in elektrische Impulse um?
  - ☐ Aktoren
  - ☐ Sensoren
  - ☐ Motoren
  - ☐ Rezeptoren
- Welches der folgenden Zahlensysteme wird nicht für die Verarbeitung der Daten im Computer verwendet?
  - ☐ Dualsystem
  - ☐ Hexadezimalsystem
  - ☐ Dezimalsystem
  - ☐ Binärsystem
- Das Byte: (mehrere Antworten sind richtig)
  - ☐ 1 Byte = 8 Bit.
  - ☐ 1 Byte = die kleinste Informationseinheit.
  - ☐ 1 Byte entspricht der Größe eines Gebissabdrucks in einem frischen Hamburger.
  - ☐ Es gibt 256 von ihnen.
- Welche der folgenden Aussagen stimmen? (mehrere Antworten sind richtig)
  - ☐ Ein Bit sind 8 Byte
  - ☐ Ein Byte ist eine Einheit zur Angabe der Größe eines Datenträgers
  - ☐ Ein Bit ist die kleinste Informationseinheit
  - ☐ Ein Bit ist der Speicherplatz für eines der beiden Zeichen: 0 oder 1
- Welcher der folgenden Begriffe ist keine Programmiersprache?
  - ☐ Blockly
  - ☐ Scratch
  - ☐ Android
  - ☐ Python
- Gottfried Wilhelm Leibnitz erkannte bereits 1679 die besondere Eignung des Dualsystems für die maschinelle Verarbeitung! ☐ true ☐ false
- Die Boolsche Algebra bildet die Grundlage der heutigen Informatik! ☐ true ☐ false
- Bluetooth und WLAN sind Schnittstellen zur drahtlosen Datenübermittlung! ☐ true ☐ false
- Ein Bit besteht aus zwei Halbbytes! ☐ true ☐ false
- Mit einem Byte können 256 Zahlen dargestellt werden! ☐ true ☐ false
- Roboter arbeiten nach dem EVA-Prinzip! ☐ true ☐ false
- Wie heißt in der Programmierung eine Kontrollstruktur, die das mehrfache Notieren von Befehlen erspart! _____

AB13_Das magische Robotik-Quadrat!

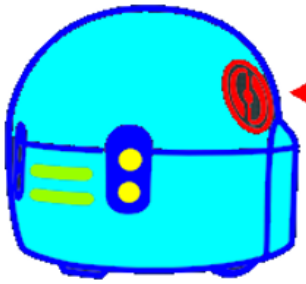
Trage jeweils die Zahl der richtigen Antwort in das Kästchen bei der Frage ein!
(1) Fuzzy Logic, **(2)** EVA-Prinzip **(3)** LEDs **(4)** Sedezimalzeichen **(5)** Hexadezimalsystem **(6)** George Boole **(7)** Mikroprozessor **(8)** Künstliche Intelligenz **(9)** Blue tooth, **(10)** Sensoren **(11)** AD-Wandler **(12)** Restwertdivision (modulo) **(13)** binary digit **(14)** Binärsystem **(15)** Byte **(16)** USB-Interface

☐ Schnittstelle für die Kommunikation mit einem Roboter?	☐ Welcher Bauteil eines Roboters kann sowohl **Sensor** und **Aktor** sein?	☐ Nach welchem Prinzip arbeiten Roboter und Computer?	☐ Was bedeutet die Abkürzung „**bit**“?
☐ Welches **Zahlensystem** findet in modernen Rechenanlagen vor allem Verwendung?	☐ Welche Bauteile eines Roboters wandeln empfangene Signale in **elektrische Impulse** um?	☐ Wie heißt ein elektronischer Bauteil, der **analoge** Signale in **digitale** Informationen umwandelt?	☐ Wofür steht die Abkürzung „**KI**“
☐ Schnittstelle zur **drahtlosen Datenüberttragung**.	☐ Wer baute ausgehende von den beiden logischen Möglichkeiten „**true**“ und „**false**“ die Grundlagen der heutigen Informatik auf?	☐ Bauteil eines Roboters zur Steuerung der **Aktoren** und **Sensoren**.	☐ Wofür wird in den meisten Programmiersprachen das Zeichen „**%**“ verwendet?
☐ Wie heißen die Buchstaben A-F, die im **Hexadezimal-system** zum Darstellen von Zahlen verwendet werden.	☐ Wie heißt eine Kette aus **8 Bits**, die zur Darstellung von Zahlen und Buch-staben verwendet wird?	☐ Andere Bezeichnung für **Dualsystem**.	☐ Wie heißt die **Logik**, die in der künstlichen Intelligenz und bei Experten-systemen verwendet wird?.

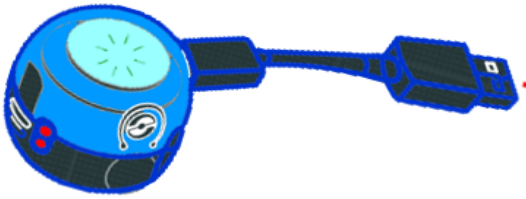
Hast du die Antworten richtig eingetragen, so ergeben die Summen der Spalten, Zeilen und der Diagonalen jeweils die Zahl 34!

Ozobot einschalten

Der Ozobot hat nur einen einzigen **Knopf auf der linken Seite**, der zum Ein- und Abschalten dient.



Einschalten: kurz drücken, LEDs beginnen zu leuchten
Ausschalten: etwas länger drücken



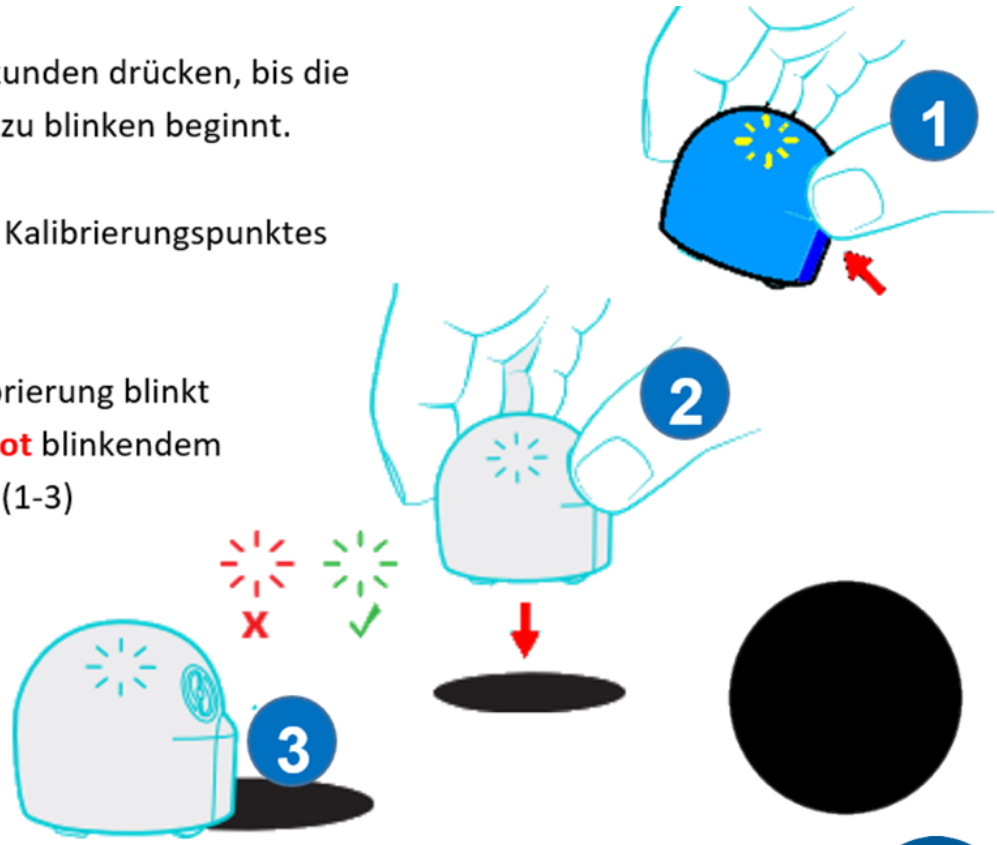
USB-Ladegerät, PC oder **Notebook**

Dein Ozobot besitzt zur **Stromversorgung** einen **AKKU**.
Sollte er leer sein (**rot** blinkendes Licht), muss er zunächst über ein **USB-Kabel** aufgeladen werden. Der **Micro USB-Anschluss** befindet sich auf der **Rückseite**!
Während des Ladevorgangs blinken die LEDs grün. Sobald dein Ozobot vollständig aufgeladen ist leuchten die LEDs konstant grün.

Ozobot kalibrieren

Die Sensoren deines Ozobots müssen nach dem Einschalten mit ihrer Umgebung vertraut gemacht werden. Dieser Vorgang wird als _____ bezeichnet.

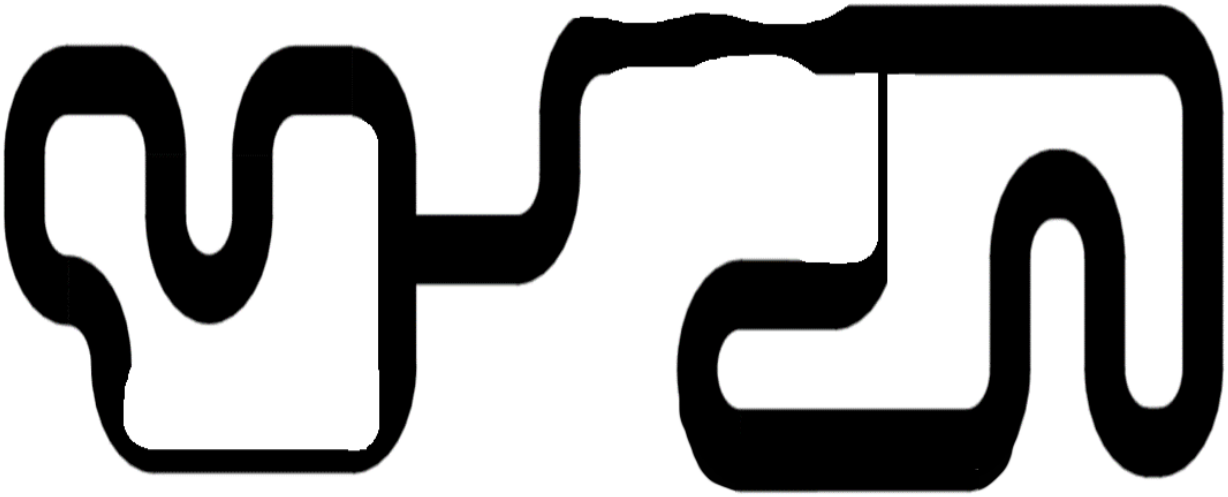
1. Einschaltknopf ca. 2 Sekunden drücken, bis die obere LED-Lampe **weiß** zu blinken beginnt.
2. Ozobot in die Mitte des Kalibrierungspunktes setzen und loslassen.
3. Nach erfolgreicher Kalibrierung blinkt dein Ozobot **grün**. Bei **Rot** blinkendem Licht muss der Vorgang (1-3) wiederholt werden!



nö://media
Wir machen Sie DigiFit

Ozobot AB2a

Stelle deinen Ozobot auf die schwarze Linie und beobachte!
Was kannst du feststellen? Vergleiche mit der untenstehenden Anleitung zum Zeichnen von Linien!



Welche der folgenden Linien sind richtig, welche falsch?
Begründe und Markiere mit ✓ oder ✗

Ozobot AB2b

Linien

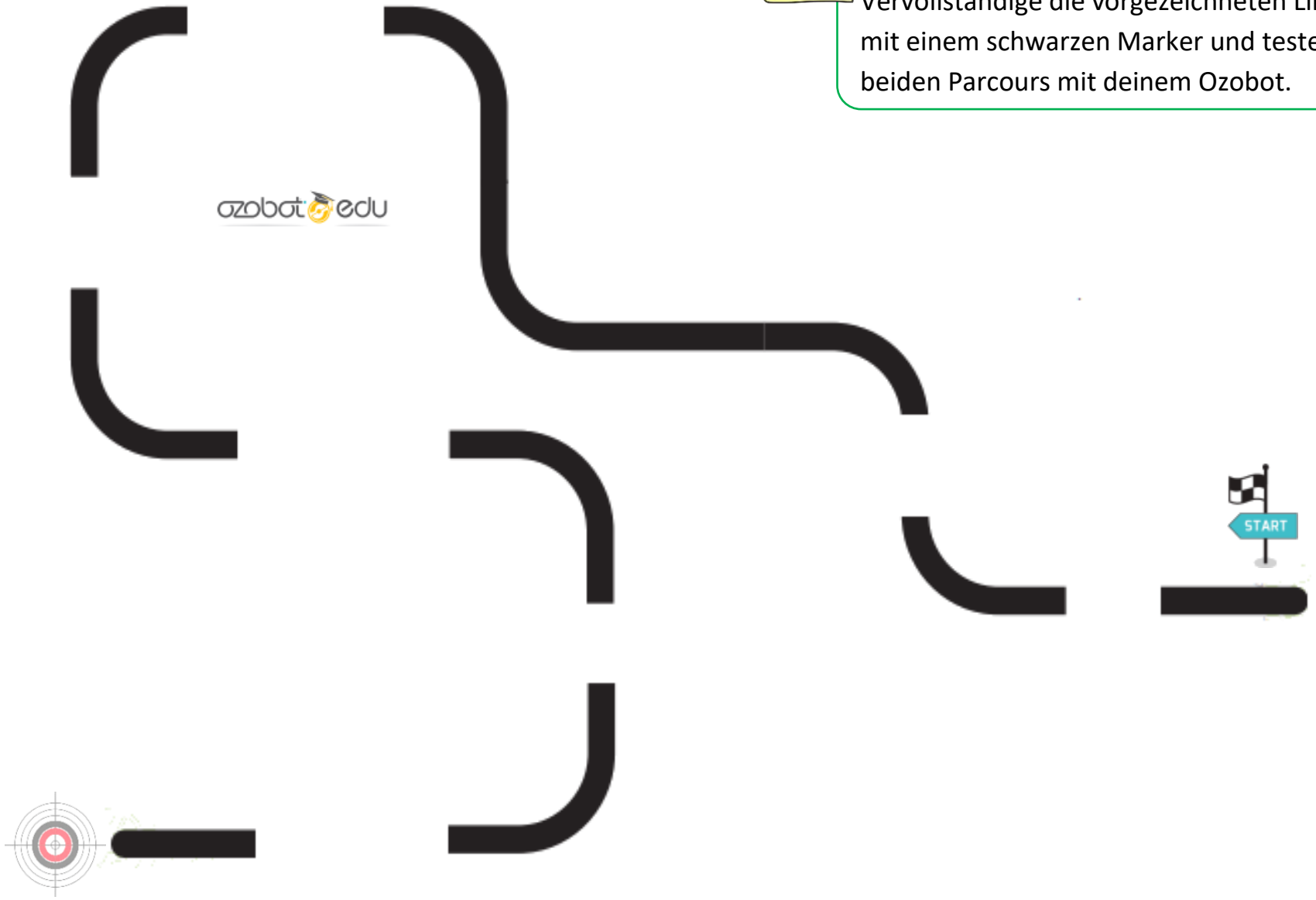
Kurven

Zum Zeichnen kannst du einen **schwarzen Marker** verwenden. Die **Strichstärke** sollte ca. **5mm** betragen.

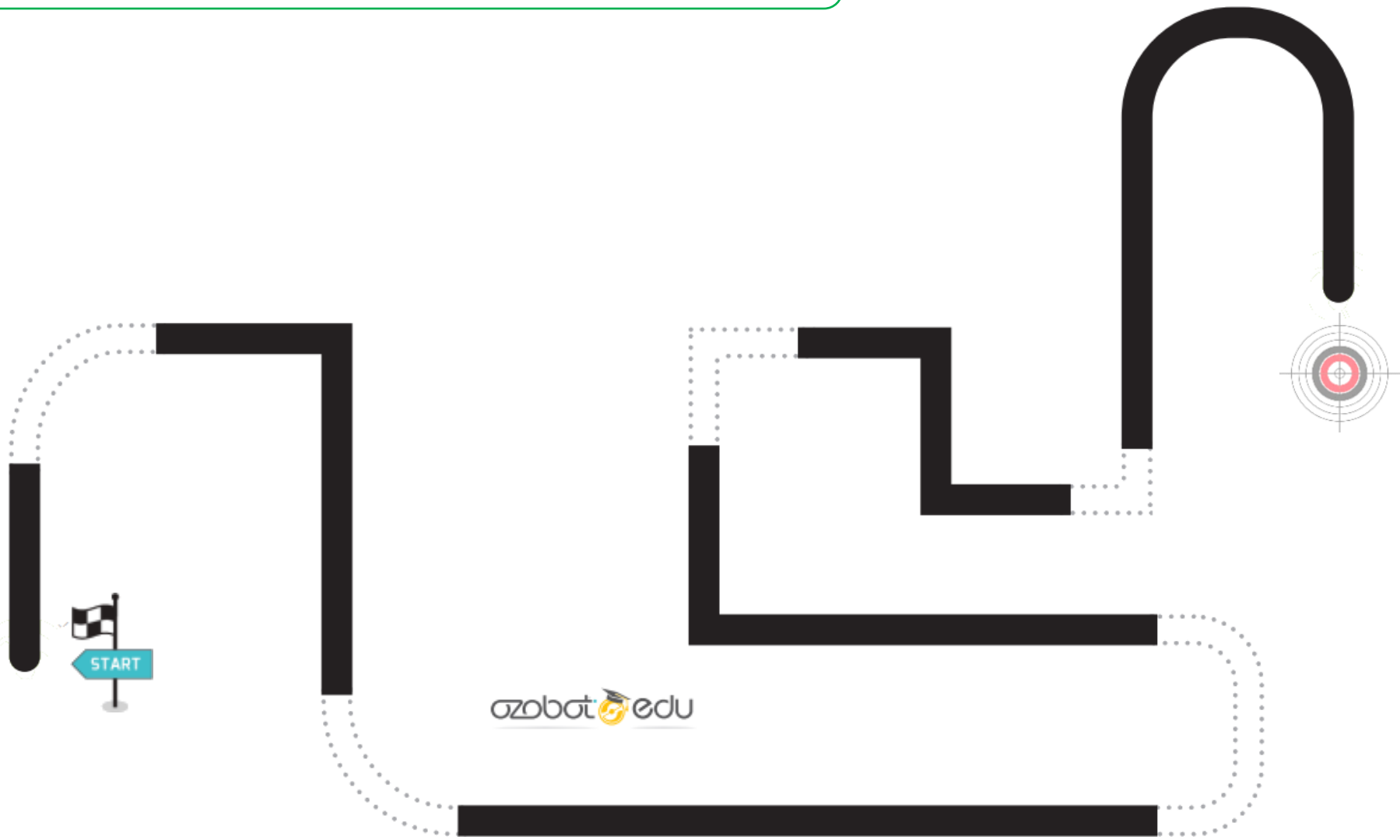
Zeichne auf ein weißes Blatt Papier deine eigene „Rennstrecke“. Die gezeichneten Linien sollten ca. 5mm stark sein und dürfen keine Unterbrechungen aufweisen!

H. Milchram, Oktober 2021

Vervollständige die vorgezeichneten Linienzüge mit einem schwarzen Marker und teste die beiden Parcours mit deinem Ozobot.



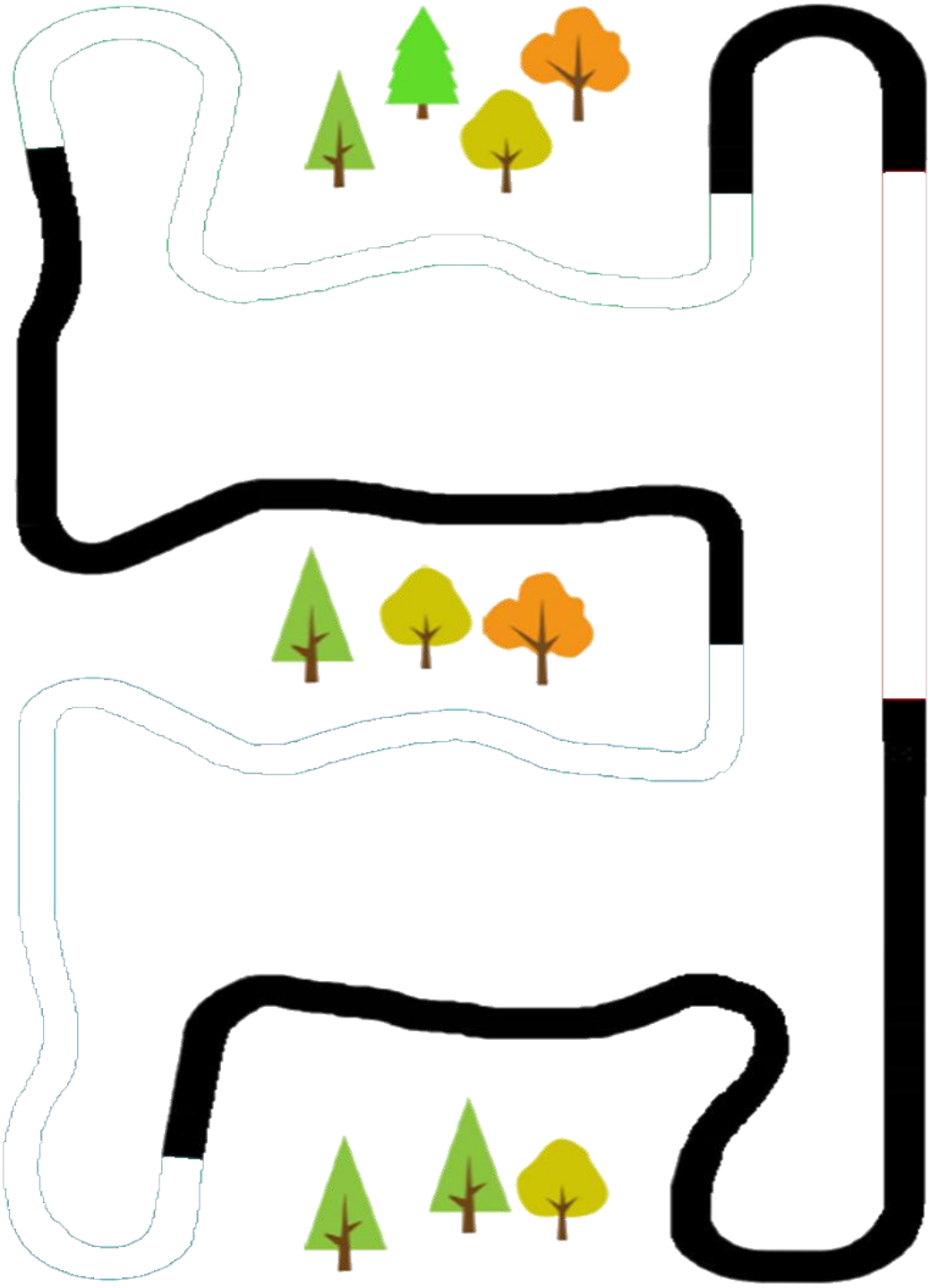
Vervollständige die vorgezeichneten Linienzüge mit einem schwarzen Marker und teste die beiden Parcours mit deinem Ozobot.



H. Milchram, Oktober 2021

Du kannst für das Zeichnen der Linien auch die Farben **ROT**, **GRÜN** und **BLAU** verwenden. Beobachte was passiert, wenn dein Ozobot auf eine andersfarbige Linie trifft?

_____



Dein Ozobot erkennt nicht nur Farben, sondern kann auch spezielle **Farbcodes** interpretieren. Damit hast du die Möglichkeit z.B. Geschwindigkeitsänderungen oder Richtungswechsel durchzuführen.

Ozobot Farbcodes: Regeln für die Verwendung

Dein **Ozobot** kann mit **Ozocodes**, einer besonderen Sprache programmiert werden, die aus kurzen **Farbsequenzen** (**rot**, **blau**, **grün**) auf **schwarzen Linien** besteht.  Erkennt dein Ozobot eine Farbsequenz, führt er bestimmte, vorprogrammierte **Aktionen** aus oder ändert sein **Aussehen** (LEDs leuchten in unterschiedlichen Farben).

Welche der folgenden Farbcodes sind richtig, welche falsch?

Markiere mit ✓ oder ✗. Begründe deine Entscheidungen!

(1) _____

(2) _____

(3) _____

(4) _____

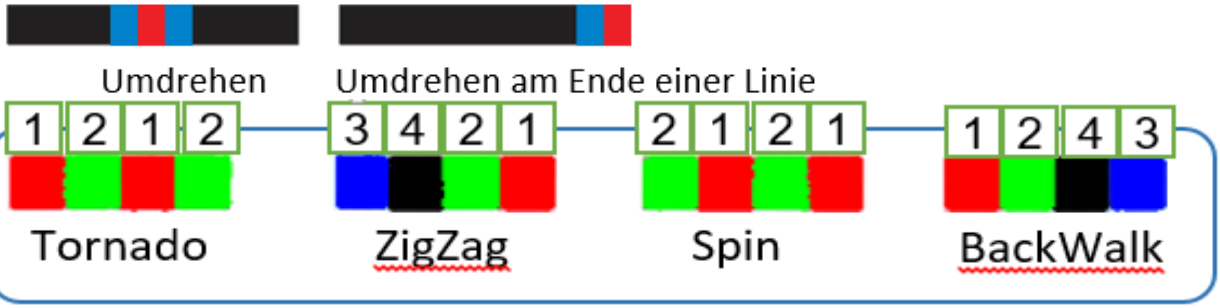
FARBCODES

Geschwindigkeit:

Richtung ändern an einer Kreuzung

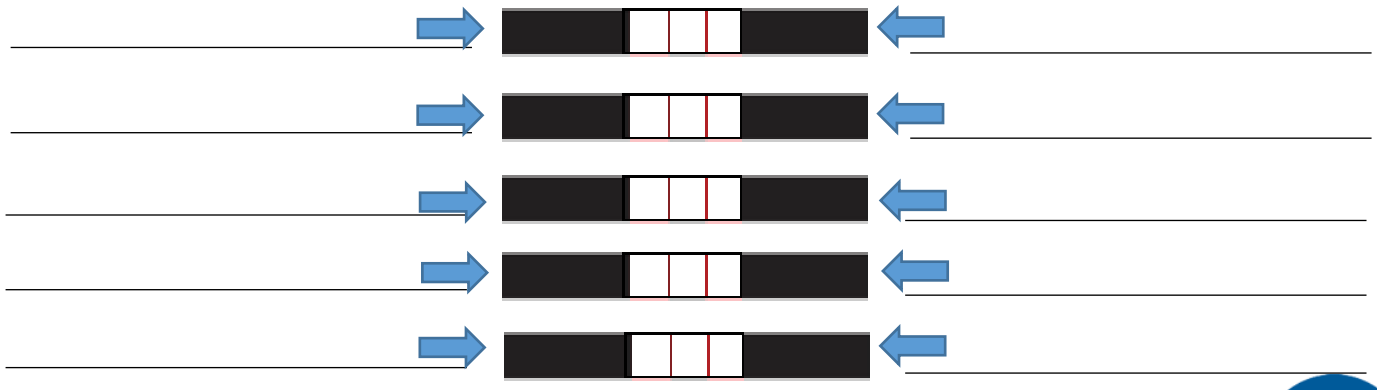
Linie suchen Sprung nach

Umdrehen (180° Drehung)



Manche Farbcodes haben je nach Richtung, von der aus sie gelesen werden, eine andere Bedeutung. zB **Tornado** →  ← **Spin** oder **ZigZag**→  ←**BackWalk**

Welche der obigen 3-teiligen Farbcodes haben abhängig von der Richtung in der sie gelesen werden eine andere Bedeutung. Markiere sie mit einem ✓ und trage unten die Lösung ein.



Füge die drei Codes fürs Umdrehen ein, damit dein **Ozobot** nicht stecken bleibt und Ins Ziel zurückkehrt! Verwende dazu die folgenden Farbcodes:

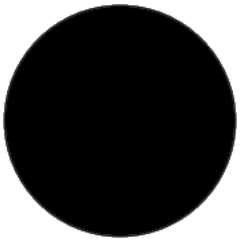
✓ Zweifarbige Codes stehen nur am Ende einer schwarzen Linie.

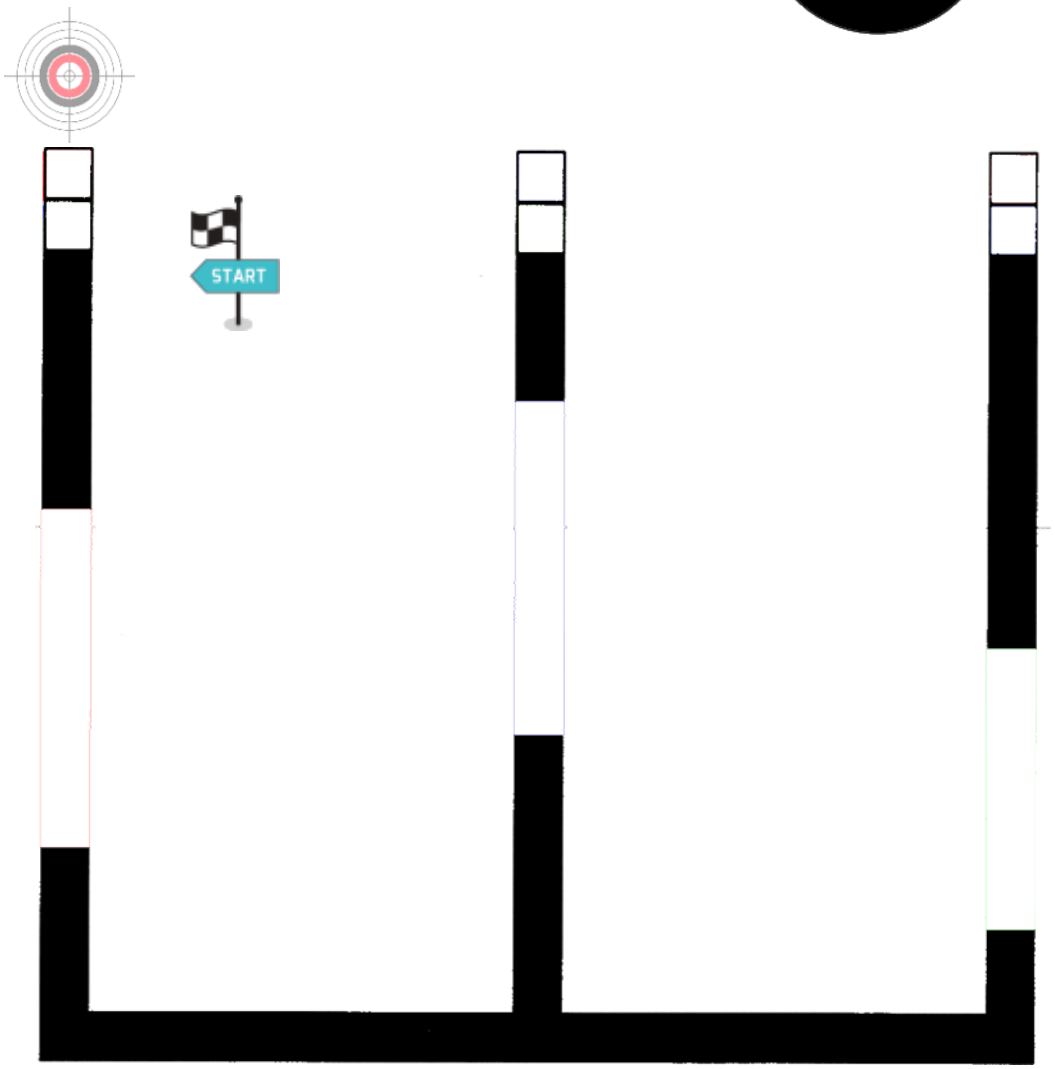
Umdrehen

Ende (nochmals spielen)

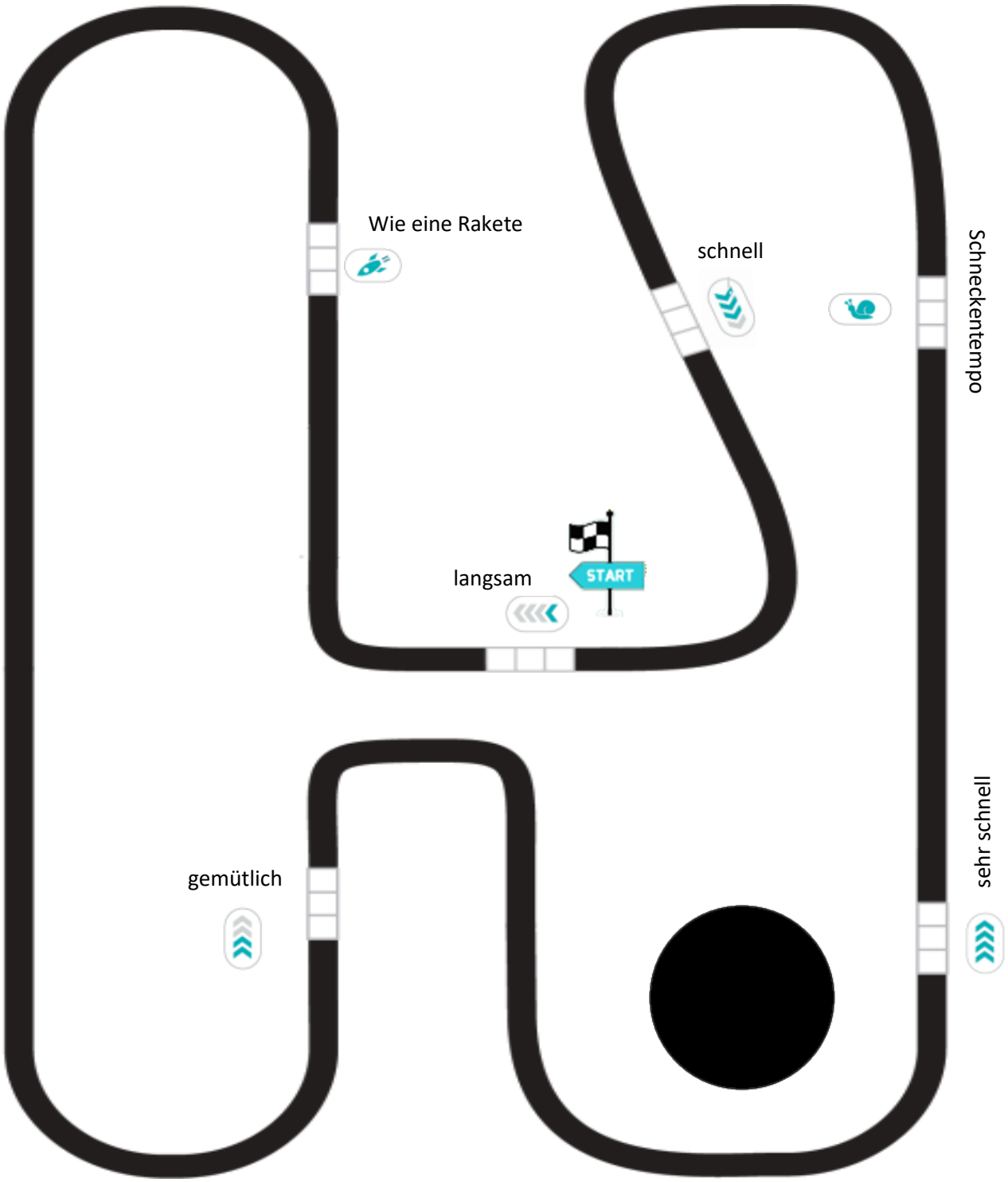
Ende (Spiel vorbei)

Ergänze die fehlenden Streckenteile mit den Farben **ROT**, **BLAU** und **GRÜN**!





Ergänze die Farbcodes. Benutze dazu die Farben **Blau**, **Grün** und **Rot**!
Denke daran, dass manche Farbcodes je nach Richtung, von der aus sie gelesen werden, eine andere Bedeutung haben. Der Pfeil gibt die Richtung an, in die dein **Ozobot** fahren soll



Geschwindigkeit:

Schneckentempo	Langsam	gemütlich
schnell	sehr schnell	wie eine Rakete

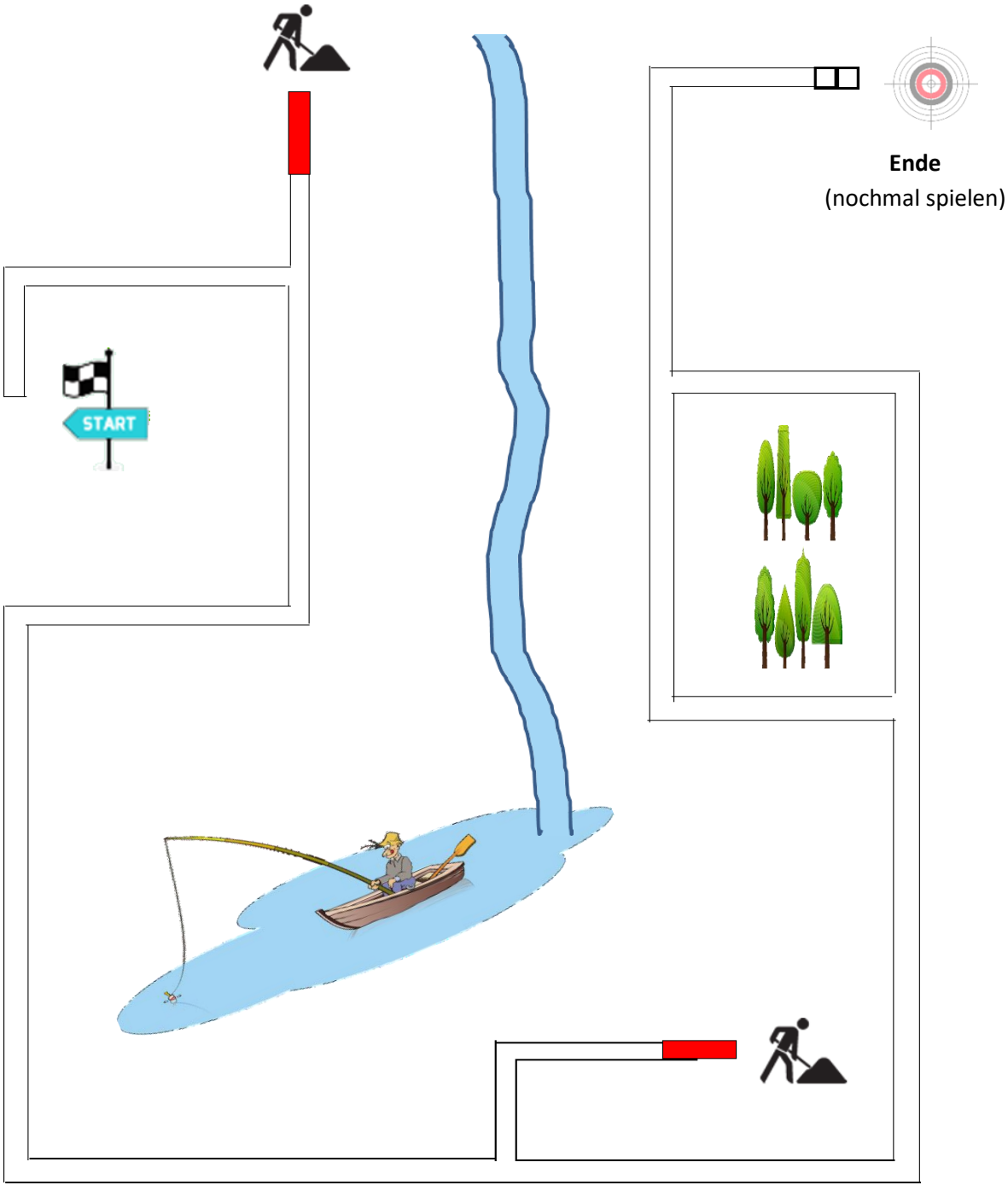
nö://media
Wir machen Sie DigiFit

H. Milchram, Oktober 2021

Kopfnuss: Bist du dieser Herausforderung gewachsen?
Du darfst nur folgende Codes verwenden, um ins Ziel zu gelangen:

2x 

2x 

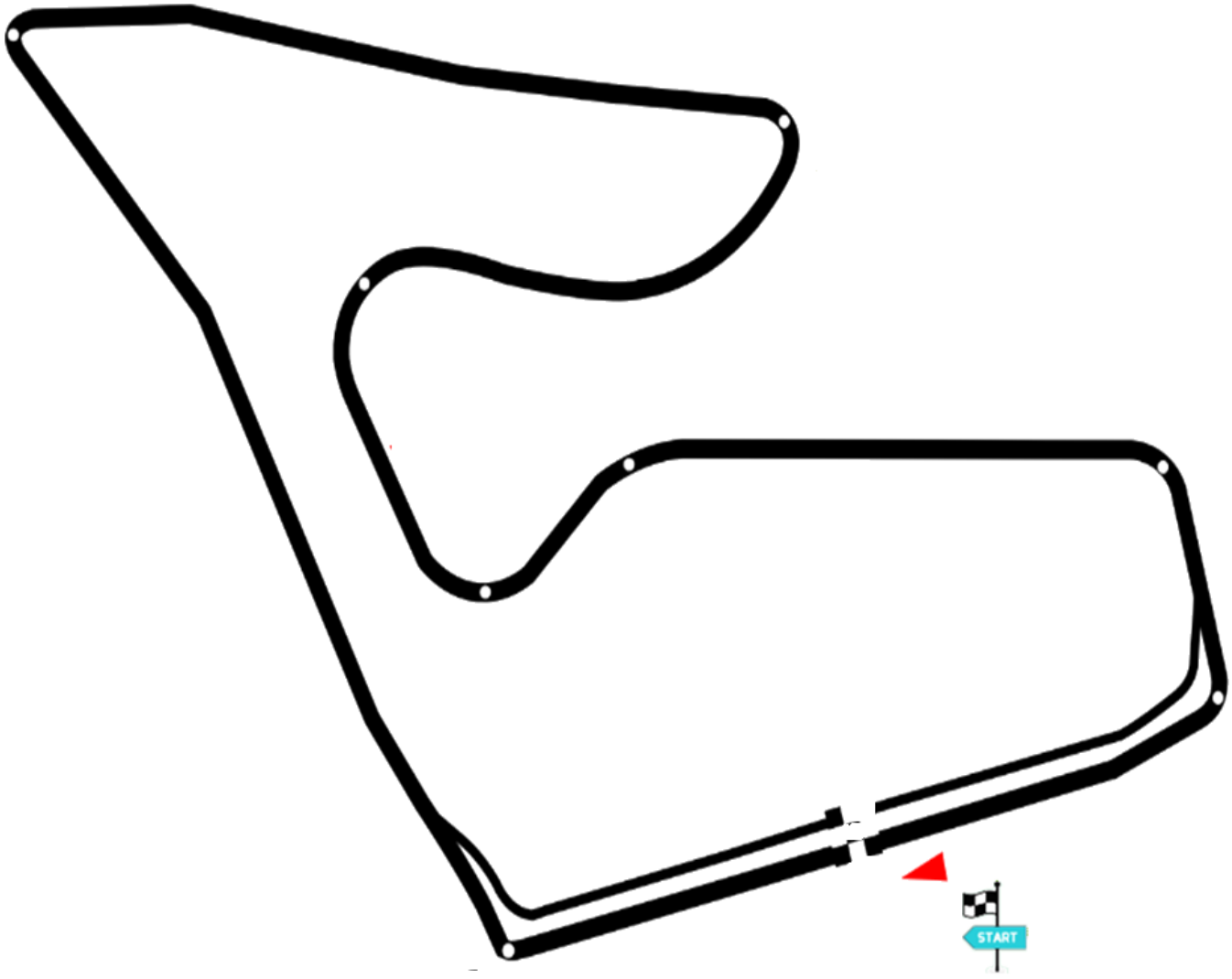


Ende
(nochmal spielen)

OzoBot: Ozobot evo APP

- ✓ Starte auf deinem Handy oder Tablet die Ozobot evo APP.
- ✓ Wähle „**Play without sign in**“ aus und folge den Anweisungen am Bildschirm
- ✓  Über diese Schaltfläche bekommst du fünf Möglichkeiten mit deinem Ozobot zu arbeiten.
- ✓ Wähle 

aus



Steuere deinen OzoBot über den **Spielbergring** und stoppe die Zeit!

- ✓ Ändere die **Geschwindigkeit!**
- ✓ Schalte die **Linienverfolgung** ein!
- ✓ Schalte die **LEDs** ein und aus!
- ✓ Ändere die **Farben** der LEDs!
- ✓ Probiere verschiedene **Lichteffekte** und **Soundeffekte** aus!

Programmieren mit Ozoblockly

Der Ozobot lässt sich außerdem mit **OzoBlockly** (**https://ozoblockly.com**), einer **visuellen Programmiersprache,** auch ganz ohne Linien und Farben verwenden und so vollkommen frei steuern. OzoBlockly kennt fünf Schwierigkeitsstufen von **NOVICE** (Programmierung mit ganz einfachen Symbolen, Eingabe von Texten) bis hin zu **MASTER** (Schleifen, komplexe Funktionen, Variablen) und die genaue Steuerung der einzelnen Motoren des Ozobots.

Ozobot AB11a

Dein Ozobot fährt auf der Eisbahn und zeigt verschiedene Bewegungen und Lichteffekte.

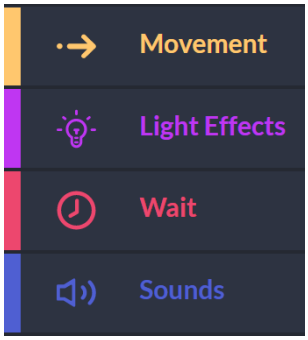
(1) Gehe auf die Website https://ozobot.com/

(2) Starte den OzoBlockly-Editor

(3) Wähle den Level 1 (Pre-Reader) aus!

(4) Aus den4 Registern kannst du die erforderlichen Blöcke per **Drag&Drop** in den Arbeitsbereich ziehen.

(5) Ausführen des Programms:



LIVE-Modus → Verbindung über Bluetooth herstellen!

Flashing → Übertragung des Programms über Lichsignale an deinen Ozobot (**autonomer Modus**)

Bitte deine Lehrerin/deinen Lehrer dir den Unterschied zu erklären!

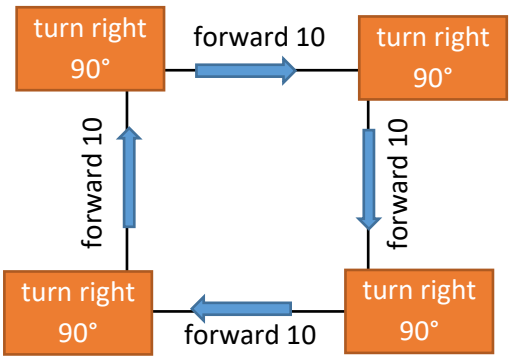
Ozobot AB11b

Dein Ozobot soll einen **Spaziergang** machen. Die zurückgelegte Wegstrecke soll dabei die Form eines **Quadrats** mit einer **Seitenlänge von 5 cm** betragen (1 Schritt = 1cm; 1 step). Bei jeder **Vorwärtsbewegung** soll die **obere LED grün** leuchten, bei jeder **Drehung rot**. Am **Ende des Spaziergangs** soll dein Ozobot sich **einmal linksherum im Kreis** drehen und dann eine **fröhliche Melodie** abspielen!

Auf der Website https://ozobot.com/create/challenges kannst du mit **ShapeTracer1** und **ShapeTracer2**, sowie mit **OZOTown**, Schritt für Schritt deine Programmierkenntnisse mit einem **virtuellen Ozobot** erweitern.

Ozobot AB12, AB13, AB14

Bringe deinen Ino-Bot dazu, ein **Quadrat mit einer Seitenlänge von 10 cm** zu zeichnen!



- ✓ **Das brauche ich:**
  InO-Bot + passenden Stift
- ✓ Tablet (Handy) mit Android oder iOS Betriebssystem, InO-Bot APP muss installiert sein!
- ✓ Bogen Packpapier oder ähnliches.

Trage hier die Befehle in der richtigen Reihenfolge ein:

1. Pen down
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____

Der **InO-Bot** besitzt eine integrierte **Stifthalterung**. Ein eingesetzter Stift kann über folgenden Block gesteuert werden:

Folgende Befehle aus dem Register „InO-Bot Facile“ werden benötigt:

Stift **heben** (up) und **senken** (down)

TIPP: Denke daran, dass du den **Stift**, dass du den Stift vor der ersten Bewegung des Roboters **senken** (pen down)und am Schluss wieder **heben** (pen up) musst!

Moves (Bewegungen) & **Turns** (Drehungen):

Richtung der Bewegung **forward** (vorwärts), **reverse** (rückwärts)

Geschwindigkeit: **lent** (langsam), **moyen** (mittel), **vite** (schnell)

Strecke in **cm**

Richtung der Drehung **left** (links), **right** (rechts)

Winkel in Grad

Geschwindigkeit: **lent** (langsam), **moyen** (mittel), **vite** (schnell)

Zusätzlich ist ein **Ereignis** erforderlich, durch welches das Programm gestartet wird. Wir verwenden dazu das Ereignis

Was fällt dir auf, wenn du den Programmablauf betrachtest?

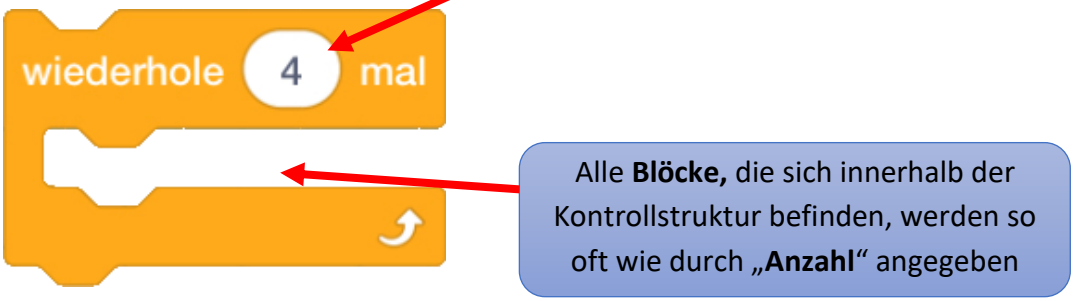
_____

Ersuche deine(n) Lehrer(in) dir das Prinzip einer **Schleife** (Iteration) zu erklären!

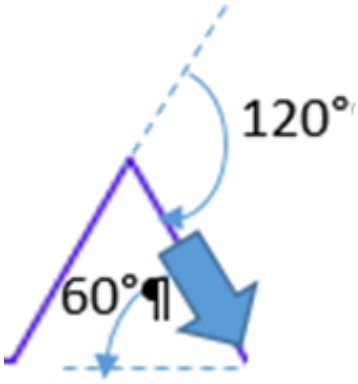
Versuche nun, das **Quadrat mit Hilfe einer Schleife** zu programmieren!

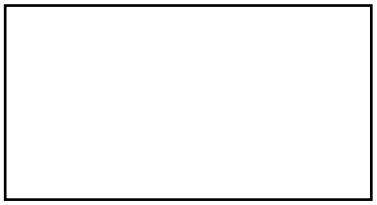
Eine Schleife (**Iteration**) ist bei der Programmierung eine Kontrollstruktur, bei der eine Anweisung oder ein Anweisungsblock solange wiederholt wird, bis eine Abbruchbedingung erfüllt wird. **Die Schleife** erspart das mehrfache Notieren von Befehlen.

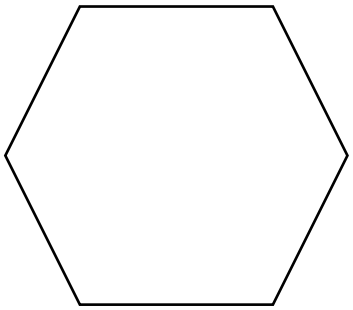
Die Blöcke zum Programmieren einer Schleife findest du im Register „**Steuerung**“ 



Versuche auf dieselbe Art und Weise auch Programme für andere geometrische Figuren wie **Dreieck**, **Rechteck** und **Sechseck** zu schreiben! Überlege welche Winkel du brauchst und zeichne sie in die Figuren ein! Probiere unterschiedliche **Geschwindigkeiten** und **Drehrichtungen** aus!







lent …langsam

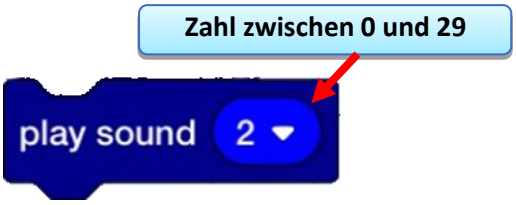
moyen …mittel

vite …schnell

TÖNE und GERÄUSCHE

Der Befehl „play“ für die Ausgabe von Tönen und Geräuschen befindet sich im Register „**InO-Bot Facile**“ 

Es gibt insgesamt **29** verschiedene voreingestellte Möglichkeiten, die über den folgenden Block zugewiesen werden!



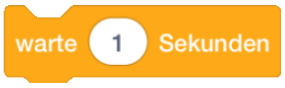
29 verschiedene Sounds

- ✓ 1 – 9: verschiedene **Klangeffekte**
- ✓ 10 – 16: **Klavier**
- ✓ 7 – 29: **Xylophon**

InO-Bot AB3a

Erstelle ein Programm, das hintereinander 10 zufällig ausgewählte Töne und Geräusche ausgibt. Nach jedem Ton oder Geräusch soll eine **Pause** von einer Sekunde (**1‘**) sein!

Tipp:

Verwende für die Pause den Block 

Für die Auswahl eines zufälligen Tones oder Geräusches kannst du den Operator 

verwenden

InO-Bot AB3b

Ändere das Programm so, dass

- ☑ nur die **Klangeffekte**
- ☑ nur die **Klaviersounds**
- ☑ nur die **Xylophon-Sounds**

abgespielt werden!

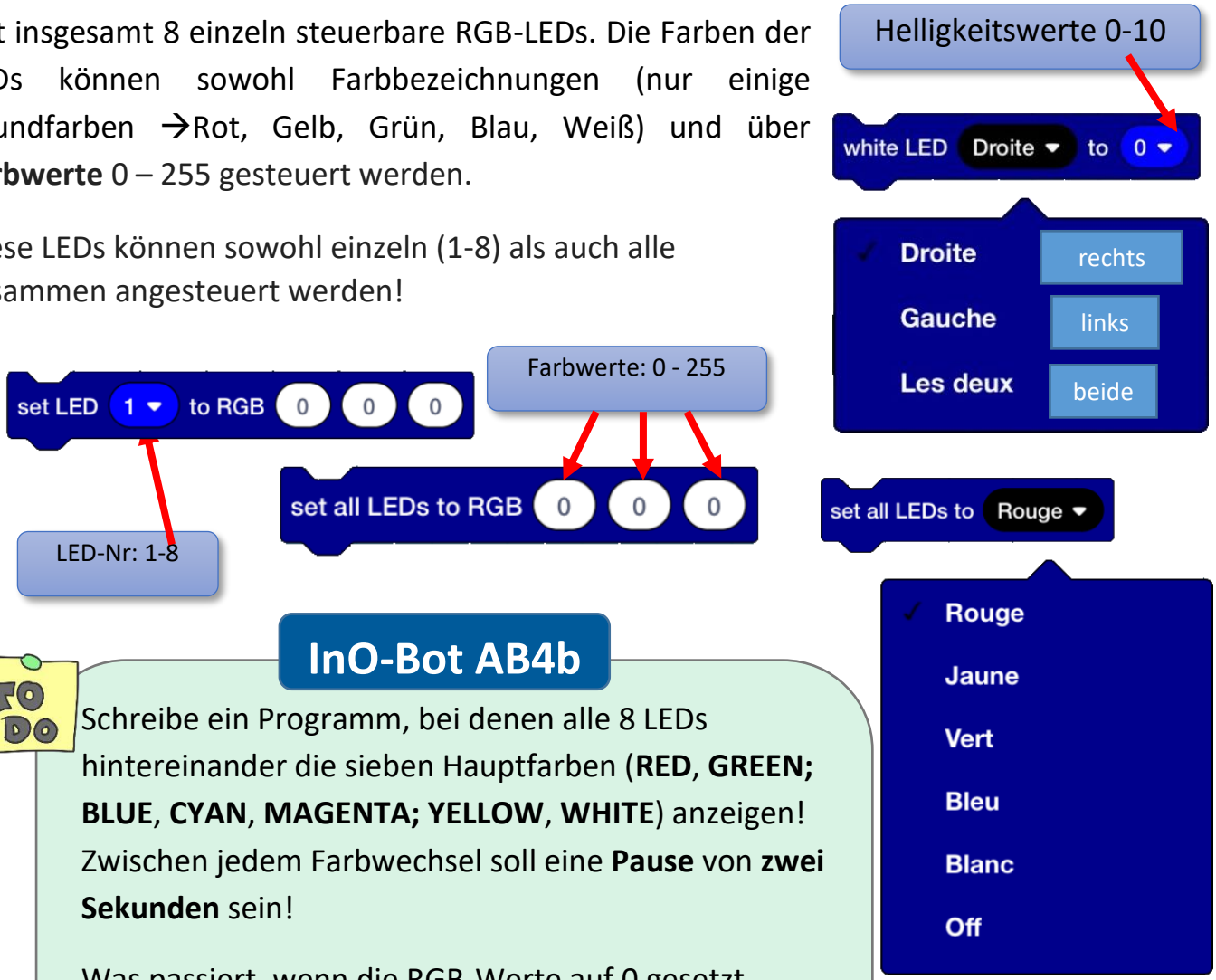
LED Steuerung

Der **InO-Bot** besitzt **2 Front-LEDs** (weiß) und **8 RGB-LEDs**, die unabhängig voneinander gesteuert werden können!

Schreibe ein Programm, dass zuerst das **rechte Front-LED** und dann das **linke Front-LED** für jeweils **1“** bei **mittlerer Helligkeit** einschaltet. Anschließend sollen **beide LEDs** für **3“** leuchten. Der Abschluss des Programms wird durch ein **Geräusch** angezeigt. Lasse das gesamte in einer **Endlos-Schleife** laufen!

Zusätzlich zu den beiden weißen Front-LEDs besitzt der InO-Bot insgesamt 8 einzeln steuerbare RGB-LEDs. Die Farben der LEDs können sowohl Farbbezeichnungen (nur einige Grundfarben →Rot, Gelb, Grün, Blau, Weiß) und über **Farbwerte** 0 – 255 gesteuert werden.

Diese LEDs können sowohl einzeln (1-8) als auch alle zusammen angesteuert werden!



InO-Bot AB4b

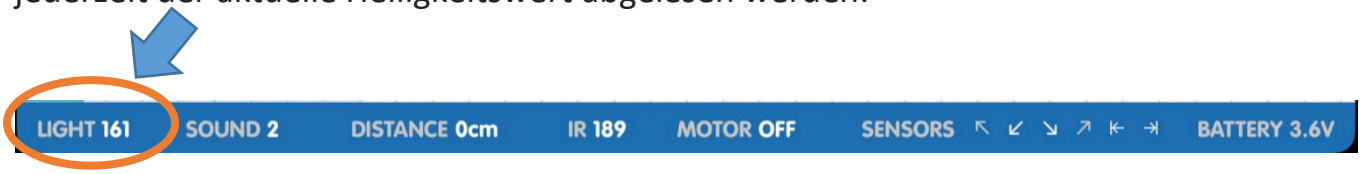
Schreibe ein Programm, bei denen alle 8 LEDs hintereinander die sieben Hauptfarben (**RED**, **GREEN; BLUE**, **CYAN**, **MAGENTA; YELLOW**, **WHITE**) anzeigen! Zwischen jedem Farbwechsel soll eine **Pause** von **zwei Sekunden** sein!

Was passiert, wenn die RGB-Werte auf 0 gesetzt werden? _____

_____

Lichtsensor

Der InO-Bot besitzt einen **Lichtsensor,** mit dem die LEDs abhängig von der Helligkeit ein- oder ausgeschaltet werden können! In der Statusleiste der Programmieroberfläche kann jederzeit der aktuelle Helligkeitswert abgelesen werden.



Programmiere deinen InO-Bot so, dass die **Front-LEDs bei Dunkelheit** eingeschaltet und sobald es hell wird, wieder ausgeschaltet werden!

Wert, bei dem eine bestimmte Aktion gesetzt wird.

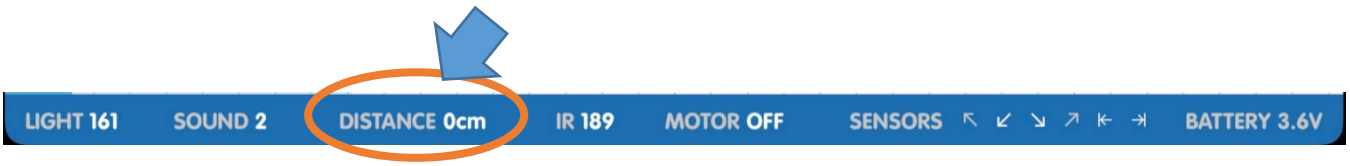
Mit der Funktion „light level“ wird der aktuelle Wert für die Umgebungshelligkeit ausgelesen!

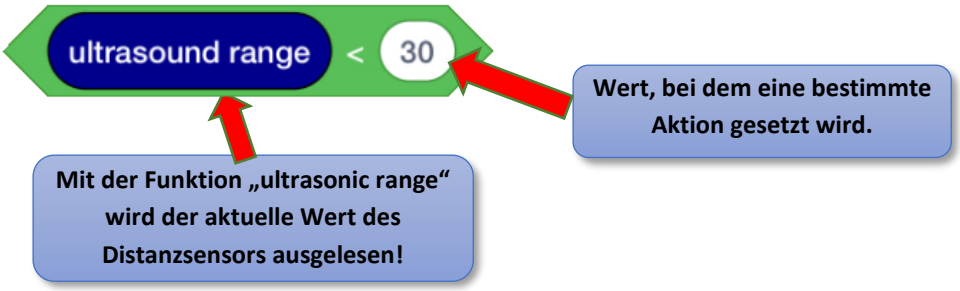
Distanzsensor

Mit Hilfe seiner beiden Ultraschall **Distanzsensoren** kann dein InO-Bot erkennen, ob sich ein Hindernis vor ihm befindet und damit eine Kollision droht.

Achtung Hindernisse

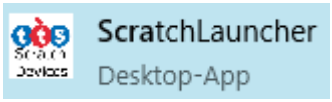
Dein InO-Bot soll sich langsam vorwärtsbewegen, nähert sich ein Hindernis, soll er stoppen und ein akustisches Signal abgeben. Anschließend soll der InO-Bot 10 cm zurück fahren und sich um 90° nach rechts drehen. Falls sich nun kein Hindernis im angegebenen Abstand befindet, beginnt er wieder vorwärts zu fahren.



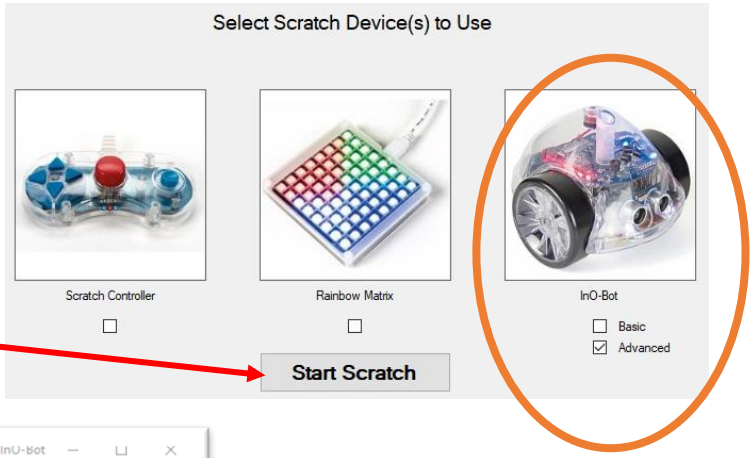


Programmierung mit Scratch

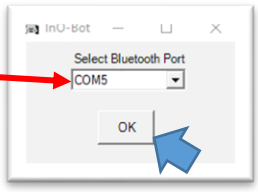
- ✓ **InO-Bot** einschalten
- ✓ **Scratch Launcher** starten



- ✓ **InO-Bot** auswählen
- ✓ Gewünschten **Modus** (**Basic** oder **Advanced**) auswählen
- ✓ **Start Scratch**-Schaltfläche anklicken



- ✓
- ✓ **Bluetooth-Port** auswählen (Voreinstellung beibehalten!)



- ✓ **Scratch Programmierumgebung** wird aufgerufen

InO-Bot AB7a

Lichteffekte und Soundausgabe:
Wenn die Leertaste gedrückt wird sollen hintereinander für jeweils 1 Sekunde alle LEDs **blau**, dann **grün**, **rot** und **lila** leuchten.
Zum Abschluss soll eine **beliebige Melodie** gespielt und die **LEDs ausgeschalten** werden.

Thymio: Verhaltensmuster

Nach dem Einschalten deines Thymio stehen dir _____ vorprogrammierte Verhaltensweisen zur Verfügung. Benutze die **Pfeiltasten** auf dem Roboter, um zwischen den einzelnen Verhaltensmustern zu navigieren. Über die **Taste in der Mitte** wird ein **Verhaltensmuster aktiviert** oder **deaktiviert**.

Gehe auf die Website **https://www.thymio.org/de/grundlegende-verhaltensweisen/** und informiere dich über die verschiedenen Verhaltensmuster und probiere sie aus.

GRÜN: Freundlich

Thymio _____ und _____ auf einen anderen freundlichen Thymio.

GELB: Neugierig

Thymio vermeidet _____ und _____ automatisch am Tischrand oder wenn die Unterlage schwarz ist.

ROT: Ängstlich

Thymio _____ sich, wenn du dich ihm mit deiner Hand näherst und er reagiert lautstark, wenn er in die Ecke gestellt oder in die Luft geworfen wird.

BLAU: Aufmerksam

Thymio ändert seine Farben und _____ sich je nach Anzahl der erkannten Klatschtöne.

____**x Klatschen**: dreht nach rechts / fährt vorwärts

____**x Klatschen**: Start/Stopp ____ **x Klatschen**: fährt einen Kreis

TÜRKIS: Erforschend

Thymio folgt einer _____ Spur auf dem Boden. Die Spur sollte mindestens 3 cm breit sein.

ROSA: Gehorsam

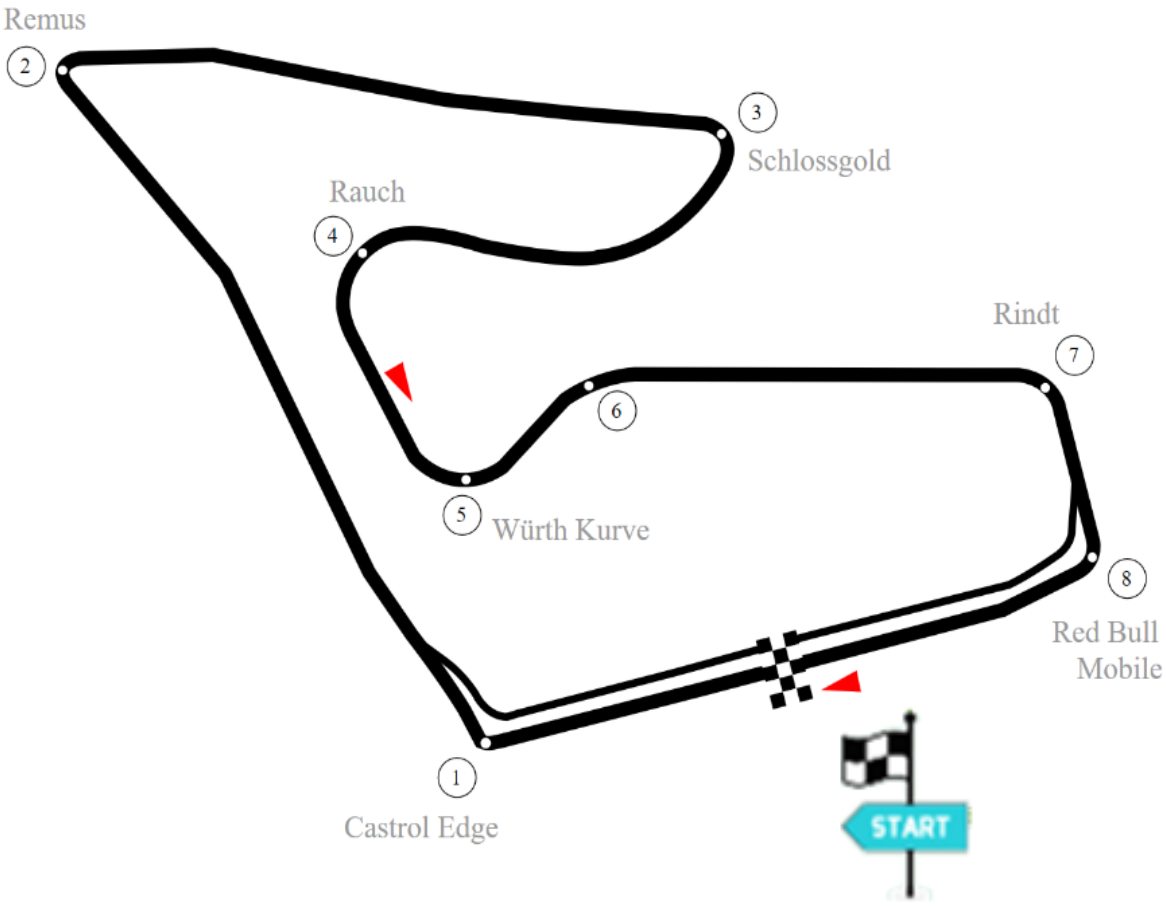
Thymio reagiert auf _____ und die _____ .

Bei mehrmaligem Betätigen der Tasten, beschleunigt oder verlangsamt sich Thymio.

Fertige auf einem Bogen Packpapier eine Zeichnung des **Red Bull Rings in Spielberg** (Steiermark) an. Die **Fahrbahnen** müssen **schwarz** und **mindestens 3 cm** breit sein. Für die Ausführung dieser Aufgabe kannst du eines der bereits in deinen Thymio einprogrammierten Verhaltensmuster verwenden. Überlege welches?

Das brauche ich:
1 Thymio, Vorlage Red Bull-Ring

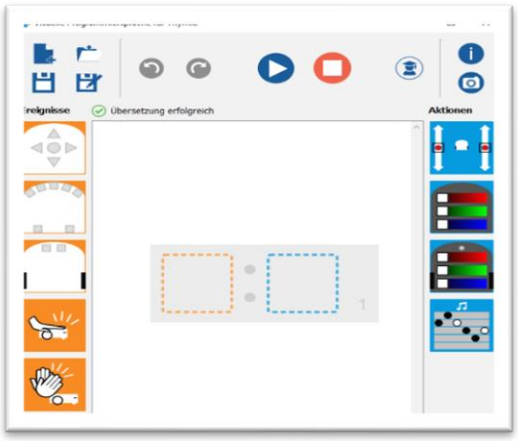
Red Bull Ring in Spielberg (Steiermark)



Thymio: VPL-Programmierung

Dein Thymio bietet dir eine Reihe von Möglichkeiten, ihn zu programmieren. Die einfachste davon ist sicher die Programmierung über eine **visuelle Programmiersprache** (**VPL**)

- ✓ Thymio einschalten und dazugehörigen Wireless-Stick am PC in eine USB-Schnittstelle stecken.
- ✓ **Thymio-Suite** starten und VPL auswählen. (Sollte das Programm nicht auf deinem PC installiert sein, kannst du es unter der folgenden URL downloaden: **https://www.thymio.org/de:start**



Schau dir das folgende Video an. Es ist zwar auf Französisch, aber du bekommst einen ersten Eindruck dafür, wie mit **VPL** umzugehen ist!

VPL-Programmierung
https://bit.ly/2ZqHgxs

Programmiere deinen Thymio so, dass du ihn mit den Sensortasten steuern kannst. Durch Drücken der mittleren Sensortaste soll der Roboter gestoppt werden und eine kurze Melodie abspielen.

Das brauche ich:
1 Thymio + Wireless Stick, PC oder Notebook mit **Thymio-Suite**

Folgende Ereignisse und Aktionen wirst du für dein erstes Programm benötigen:

gedrückte Knöpfe: grau ignorieren, rot: beachten

Motorsteuerung: Einstellung der Geschwindigkeit des linken und rechten Motors durch Verschieben der beiden Punkte mit der Maus

Töne abspielen: Linien geben die Tonhöhe an, die Farben (schwarz, weiß) die Tonlänge

Vor dem Schließen der **VPL-Programmierumgebung,** unbedingt ein eventuell noch laufendes Programm mit der Schaltfläche beenden!

Thymio: VPL-Programierung

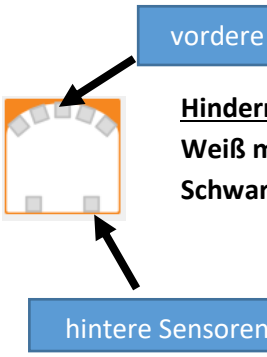
Beim Drücken der **vorderen Sensortaste** beginnt dein Thymio **langsam vorwärts zu fahren**, die oberen **LEDs leuchten** dabei **grün**.

Sobald der **mittlere Fronsensor** ein Hindernis erkennt beginnt er **rechtsherum im Kreis** zu fahren. Die oberen **LEDs leuchten** nun **gelb**.

Durch **Klopfen auf die Oberseite** des Thymio wird dein Roboter **gestoppt** und die **LEDs** beginnen **rot** zu leuchten.

Das brauche ich:
1 Thymio + Wireless Stick, PC oder Notebook mit **Aseba-Studio**

Für die **Sensoren** benötigst du folgendes Ereignis:



Hinderniserkennunggrau → ignorieren
Weiß mit roter Umrahmung →Aktion, wenn Objekt in der Nähe
Schwarz → Aktion, wenn kein Objekt vorhanden

Am PC oder bei RGB-LEDs kann eine Farbe durch ihre Anteile an den drei Primärfarben Rot, Grün und Blau definiert werden. Betrachte den Farbkreis um zu ermitteln, wie du die Farbe gelb einstellen kannst!

Klopfereignis
Wird der Roboter erschüttert (z.B. durch Klopfen auf die Oberfläche, oder anstoßen an ein Hindernis) erfolgt eine Aktion.

Farbauswahl Oberseite: Die unterschiedlichen Farben entstehen durch additive Farbmischung (**RGB**-Farben). Mit der Maus kannst du die drei Farbregler verschieben und dabei gleich die Farbänderung betrachten.

Für Profis:

Schaffst du es, das Programm so zu verändern, dass er zusätzlich auch auf die hintere Sensortaste reagiert. Bei Berührung der Rückwärts-Taste soll dein Thymio rückwärtsfahren. Sobald die beiden hinteren Sensoren ein Hindernis erkennen, beginnt er linksherum im Kreis zu fahren. Die oberen LEDs leuchten nun türkis

.

Thymio: Programmierung mit Blockly

1. Thymio-Suite starten
2. **Blockly**  auswählen
3. Thymio einschalten
4. Wireless Dongle am USB-Anschluss des Computers einstecken.
5. Thymio auswählen
6. Programm starten

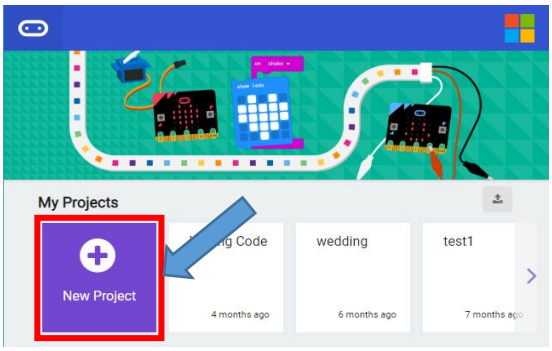
Sobald der **mittlere Knopf** am Thymio **berührt** wird, **fahre vorwärts**. Die oberen **LEDs** sollen **grün** leuchten. Wenn der **mittlere Frontsensor** ein **Hindernis erkennt,** sollen die **Motoren stoppen** und die oberen **LEDs rot** leuchten.

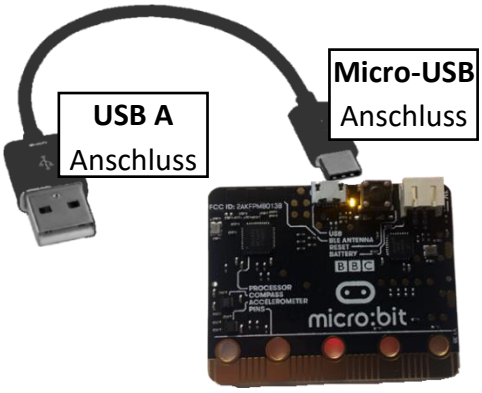
Das brauche ich:
1 Thymio + Wireless Stick, PC oder Notebook mit **Thymio-Suite**

MicroBit: Programmierung im Browser

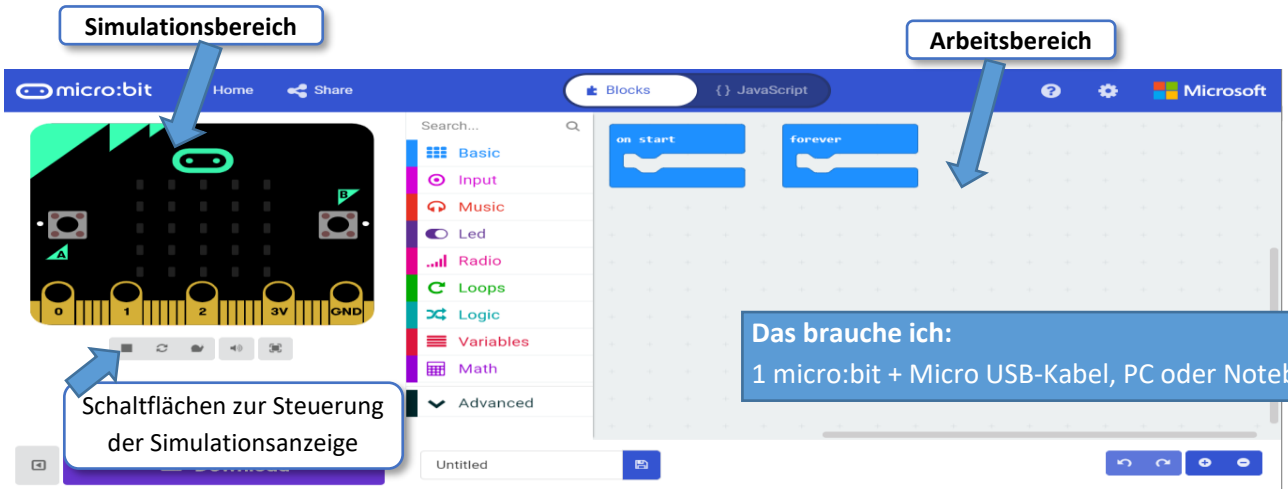
- ✓ Schließe deinen micro:bit mit einem **USB-Kabel** an einen PC oder Notebook an! Der angeschlossene micro:bit wird auf deinem Computer als Laufwerk angezeigt.

- ✓ Starte nun deinen Browser und gib folgende URL ein
  https://makecode.microbit.org/
  und starte ein neues Projekt!





Die Programmieroberfläche mit auswählbaren Programmblöcken wird geöffnet!



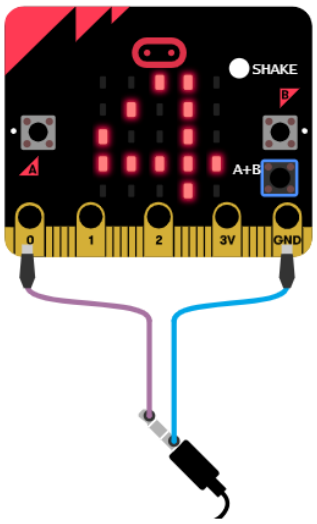
Schreibe ein Programm „**Herzschlag**“, das auf deinem micro:bit ein schlagendes Herz anzeigt!

- ✓ Überprüfe dein Programm im Simulationsbereich und lade es anschließend auf deinen micro:bit

MicroBit: Programmierung im Browser

Schreibe ein Progamm, dass auf deinem micro:bit durch Drücken der Schalter „**A**“, „**B**“, „**AB**“ und durch „**Schütteln**“ unterschiedliche Melodien abspielt! Dabei soll bei jeder Melodie eine andere Ziffer am Display eingeblendet werden! A→(1), B→(2), AB→(3), Schütteln→(4)

Das brauche ich:
1 micro:bit + Micro USB-Kabel, PC oder Notebook, Lautsprecher,
2 Kabel mit Krokodilklemmen, 1 Audiokabel zum Anschluss des Lautsprechers



Achtung: Für die beschriebene Aufgabenstellung sind insgesamt vier Teilprogramme erforderlich. Alle vier Teilprogramme können gemeinsam im Arbeitsbereich abgelegt werden und als eine HEX-Datei heruntergeladen und am micro:bit gespeichert werden!

Beachte: Dein micro:bit hat keine **Lautsprecher** eingebaut. Für die Ausgabe von Tönen, muss daher am micro:bit zusätzlich ein Lautsprecher angeschlossen werden. Im Simulationsbereich wird angezeigt, wie ein Lautsprecher über **Krokodil-Klemmen** und ein Standard **Audiokabe**l an den micro:bit angeschlossen werden muss.

Die erforderlichen **Programmierblöcke** befinden sich in den **Registern** Basic, Input und Music

Krokodilklemme

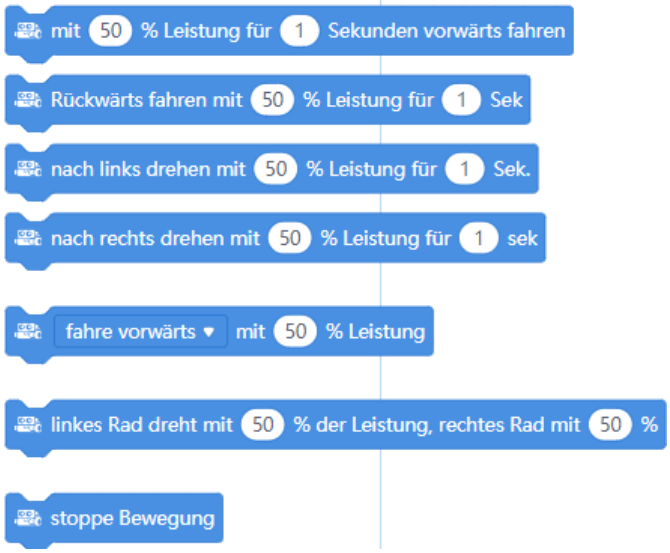
✓ Überprüfe dein Programm im Simulationsbereich und lade es anschließend auf deinen micro:bit

Grundfunktionen und Blöcke

Dein mBot soll beim Anklicken des Fähnchens folgende Aufgaben erledigen:

(1) mit 50% der Leistung für 3 Sekunden vorwärtsfahren.
(2) mit 30% der Leistung für 5 Sekunde rückwärtsfahren.
(3) mit 90% der Leistung für 1 Sekunden nach links drehen.
(4) mit 75% der Leistung für 1 Sekunde vorwärtsfahren dann 2 Sekunden warten und mit 100% der Leistung für 1 Sekunde rückwärtsfahren.

Steuerung der Motoren (Register Aktionen) Aktion



Im Register „Aktionen“ stehen sieben Blöcke zur Steuerung der Elektromotoren bereit.

- ✓ Der Prozentsatz (0% - 100%) bestimmt die Leistung (**Drehgeschwindigkeit**) der Motoren
- ✓ Die ersten 4 Blöcke ermöglichen eine zeitliche Begrenzung der Bewegung
- ✓ Der letzte Block dient zum Unterbrechen der Bewegung beim Verwenden von Blöcken ohne zeitlicher Begrenzung oder beim Erkennen eines Hindernisses

Tipp: Um ein Programm mit dem Fähnchen starten zu können, muss das Ereignis

am Beginn eines Skripts stehen! Probiere auch andere Ereignisse zum Starten aus

Der Befehl fürs Warten befindet sich im Register Steuerung →

Versuche die Skripts sowohl im **Live-Modus** als auch durch **Hochladen** auszführen!

Kommunikation mit der Umwelt

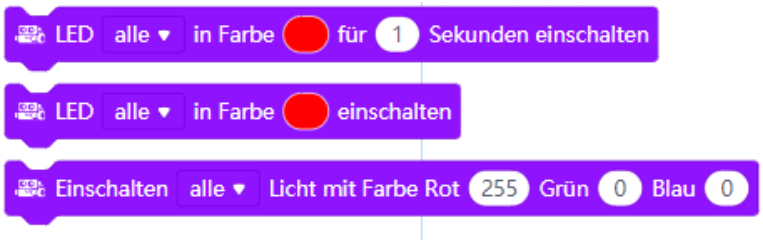
(Aussehen→RGB LEDs)

Dein mBot soll beim Anklicken des Fähnchens folgende Aufgaben erledigen:

(1) beide LEDs leuchten 2 Sekunden rot
(2) die beiden LEDs leuchten in unterschiedlichen Farben
(3) beide LEDs sollen zuerst 1 Sekunde gelb leuchten, dann für 2 Sekunden ausgeschalten werden und dann für drei Sekunden blau leuchten.
(4) die beiden LEDs sollen abwechselnd in 2 unterschiedlichen Farben 5x blinken
Tipp: Verwende dazu den Block „wiederhole Anzahl“ → Register Ereignisse
(5) Programmiere eine Verkehrsampel

✓ **Register** Zeigen **OnBoard-LEDs:**

Im vorderen Bereich der Platine sitzen zwei **RGB-LEDs**. Die beiden LEDs können einzeln angesteuert werden. Die Farbeinstellung erfolgt dabei entweder über **Schieberegler** oder durch die Eingabe der sogenannten **RGB-Werte** (Zahlen von 0-255. → siehe InO-Bot: additive Farbmischung →. Bei den beiden Blöcken ohne zeitliche Begrenzung, müssen die LEDs durch einen Block mit den Werten 0-0-0 ausgeschaltet werden.



Lichtzeichen der Ampel - Rot, Gelb, Grün

kleine Ampelkunde → https://www.wien.gv.at/verkehr/ampeln/ampelkunde.html

✓ Rot bedeutet "Halt": alle Verkehrsteilnehmerinnen und Verkehrsteilnehmer müssen anhalten. Fußgängerinnen und Fußgänger dürfen die Fahrbahn nicht mehr betreten.

✓ Gelb (nichtblinkend) gilt als Zeichen für "Halt" für sich der Kreuzung nähernde Fahrzeuge. Verkehrsteilnehmerinnen und Verkehrsteilnehmer, die sich bereits auf der Kreuzung befinden, müssen diese rasch verlassen.

✓ Gelbblinken bedeutet "Achtung".

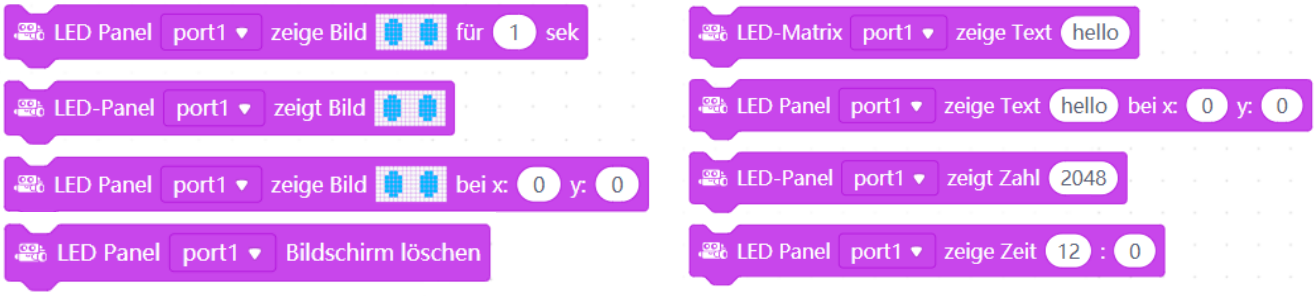
✓ Grün: Grünes Licht gilt für den Fahrzeugverkehr als Zeichen für "Freie Fahrt". Für Fußgängerinnen und Fußgänger bedeutet das grüne Licht der Ampel "Betreten und Überqueren der Fahrbahn erlaubt".

✓ Grünblinken: Am Ende der Grünphase kündigt ein viermaliges Grünblinken das Ende der Grünphase an. Das Befahren beziehungsweise Betreten der Kreuzung ist noch erlaubt.

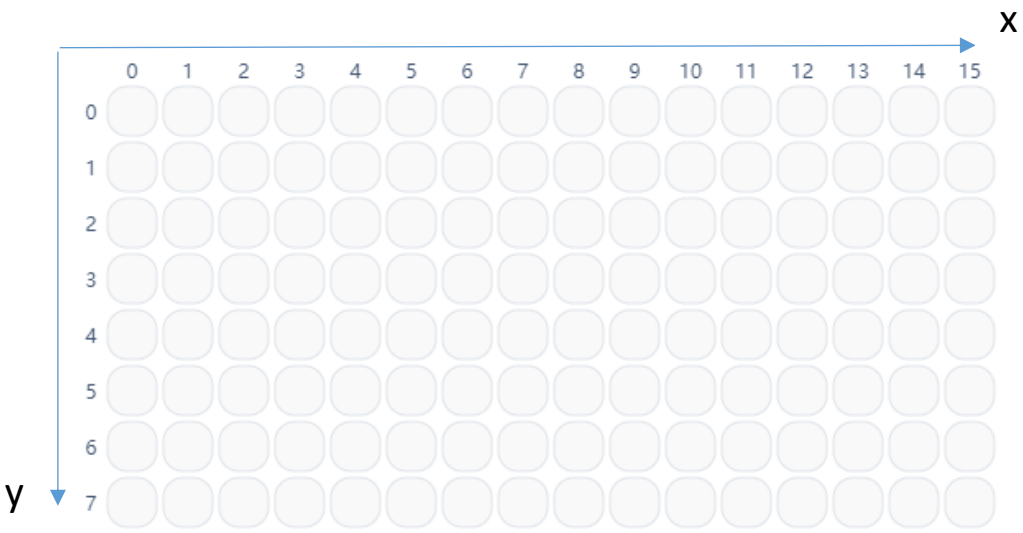
Kommunikation mit der Umwelt

(Anzeigen → LED Matrix Display)

LED-Matrix-Display (Register Aussehen) In der Kategorie „Aussehen“ gibt es hierfür acht Blöcke.

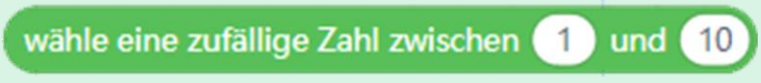


Das LED-Matrix-Display kann an den freien Ports 1 oder 4 angeschlossen werden. Sie besteht aus einer Matrix von 16x8 Pixel (x=0-15, y=0-7, 128 LEDs) die einzeln angesteuert werden können. Es können damit sowohl kurze Texte, Zahlen oder Pixelbilder angezeigt werden.



(1) Zeige für 3 Sekunden dein Lieblingssymbol aus der vorhandenen Auswahl an
(2) Zeichne dein eigens ICON und zeige es für 5 Sekunden am Display
(3) Lasse einen Punkt in der Mitte des Display von oben nach unten wandern
(4) Lasse für jeweils 0.2 Sekunden einen Punkt zufällig am Display aufleuchten

Verwende dazu „**Zufallszahl**“ aus dem Register Operatoren



Kommunikation mit der Umwelt

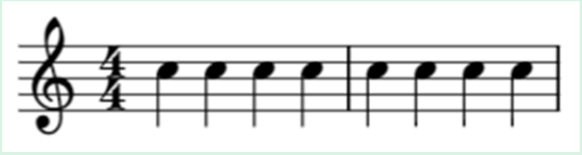
(Audio-Signale)

Die SOUNDs (Töne) befinden sich im Register 

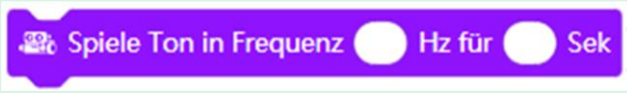
Der **Summer/Buzzer** befindet sich auf der Platine deines mBot und kann Töne erzeugen und über den eingebauten Lautsprecher wiedergeben!

Die Tonhöhe wird dabei über die Frequenz (Anzahl der Schwingungen pro Sekunde) oder über die Notennamen und die Oktavbezeichnung angegeben. Die Tonlänge kann in Sekunden bzw. Schlägen (ganze Note = 1, halbe Note = 0.5, Viertelnote = 0.25, Achtelnote = 0.125) angegeben werden. **Achtung**: Die Bezeichnungen in mBlock stimmen nicht mit den in der Musiklehre verwendeten Oktavbezeichnungen überein!
Kammerton **a**^{1} →mBlock **A4**

(1) Dein mBot soll 8 Viertelnoten C2 spielen.



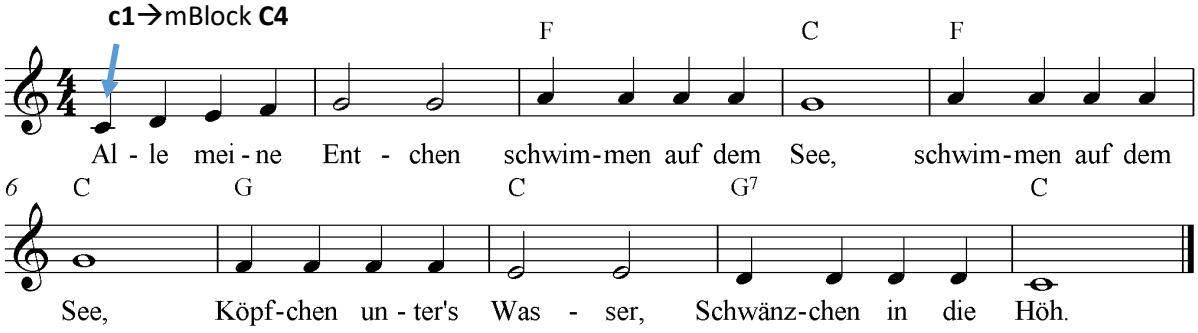
(2) Erzeuge den **Kammerton**, er soll für 3 Sekunden erklingen.

Verwende dazu folgenden Block: 

INFO zum Kammerton → *https://de.wikipedia.org/wiki/Kammerton*

(3) Versuche die Melodie von „Alle meine Entchen zu spielen“ (Beginne mit C4). Zum Schreiben von Noten kannst du den Online-Noteneditor „Scorio“ verwenden! http://www.scorio.com/de_DE/web/scorio/new-score

Alle meine Entchen



Kommunikation mit der Umwelt (Helligkeitssensor)

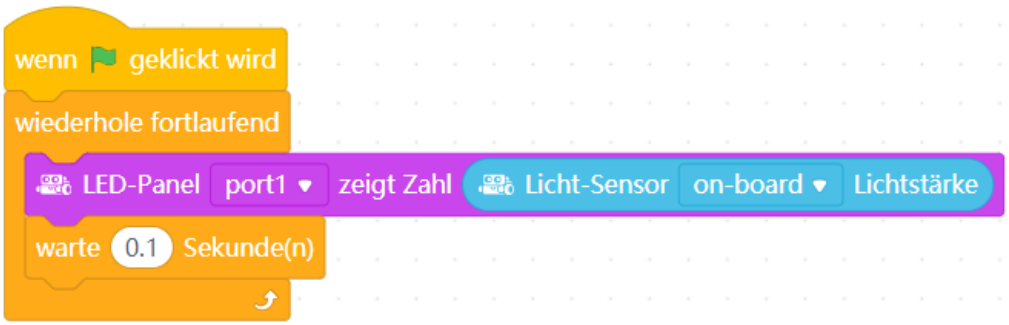
Der Helligkeitssensor befindet sich vorne auf der Platine.

Der zugehörige Block ist im Register 

zu finden.

Wertebereich: **0** (ganz dunkel) bis ca. **1030** (sehr hell)

Mit folgendem Skript wird der aktuelle Wert der Helligkeit abgefragt und auf dem LED-Matrix-Modul angezeigt:



Setze den Sensor unterschiedlichen Helligkeiten (abdecken; aufdecken) aus und beobachte die angezeigten Werte. Abhängig von der Helligkeit, sollen die LEDs unterschiedlich leuchten.

Kommunikation mit der Umwelt (Ultraschallsensor)

Der Ultraschallsensor (Abstandssensor) befindet sich vorne am Roboter und sollte mit dem Port 3 verbunden sein.

Der zugehörige Block ist im Register Fühlen zu finden.

Die Entfernungen werden in cm angegeben

Mit dem folgendem Skript, wird die Entfernung am LED-Matrix-Display anzeigt.

Schreibe ein Programm das die jeweilige Entfernung auf dem LED-Matrix-Display anzeigt und abhängig von der Entfernung einen hohen Ton (Abstand < 10) bzw. einen tiefen Ton (Abstand > 10) ausgibt. Solange kein Hindernis in der Nähe ist, soll dein mBot vorwärts fahren. Taucht ein Hindernis auf, soll er bei einem Abstand <10 stoppen und die LEDs sollen rot leuchten, während der Vorwärtsfahrt sollen die LEDs grün leuchten.

Kommunikation mit der Umwelt
(Linienverfolgungssensor)

Der Linienverfolgesensor befindet sich vorne an der Unterseite deines mBots und sollte an Port 2 angeschlossen sein. Der Linienverfolgungssensor besteht aus zwei Sensoren (links/rechts), die unabhängig voneinander überprüfen, ob der Untergrund hell oder dunkel ist.

Es gibt zwei Blöcke in der Kategorie Fühlen

Rückgabewerte:

✓	0	li: schwarz	re: schwarz → Roboter ist in der Spur
✓	1	li: schwarz	re: weiß → Roboter zu weit rechts
✓	2	li: weiß	re: schwarz→ Roboter zu weit links
✓	3	li: weiß	re: weiß → Roboter ist neben der Spur

Die **Entfernungen** werden in cm angegeben

Mit dem folgendem Skript werden die **Sensorwerte am LED-Matrix-Display** anzeigt.

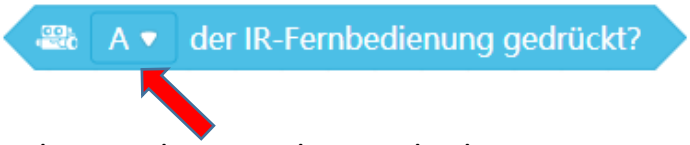
Schreibe ein Programm das deinen mBot auf einer schwarzen Linie dahinfahren lässt.

Steuerung mit der Fernbedienung

Der Infrarot-Sender befindet sich in der Fernbedienung, der Empfänger auf der Platine.

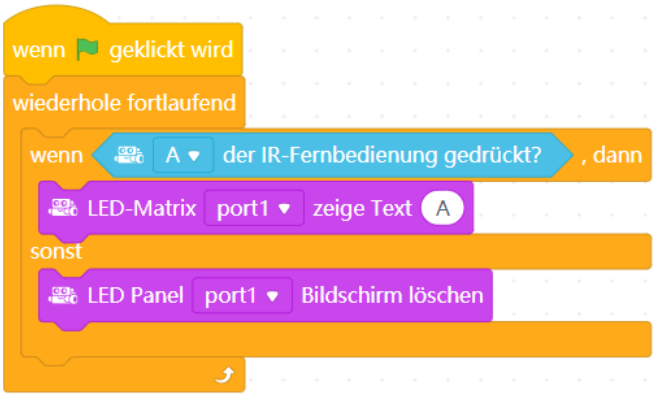
Der zugehörige Block ist unter 

zu finden.



Hier kann jede Taste der Fernbedienung eingestellt werden.

Teste die Fernbedienung mit folgendem Skript. Das LED-Matrix-Display soll „A“ zeigen, wenn die Taste A gedrückt wird, sonst soll das Display leer bleiben.



Schreibe ein Programm, mit dem du deinen Roboter über die Fernbedienung steuern kannst.

Achtung: Um den mBot über die Fernbedienung steuern zu können, muss die Firmware der Fabrik(06.01.009) eingespielt sein!

Taster

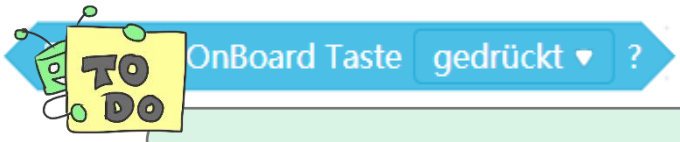
Auf dem mBot befindet sich ein Taster, mit dem im Auslieferungszustand die drei verschiedenen Modi

- ✓ **Remote-Control**
- ✓ **Linienverfolgung**
- ✓ **Hinderniserkennung**

eingestellt werden können.

Der zugehörige Block ist unter 

zu finden.



Teste den Taster mit folgendem Skript. Beim Drücken des Tasters (OnBoard-Taste) soll auf dem LED-Matrix-Modul „ups“ zu lesen sein. Wird der Taster losgelassen, soll das Display leer bleiben.

Achtung: Um den mBot über den **Taster** steuern zu können, muss die Firm Firmware der Fabrik(06.01.009) eingespielt sein!

Nun hast du die wichtigsten Funktionen deines Roboters kennengelernt. Der Verwirklichung eigener Projekte steht nun nichts mehr im Weg. Lass deiner Phantasie freien Lauf.

Viel Erfolg!

Kalibrierung der Motoren

Der Roboter sollte während der Vor- oder Rückfahrt zwischen den beiden Begrenzungslinien bleiben!

Zum Überprüfen den Modus „**Gehorsam**“ rosa verwenden!