LMZ LANDESMEDIENZENTRUM BADEN-WÜRTTEMBERG

COMTHINK

LERNEINHEIT:
EINFÜHRUNG IN ALGORITHMEN MIT DEM COMTHINK-SET

www.lmz-bw.de

IMPRESSUM

Herausgeber und Bezugsadresse

Landesmedienzentrum
Baden-Württemberg
Vertreten durch Direktor Michael Zieher
Rotenbergstraße 111
70190 Stuttgart

Telefon: 0711 4909-6399
E-Mail: comthink@lmz-bw.de

Dieses Materialpaket ist Teil des Projekts Computational Thinking im Sekundarbereich I

Projektleitung:
Juanjuan Jia

Autor:
Timo Münzing

Redaktion:
ComThink-Team LMZ

Gestaltung:
Bianca Dreja Grafikdesign

Stuttgart, April 2026

Bilder und Grafiken

Sämtliche in diesem Dokument verwendeten Fotos stammen vom Landesmedienzentrum Baden-Württemberg. Die Bilder der Präsentation (außer der Fotos der Modelle) wurden KI-gestützt generiert (erstellt mit ChatGPT).

Internetseiten dritter Anbieter / Links

Soweit Inhalte dieser Materialien auf externe Internetseiten verweisen, hat das LMZ auf den Inhalt dieser Seiten keinen Einfluss. Diese Internetseiten unterliegen der Haftung der jeweiligen Betreiber. Das LMZ hat bei der erstmaligen Verknüpfung der externen Links die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu diesem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Eine ständige inhaltliche Überprüfung der externen Links ist ohne konkrete Anhaltspunkte einer Rechtsverletzung nicht möglich. Bei Kenntnis von Rechtsverstößen werden derartige externe Links unverzüglich gelöscht.

LERNEINHEIT ZUR EINFÜHRUNG IN ALGORITHMEN MIT DEM COMTHINK-SET

In dieser Lerneinheit erlernen Schülerinnen und Schüler schrittweise grundlegende Programmierstrukturen mithilfe des STEM Coding Max-Sets von Fischertechnik und der zugehörigen App. Als Programmiersprache dient Scratch, das durch seine visuelle, blockbasierte Oberfläche einen motivierenden und leicht zugänglichen Einstieg bietet.

Die Einheit umfasst drei Doppelstunden (alternativ sechs Einzelstunden) und wird durch optionale Projektideen zur Vertiefung und kreativen Anwendung ergänzt.

Didaktisch ist die Programmierung in eine Rahmengeschichte eingebettet: Vier Jugendliche stranden auf einer Insel und müssen technische Geräte programmieren, um zu entkommen. Die Lernenden übernehmen dabei selbst die Rolle der Problemlöserinnen und Problemlöser. Dieses Storytelling-Element schafft eine immersive Lernumgebung, erleichtert den Wiedereinstieg bei Einzelstunden und wird durch eine Lehrkraftpräsentation im Plenum vermittelt.

Abbildung 1:
Darstellung des Storytellings

Das Arbeitsmaterial für die Lernenden besteht überwiegend aus einem „Buch“, das die Jugendlichen in der Geschichte finden. Dieses Buch enthält die systematische Einführung in die grundlegenden Strukturen der Programmierung.

Abbildung 2:
Ausschnitt aus dem Buch, anhand dessen die Lehrkraft die Programmierstrukturen einführen und die Lernenden selbstständig auf diese zurückgreifen können.

Abbildung 3:
Beispiel für einen Arbeitsauftrag.

Für die Durchführung wird pro Gruppe ein STEM Coding Max-Set sowie ein digitales Endgerät benötigt. Die Lernmaterialien können digital genutzt oder bei Bedarf in ausgedruckter Form bereitgestellt werden. Optional kann ein zweites Endgerät für die Anzeige des Arbeitsmaterials eingesetzt werden.

Diese Lerneinheit verbindet so technische Grundlagen der Programmierung mit kreativer Erzählung, um die Lernenden praxisnah, motivierend und nachhaltig an das Thema heranzuführen.

01 ZIELGRUPPE & UMFANG

- **Zielgruppe:** Informatik ab Klasse 7, Technik ab Klasse 7
- **Umfang:** Drei Doppel- oder sechs Einzelstunden
- **Optional:** Erweiterung durch zusätzliche Projekte

02 LERNZIELE

Die Lernenden...

- … nutzen die visuelle Programmiersprache Scratch, um Problemlöseaufgaben umzusetzen.
- … verstehen die algorithmischen Grundbausteine Sequenz, Wiederholung / Schleife, Verzweigung und Bedingung und wenden diese in eigenen Programmen an.
- … entwickeln Problemlösestrategien, indem sie technische Herausforderungen innerhalb der Rahmengeschichte analysieren und schrittweise umsetzen.
- … kooperieren im Team, indem sie Aufgaben innerhalb der Gruppe abstimmen und gemeinsam Lösungen erarbeiten.

03 BEZUG ZUM BILDUNGSPLAN

Die vorliegende Einheit orientiert sich am Bildungsplan 2016 Informatik, Aufbaukurs Klasse 7 (Sekundarstufe I). Insbesondere der Bereich „3.1.2 Algorithmen“ wird sehr stark durch die Lerneinheit abgedeckt. Aber auch Aspekte aus dem Bereich „3.1.1 Daten und Codierung“ werden behandelt. Die Zahlen in den Klammern entsprechen der Zuordnung der beschriebenen Tätigkeiten zu den Lernzielen im Bildungsplan.

3.1.2 Algorithmen

- Es werden die algorithmischen Grundbausteine Anweisung, Sequenz, Schleife / Wiederholung, Verzweigung und Bedingung erarbeitet und angewandt. (1)(2)
- Die Lernenden müssen Algorithmen zu gegebenen Problemstellungen entwerfen und in einer Programmierumgebung implementieren. Dabei setzen sie die algorithmischen Grundbausteine zielorientiert ein. (4)(5)
- Insbesondere bei der Fehlersuche müssen die Lernenden den Code schrittweise untersuchen, um ihn zu korrigieren. (7)
- Während der optionalen Projektphase werden Variablen eingeführt und angewandt. (4)

3.1.1 Daten und Codierung

- Die Schülerinnen und Schüler lernen den Morsecode kennen und kodieren ihn selbstständig auf einer LED. (1)(2)(3)

04 MATERIALBEDARF

- Ein STEM Coding Max-Set je Team
- Ein digitales Endgerät mit installierter App STEM Suite je Team
- Präsentation mit Projektionsmöglichkeit im Lernraum
- Material für Lernende: Das gefundene Buch
- Material für Lernende: Das gefundene Buch (s/w-Druckversion)

05 QUELLEN

- Gymnasium – Aufbaukurs Informatik – Klasse 7
  https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/GYM/INF7
- Screenshots wurden von Timo Münzing erstellt
- Klebebandgrafik im Buch des Schülermaterials wurde von Pixabay entnommen
  https://pixabay.com/de/illustrations/ai-generiert-klebeband-band-gelb-8885655/
- Grafische Elemente innerhalb der Unterrichtseinheit für das Storytelling wurden innerhalb von ChatGPT erzeugt

06 VERLAUFSPLAN

Der vorliegende Verlaufsplan ist in Doppelstunden beschrieben, kann aber jeweils an den blauen Zeilen in Einzelstunden aufgespaltet werden.

Doppelstunde 1 / Einzelstunde 1+2

Phase / Zeit / Sozialform	Inhalt	Material / Hinweise
Einstieg / 10 min / Plenum	Lehrkraft kündigt an, dass die Lernenden in den kommenden Stunden mit einem Programmier- und Robotik-Set arbeiten werden. Sie führt in die Geschichte der vier Jugendlichen ein, die als Rahmenhandlung dient. Die Startseite der Präsentation ist bereits projiziert. Lehrkraft führt an den Folien 2-6 die Story ein und führt zur ersten Aufgabe hin.	Präsentation Folie 2-6

Phase / Zeit / Sozialform	Inhalt	Material / Hinweise
Erarbeitung / 35 min / Plenum	Lehrkraft verteilt die STEM Coding Max-Sets an die Lernenden. Lernende erarbeiten gemeinsam mit der Lehrkraft anhand der Einstiegsmodelle (unter „Aufgabe auswählen“) in der STEM Suite App das grundlegende Vorgehen beim Einsatz der Sets. Lehrkraft entscheidet in Abhängigkeit des Leistungsstands der Gruppe, wie weit innerhalb der Einstiegsmodelle gearbeitet wird. Seite 23 sollte hierbei nicht überschritten werden – ein früherer Übergang in die nächste Phase ist möglich, sofern die folgenden Punkte bereits behandelt wurden: • das Verbinden von Akku und RX-Controller, • das Anschließen von LED und Taster, • sowie das Überspielen eines Programms auf den Controller.	**STEM Suite App** **STEM Coding Max Set**
Bei Aufteilung in Einzelstunden: Vorherige Phase um 5 Minuten kürzen zum Aufräumen, in der nächsten Stunde Zeitaufwand zum Austeilen der Sets beachten.		
Erarbeitung / 10 min / Plenum	Story wird mithilfe der Folien 7 und 8 fortgeführt. Auf Folie 8 kann die Lehrkraft, abhängig vom Niveau und Lernfortschritt der Gruppe, das bisher Erarbeitete wiederholen oder ergänzende Details zur Programmierung vorstellen. Anschließend stellt sie den Arbeitsauftrag (Folie 9) vor, liest ihn gegebenenfalls laut vor und entlässt die Lernenden in die Freiarbeitsphase.	Präsentation Folie 7-9
Übung / 30 min / Gruppenarbeit	Lernende bearbeiten in ihren Teams die gestellten Aufgaben selbstständig. Lehrkraft agiert beratend, unterstützt bei Fragen und lässt sich von den Teams regelmäßig Zwischenergebnisse oder fertige Lösungen präsentieren. Zur Unterstützung steht zusätzlich Seite 1 aus dem Material für Lernende zur Verfügung.	**STEM Suite App** **STEM Coding Max Set** Material für Lernende S.1
Abschluss / 5 min / Plenum	Aufräumen der Sets	**STEM Coding Max Set**

Doppelstunde 2 / Einzelstunde 3+4

Phase / Zeit / Sozialform	Inhalt	Material / Hinweise
Einstieg / 10 min / Plenum	Story wird mithilfe der Folien 10 und 11 fortgeführt. Im Rahmen der Handlung wird der Morsecode sowie seine Bedeutung erläutert. Hierbei kann Vorwissen von einzelnen Lernenden eingebunden werden. Lehrkraft führt auf Folie 11 in Abhängigkeit des Leistungsniveaus der Lernenden die Kontrollstruktur „Wiederholung“ ein. Lehrkraft stellt die Aufgaben auf Folie 12 vor, beantwortet offene Fragen und entlässt die Lernenden in die Freiarbeitsphase.	Präsentation Folie 10-12
Erarbeitung / 35 min / Plenum	Lernende bearbeiten in ihren Teams die gestellten Aufgaben selbstständig. Lehrkraft begleitet den Prozess beratend, unterstützt bei technischen oder inhaltlichen Fragen und lässt sich regelmäßig Zwischenergebnisse oder fertige Lösungen präsentieren. Zur Unterstützung steht zusätzlich Seite 2 aus dem Material für Lernende zur Verfügung.	**STEM Suite App** **STEM Coding Max Set** Material für Lernende S.2
Bei Aufteilung in Einzelstunden: Vorherige Phase um 5 Minuten kürzen zum Aufräumen, in der nächsten Stunde Zeitaufwand zum Austeilen der Sets beachten.		
Erarbeitung / 10 min / Plenum	Story wird mit der Folien 13 fortgeführt. Lehrkraft erläutert anhand von Folie 14 die Kontrollstruktur „Verzweigung“ und erklärt die Besonderheiten des Tasters im Programmierkontext. Arbeitsauftrag auf Folie 15 wird vorgestellt und offene Fragen werden beantwortet. Anschließend werden die Lernenden in die Freiarbeitsphase entlassen.	Präsentation Folie 13-15

Phase / Zeit / Sozialform	Inhalt	Material / Hinweise
Übung / 30 min / Gruppenarbeit	Die Lernenden bearbeiten die gestellten Aufgaben in ihren Teams weitgehend selbstständig. Die Lehrkraft begleitet den Arbeitsprozess beratend, unterstützt bei technischen oder inhaltlichen Fragen und lässt sich regelmäßig Zwischenergebnisse oder fertige Lösungen präsentieren. Zur Unterstützung steht zusätzlich Seite 3 aus dem Material für Lernende zur Verfügung.	**STEM Suite App** **STEM Coding Max Set** Material für Lernende S.3
Abschluss / 5 min / Plenum	Aufräumen der Sets	STEM Coding Max Set

Doppelstunde 3 / Einzelstunde 5+6

Phase / Zeit / Sozialform	Inhalt	Material / Hinweise
Einstieg / 15 min / Plenum	Story wird mit der Folie 16 weiterentwickelt und zum Abschlussprojekt hingeführt. Auf Folie 17 wird den Lernenden das Ziel der heutigen Stunde aufgezeigt: Sie programmieren ein eigenes Spiel. Im anschließenden Plenum werden gemeinsam wichtige Aspekte für die Umsetzung erarbeitet, wie das bisherige Model erweitert werden muss.	Präsentation Folie 16-17
Erarbeitung / 30 min / Plenum	Die Lernenden bearbeiten die gestellten Aufgaben in ihren Teams weitgehend selbstständig. Die Lehrkraft begleitet den Arbeitsprozess unterstützend und beratend, beantwortet Fragen und lässt sich regelmäßig Zwischenergebnisse oder fertige Lösungen präsentieren. Zur Unterstützung steht zusätzlich Seite 4 aus dem Material für Lernende zur Verfügung, auf das die Lehrkraft auch auf Seite 18 der Präsentation eingehen kann.	**STEM Suite App** **STEM Coding Max Set** Material für Lernende S.4 Präsentation Folie 18

Phase / Zeit / Sozialform	Inhalt	Material / Hinweise
Bei Aufteilung in Einzelstunden: Vorherige Phase um 5 Minuten kürzen zum Aufräumen, in der nächsten Stunde Zeitaufwand zum Austeilen der Sets beachten.		
Überleitung / 10 min / Gruppenarbeit	Den Lernenden wird die Beispiellösung auf Seite 19 der Präsentation demonstriert. Verschiedene Lösungsansätze werden diskutiert. Diejenigen Gruppen, die noch keine funktionsfähige Konstruktion haben, bauen diese nun anhand der Musterlösung nach (Material für Lernende, S.4), die auch an alle ausgeteilt wird, die diese noch nicht erhalten haben. Die anderen Gruppen erhalten die Aufgabe, ihr Spiel für drei Spieler zu erweitern.	Präsentation Folie 19
Übung / 25 min / Gruppenarbeit	Die Lernenden bearbeiten die gestellten Aufgaben in ihren Gruppen weitgehend selbstständig. Die Lehrkraft begleitet den Arbeitsprozess beratend und unterstützend, beantwortet Fragen und lässt sich regelmäßig Zwischenergebnisse oder fertige Lösungen präsentieren.	**STEM Suite App** **STEM Coding Max Set** Material für Lernende S.4
Abschluss / 10 min / Plenum	Mithilfe der Folie 20 wird die Story gemeinsam mit der Lerngruppe abgeschlossen. Die Schülergruppen räumen ihre Sets auf.	Präsentation Folie 20 **STEM Coding Max Set**

Erweiterung (Folgestunden)

Phase / Zeit / Sozialform	Inhalt	Material / Hinweise
Bei Aufteilung in Einzelstunden: Vorherige Phase um 5 Minuten kürzen zum Aufräumen, in der nächsten Stunde Zeitaufwand zum Austeilen der Sets beachten.		
Einstieg/ 10 min/ Plenum	Story wird mit der Folie 21 fortgeführt. Lehrkraft entscheidet, abhängig vom Niveau und Arbeitsstand der Lernenden, über die konkrete Ausgestaltung der nächsten Arbeitsphase: • eine vollständig freie Projektarbeit, bei der die Lernenden eigenständig Ideen entwickeln und umsetzen, oder • eine strukturierte Besprechung einzelner Projekt-ideen im Plenum, um die Umsetzung gezielt zu unterstützen.	Präsentation Folie 21
Übung / Dauer frei wählbar / Gruppenarbeit	Die Lernenden nutzen die STEM Coding Max-Sets, um ihre eigenen Projektideen praktisch umzusetzen. Ein Beispiel für einen Projektauftrag ist in Abbildung 4 dargestellt.	**STEM Suite App** **STEM Coding Max Set** Material für Lernende S.5-8

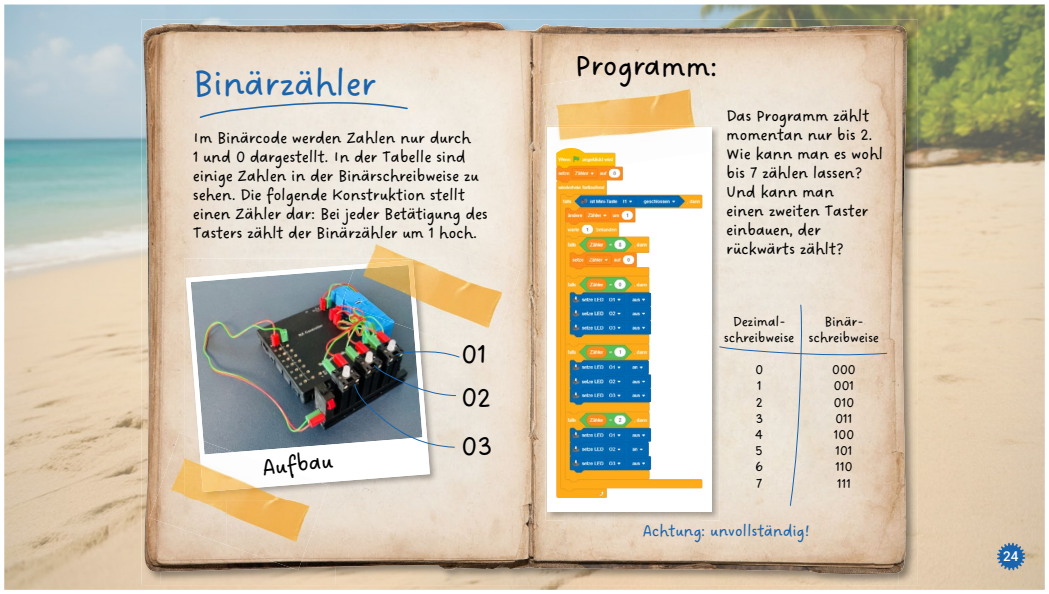
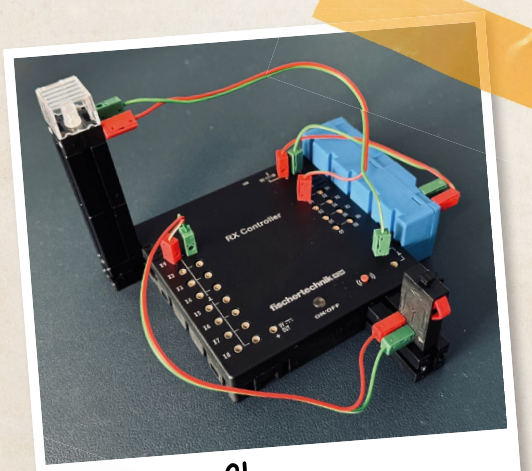


Abbildung 4:
Beispiel für eine Projektseite aus dem Buch. Die hier vorgestellten Konstruktionen sind deutlich komplexer als die vorherigen und geben Ideen (blau geschriebener Text), wie die Projekte weiterentwickelt werden können.

07 MATERIAL FÜR LERNENDE – DAS GEFUNDENE BUCH

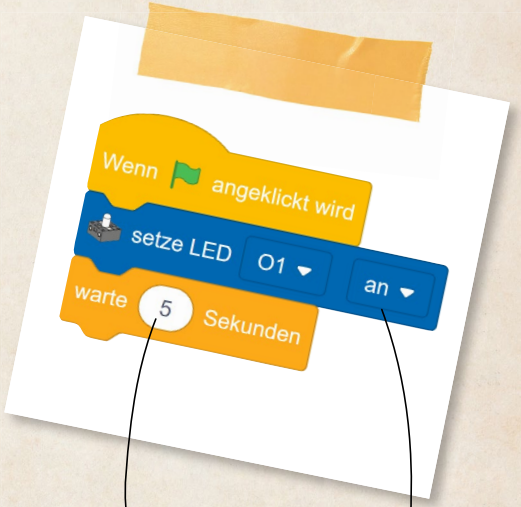
Signallampe

Die folgende Konstruktion lässt beim Start die Lampe für 5 Sekunden leuchten.



Aufbau

Programm:



Einstellungen, die man Anweisungen mitgeben kann, nennt man Parameter

Einzelne Blöcke nennt man Anweisungen oder Befehle

AUFTRAG 1:

Baut eine Signallampe und bringt die Lampe zum Leuchten.

AUFTRAG 2:

Versucht nun das folgende Signal auf der Lampe anzeigen zu lassen: 3 Sekunden an - 5 Sekunden aus - 1 Sekunde an - 4 Sekunden aus - 7 Sekunden an.

AUFTRAG 3 (FÜR SCHNELLE):

Fügt der Konstruktion weitere LEDs hinzu. Lasst die LEDs erst abwechselnd und dann alle gemeinsam leuchten. Erfindet ein eigenes, möglichst komplexes, Leuchtschema!

Wiederholung von Signalen

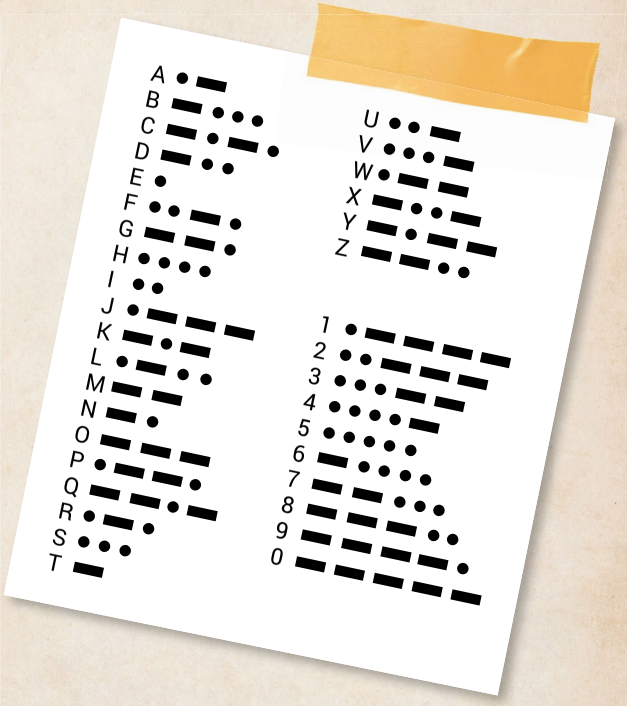
Folgendes Programm schaltet die Signallampe 4-mal hintereinander eine Sekunde an und eine Sekunde aus.

Programm:

Diese Struktur nennt man eine **Wiederholung** oder eine **Schleife**.

Ohne die Sekunde Pause würde die Wiederholung sofort starten und die LED direkt wieder leuchten.

Morsecode



Morsecode ist eine Art „Geheimsprache" aus kurzen und langen Signalen (oft dargestellt als Punkte und Striche).

AUFTRAG 1:

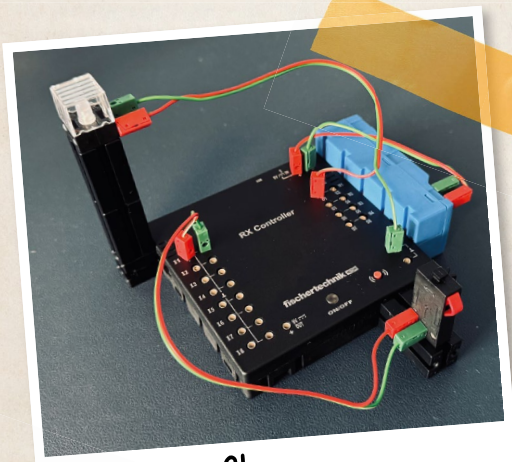
Programmiert ein SOS-Signal mit eurer Signallampe. Lasst das Signal dauerhaft abspielen.

AUFTRAG 2 (FÜR SCHNELLE):

Programmiert mit eurer Signallampe ein Wort oder einen kurzen Satz. Lasst anschließend jemanden aus einer anderen Gruppe erraten, was ihr gesendet habt.

Signallampe durch Taster steuern

Die folgende Konstruktion lässt die Lampe immer dann leuchten, wenn der Taster gedrückt wird.



Aufbau

Programm:

Das Programm soll immer wieder abfragen, ob der Taster gedrückt ist. Daher benötigen wir eine **fortlaufende Wiederholung** um das ganze Programm.

Das ist eine **Bedingung.**

Solche Strukturen nennt man **Verzweigungen.** Abhängig von der Bedingung wird nur ein Zweig der Struktur ausgeführt.

AUFTRAG 1:

Baut einen Taster ein, bei dessen Betätigung 5-mal nacheinander das SOS-Signal abgespielt wird..

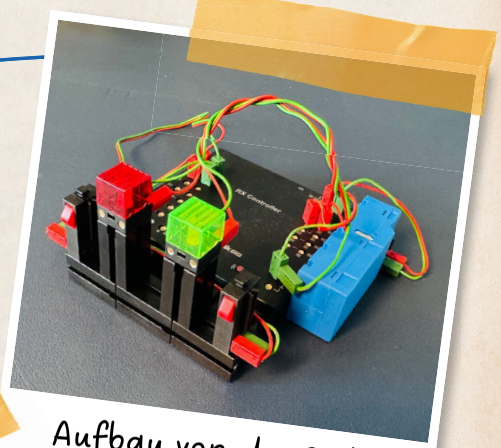
AUFTRAG 2 (FÜR SCHNELLE):

Baut einen zweiten Taster ein. Programmiert die Konstruktion so, dass ihr ja und nein morsen könnt und unterhaltet euch damit.

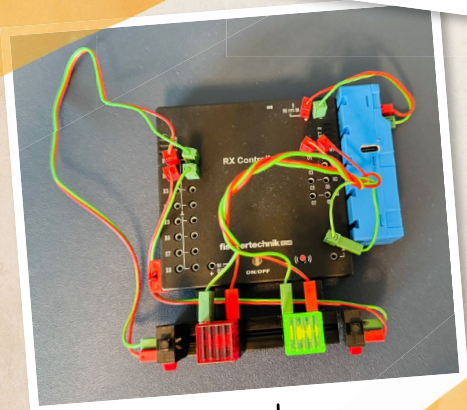
AUFTRAG 3 (FÜR GANZ SCHNELLE):

Baut und programmiert die Konstruktion so, dass jeder Taster ein anderes Wort ausgibt. Könnt ihr vielleicht durch einen Trick mit einem Taster auch zwei verschiedene Wörter auswählen?

Eigenes Spiel – Buzzer



Aufbau von der Seite



Aufbau von oben

Programm:

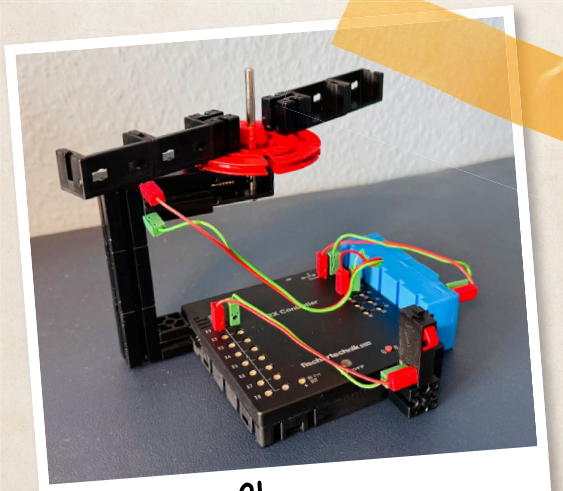
Falls Taster 1 **gedrückt** ist, wird die erste LED für 5 Sekunden eingeschaltet. Dadurch kann nicht gleichzeitig die zweite LED aktiviert werden.

Danach wird dasselbe für Taster 2 geprüft.

Ob man wohl noch mehr Buzzer programmieren kann? Oder vielleicht sogar eine Art Pfeil, der zu dem Spieler zeigt, der schneller ist?

Programm, um die Geschwindigkeit eines Motors zu verändern

Die folgende Konstruktion besteht aus einem Motor mit Propeller und einem Taster. Durch Betätigung des Tasters wird die Geschwindigkeit des Motors erhöht.



Aufbau

Programm:

In Variablen können Werte gespeichert werden. Hier wird der Variablen Geschwindigkeit der Wert Null zugewiesen.

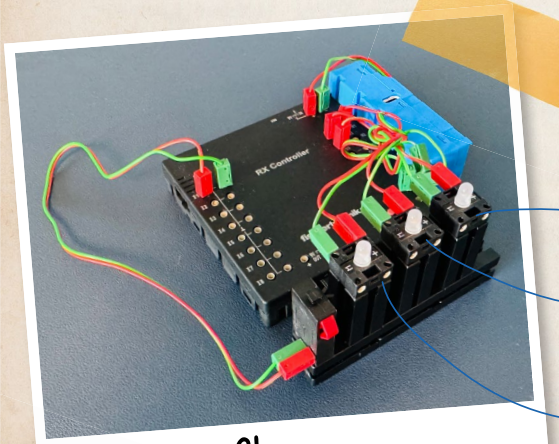
Bei jedem Durchgang wird die Geschwindigkeit des Motors auf den Wert der Variablen gesetzt.

Wenn der Taster gedrückt ist, wird die Geschwindigkeit um 50 erhöht, aber nie über 512 (Maximalwert).

Ob man noch einen zweiten Taster einbauen kann, der die Geschwindigkeit verlangsamt? Oder den Motor in die andere Richtung drehen lässt?

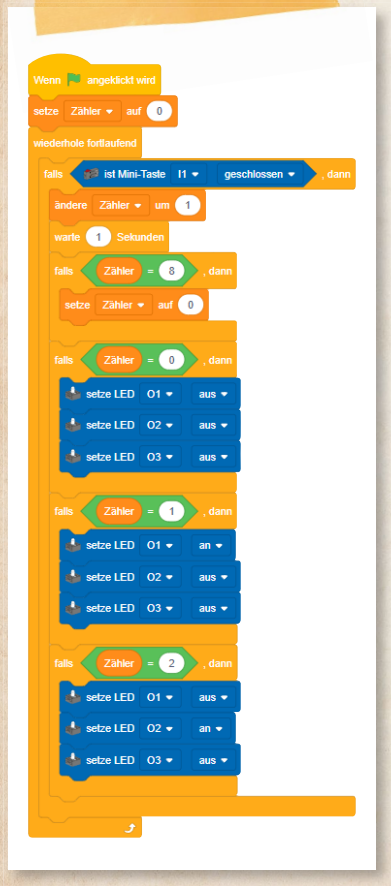
Binärzähler

Im Binärcode werden Zahlen nur durch 1 und 0 dargestellt. In der Tabelle sind einige Zahlen in der Binärschreibweise zu sehen. Die folgende Konstruktion stellt einen Zähler dar: Bei jeder Betätigung des Tasters zählt der Binärzähler um 1 hoch.



Aufbau

Programm:

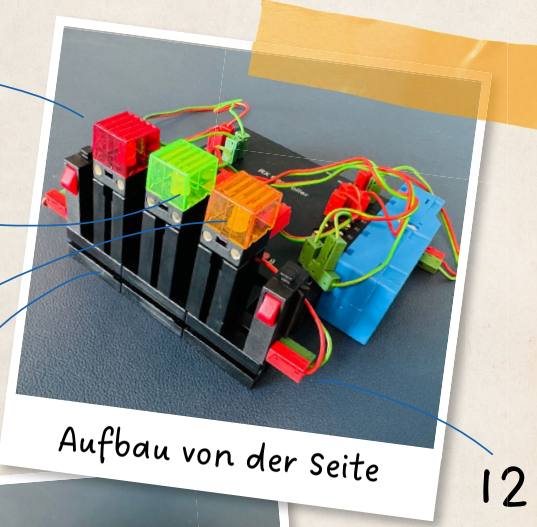


Das Programm zählt momentan nur bis 2. Wie kann man es wohl bis 7 zählen lassen? Und kann man einen zweiten Taster einbauen, der rückwärts zählt?

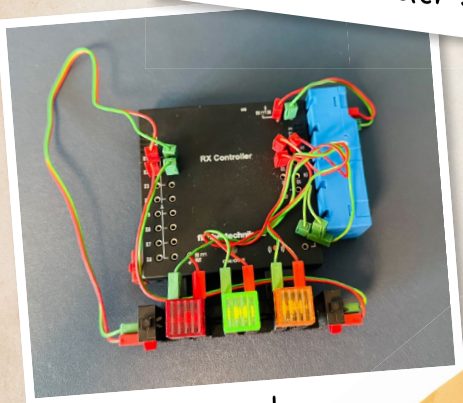
Dezimal-schreibweise	Binär-schreibweise
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Achtung: Programm noch unvollständig!

Reaktionsspiel



Aufbau von der Seite



Aufbau von oben

Folgendes Verhalten soll dem Spiel einprogrammiert werden:

- Nach einer zufälligen Wartezeit von bis zu 5 Sekunden leuchtet die mittlere LED auf.
- Wer nun als Erstes seinen Taster drückt, erhält einen Punkt – seine eigene LED leuchtet kurz auf.
- Sobald jemand 5 Punkte erreicht, gewinnt der Spieler oder die Spielerin. Das wird durch ein Blinken der mittleren LED und das dauerhafte Leuchten der LED des zugehörigen Spielers angezeigt.

Reaktionsspiel

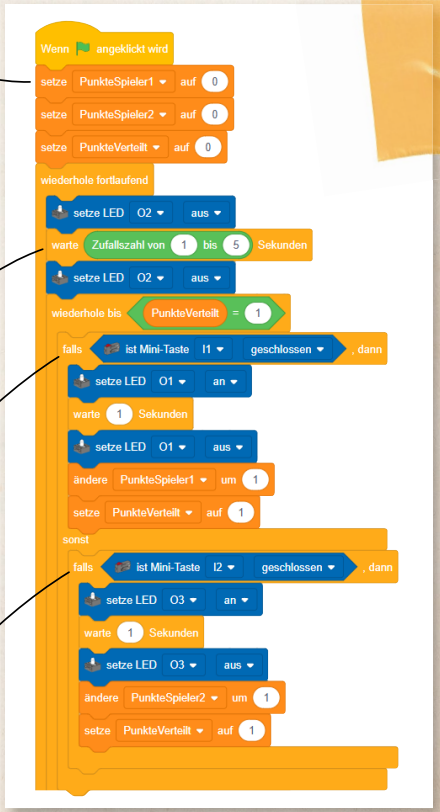
Variablen für Punkte der Spieler und Testvariable, ob Punkt verteilt wurde

LED 2 wird nach zufälliger Zeit eingeschaltet

Test ob Spieler 1 den Taster betätigt hat, falls ja, Signal an LED 1 und Verteilung der Punkte

Test ob Spieler 2 den Taster betätigt hat

Programm:

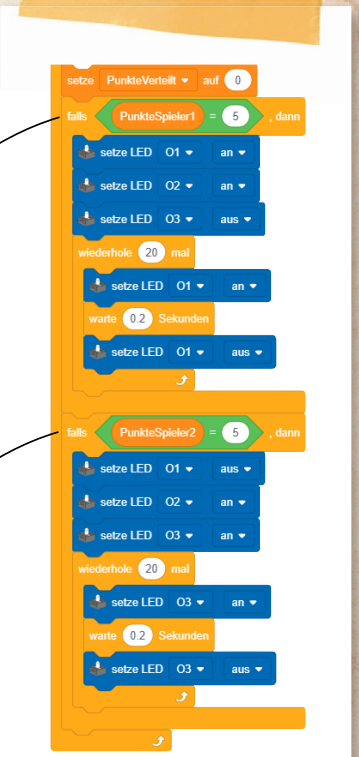


Oben rechts geht´s weiter...

Hier geht´s weiter...

Falls Spieler 1 genug Punkte hat, wird er durch die LEDs als Sieger angezeigt

Falls Spieler 2 genug Punkte hat, wird er als Sieger angezeigt

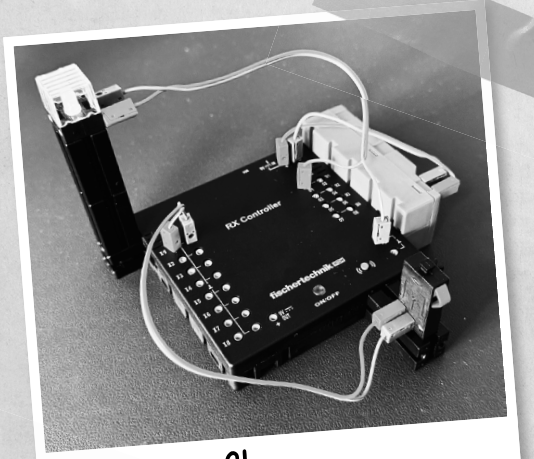


Ob man verhindern kann, dass jemand gewinnt, wenn derjenige dauerhaft den Taster drückt? Kann man das Spiel noch komplexer machen?

08 MATERIAL FÜR LERNENDE – DAS GEFUNDENE BUCH (S/W DRUCKVERSION)

Signallampe

Die folgende Konstruktion lässt beim Start die Lampe für 5 Sekunden leuchten.



Aufbau

Programm:

Einstellungen, die man Anweisungen mitgeben kann, nennt man Parameter

Einzelne Blöcke nennt man Anweisungen oder Befehle

AUFTRAG 1:

Baut eine Signallampe und bringt die Lampe zum Leuchten.

AUFTRAG 2:

Versucht nun das folgende Signal auf der Lampe anzeigen zu lassen: 3 Sekunden an - 5 Sekunden aus - 1 Sekunde an - 4 Sekunden aus - 7 Sekunden an.

AUFTRAG 3 (FÜR SCHNELLE):

Fügt der Konstruktion weitere LEDs hinzu. Lasst die LEDs erst abwechselnd und dann alle gemeinsam leuchten. Erfindet ein eigenes, möglichst komplexes, Leuchtschema!

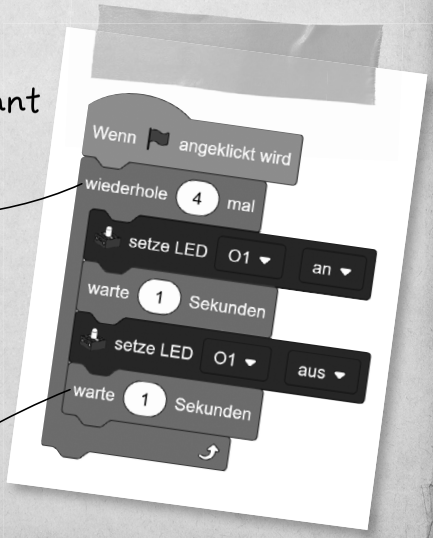
Wiederholung von Signalen

Folgendes Programm schaltet die Signallampe 4-mal hintereinander eine Sekunde an und eine Sekunde aus.

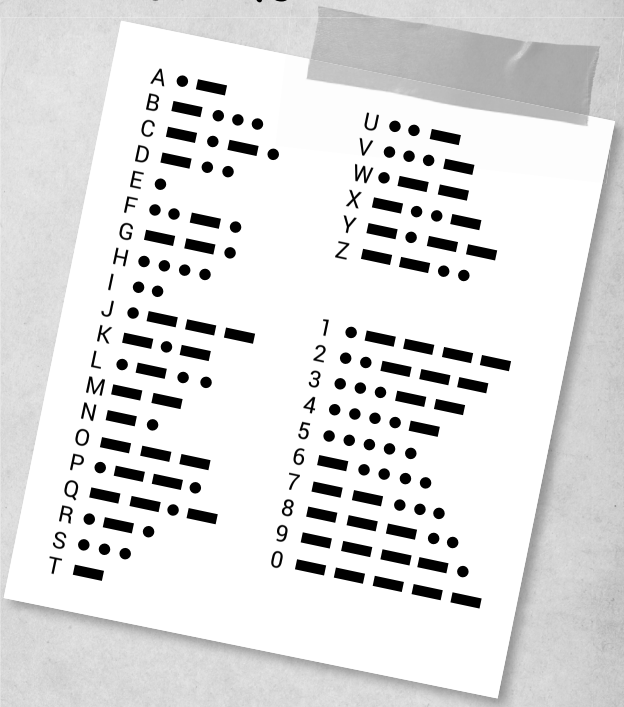
Programm:

Diese Struktur nennt man eine **Wiederholung** oder eine **Schleife**.

Ohne die Sekunde Pause würde die Wiederholung sofort starten und die LED direkt wieder leuchten.



Morsecode



Morsecode ist eine Art „Geheimsprache" aus kurzen und langen Signalen (oft dargestellt als Punkte und Striche).

AUFTRAG 1:

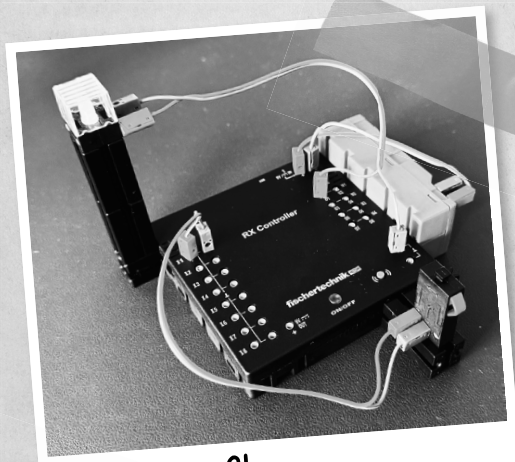
Programmiert ein SOS-Signal mit eurer Signallampe. Lasst das Signal dauerhaft abspielen.

AUFTRAG 2 (FÜR SCHNELLE):

Programmiert mit eurer Signallampe ein Wort oder einen kurzen Satz. Lasst anschließend jemanden aus einer anderen Gruppe erraten, was ihr gesendet habt.

Signallampe durch Taster steuern

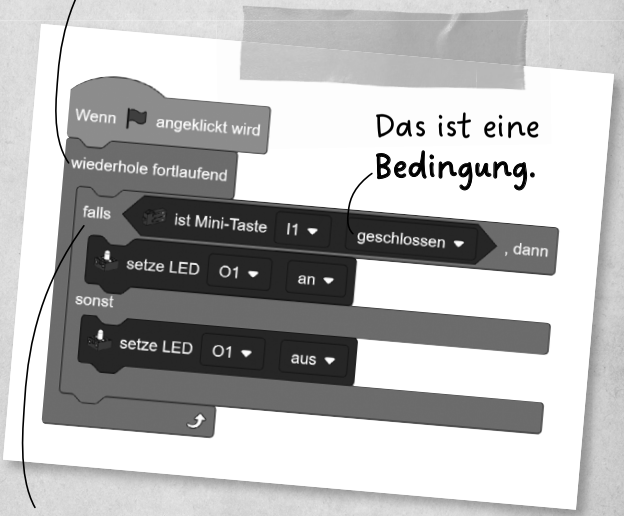
Die folgende Konstruktion lässt die Lampe immer dann leuchten, wenn der Taster gedrückt wird.



Aufbau

Programm:

Das Programm soll immer wieder abfragen, ob der Taster gedrückt ist. Daher benötigen wir eine **fortlaufende Wiederholung** um das ganze Programm.



Das ist eine **Bedingung.**

Solche Strukturen nennt man **Verzweigungen.** Abhängig von der Bedingung wird nur ein Zweig der Struktur ausgeführt.

AUFTRAG 1:

Baut einen Taster ein, bei dessen Betätigung 5-mal nacheinander das SOS-Signal abgespielt wird..

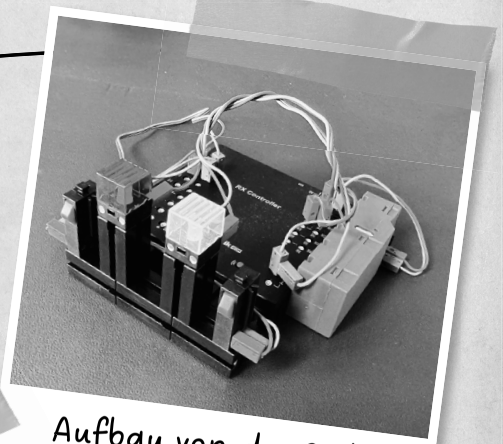
AUFTRAG 2 (FÜR SCHNELLE):

Baut einen zweiten Taster ein. Programmiert die Konstruktion so, dass ihr ja und nein morsen könnt und unterhaltet euch damit.

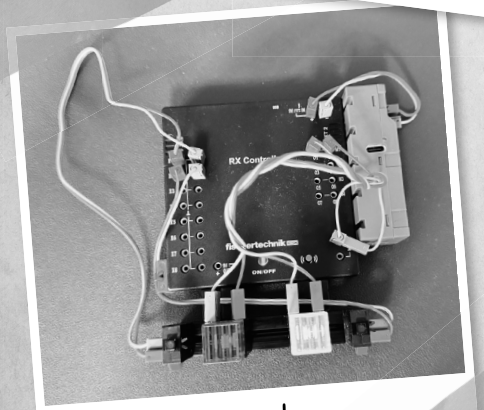
AUFTRAG 3 (FÜR GANZ SCHNELLE):

Baut und programmiert die Konstruktion so, dass jeder Taster ein anderes Wort ausgibt. Könnt ihr vielleicht durch einen Trick mit einem Taster auch zwei verschiedene Wörter auswählen?

Eigenes Spiel – Buzzer



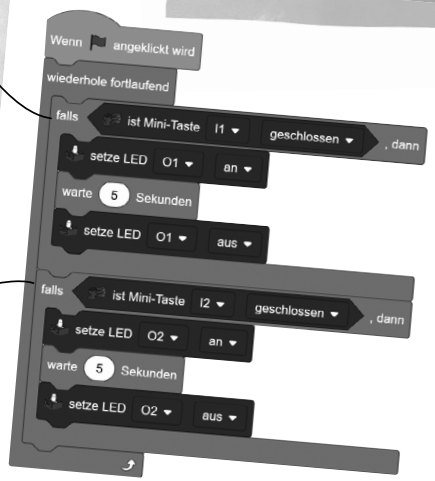
Aufbau von der Seite



Aufbau von oben

Programm:

Falls Taster 1 **gedrückt** ist, wird die erste LED für 5 Sekunden eingeschaltet. Dadurch kann nicht gleichzeitig die zweite LED aktiviert werden.

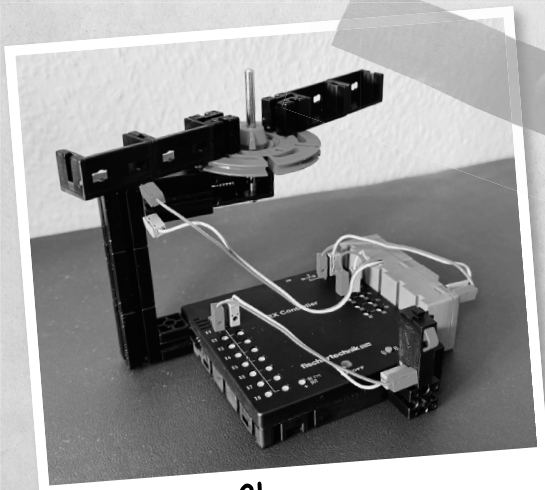


Danach wird dasselbe für Taster 2 geprüft.

Ob man wohl noch mehr Buzzer programmieren kann? Oder vielleicht sogar eine Art Pfeil, der zu dem Spieler zeigt, der schneller ist?

Programm, um die Geschwindigkeit eines Motors zu verändern

Die folgende Konstruktion besteht aus einem Motor mit Propeller und einem Taster. Durch Betätigung des Tasters wird die Geschwindigkeit des Motors erhöht.



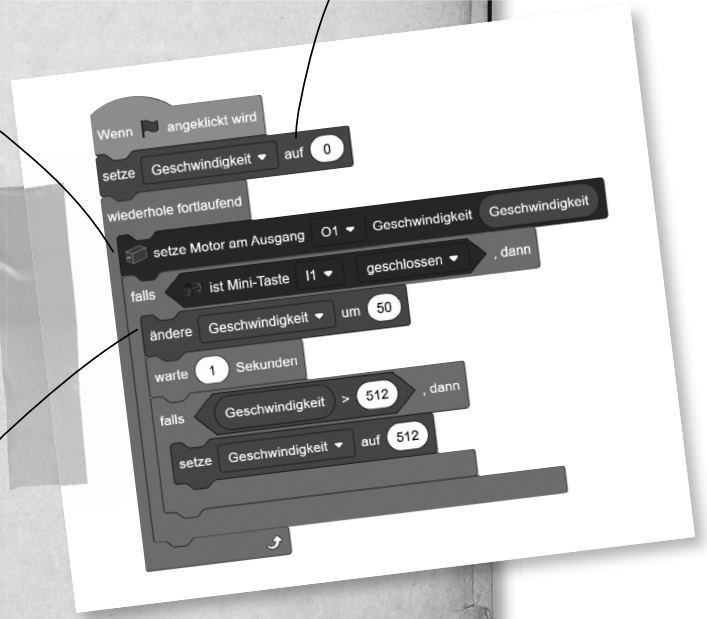
Aufbau

Programm:

Bei jedem Durchgang wird die Geschwindigkeit des Motors auf den Wert der Variablen gesetzt.

In Variablen können Werte gespeichert werden. Hier wird der Variablen Geschwindigkeit der Wert Null zugewiesen.

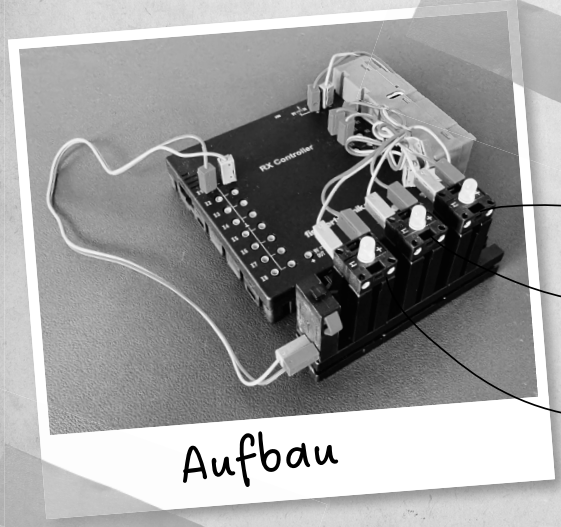
Wenn der Taster gedrückt ist, wird die Geschwindigkeit um 50 erhöht, aber nie über 512 (Maximalwert).



Ob man noch einen zweiten Taster einbauen kann, der die Geschwindigkeit verlangsamt? Oder den Motor in die andere Richtung drehen lässt?

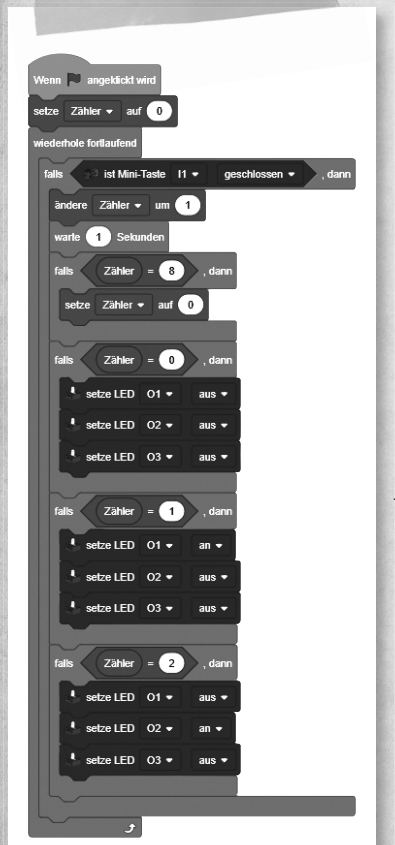
Binärzähler

Im Binärcode werden Zahlen nur durch 1 und 0 dargestellt. In der Tabelle sind einige Zahlen in der Binärschreibweise zu sehen. Die folgende Konstruktion stellt einen Zähler dar: Bei jeder Betätigung des Tasters zählt der Binärzähler um 1 hoch.



Aufbau

Programm:

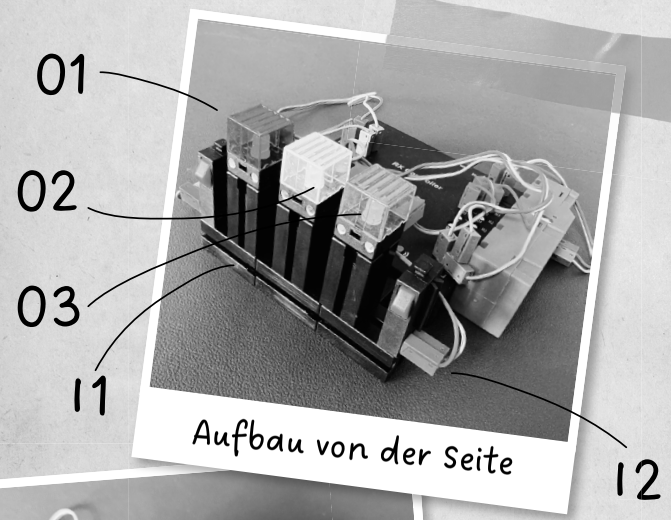


Das Programm zählt momentan nur bis 2. Wie kann man es wohl bis 7 zählen lassen? Und kann man einen zweiten Taster einbauen, der rückwärts zählt?

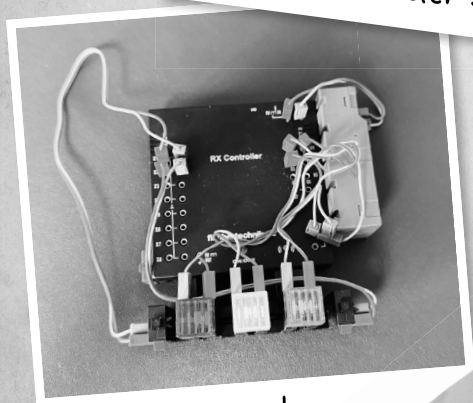
Dezimal-schreibweise	Binär-schreibweise
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Achtung: Programm noch unvollständig!

Reaktionsspiel



Aufbau von der Seite



Aufbau von oben

Folgendes Verhalten soll dem Spiel einprogrammiert werden:

- Nach einer zufälligen Wartezeit von bis zu 5 Sekunden leuchtet die mittlere LED auf.
- Wer nun als Erstes seinen Taster drückt, erhält einen Punkt – seine eigene LED leuchtet kurz auf.
- Sobald jemand 5 Punkte erreicht, gewinnt der Spieler oder die Spielerin. Das wird durch ein Blinken der mittleren LED und das dauerhafte Leuchten der LED des zugehörigen Spielers angezeigt.

Reaktionsspiel

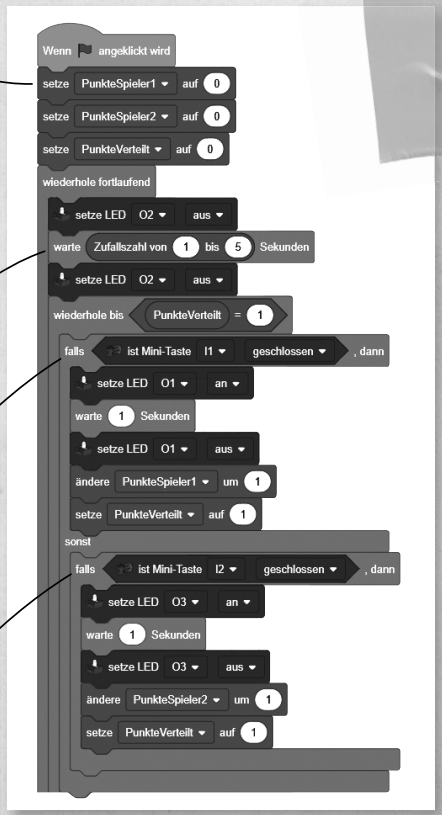
Variablen für Punkte der Spieler und Testvariable, ob Punkt verteilt wurde

LED 2 wird nach zufälliger Zeit eingeschaltet

Test ob Spieler 1 den Taster betätigt hat, falls ja, Signal an LED 1 und Verteilung der Punkte

Test ob Spieler 2 den Taster betätigt hat

Programm:

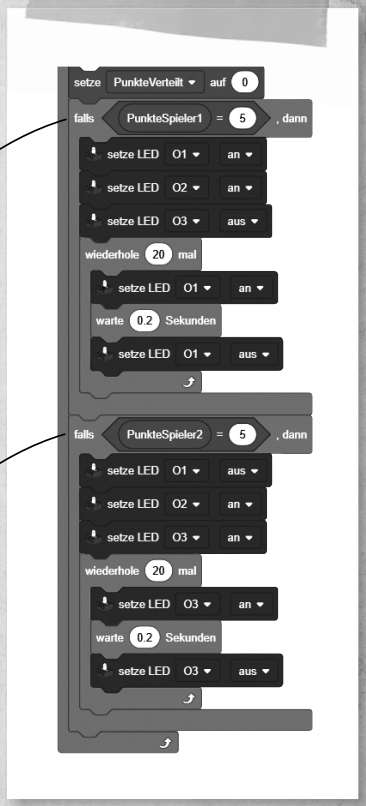


Oben rechts geht´s weiter...

Hier geht´s weiter...

Falls Spieler 1 genug Punkte hat, wird er durch die LEDs als Sieger angezeigt

Falls Spieler 2 genug Punkte hat, wird er als Sieger angezeigt



Ob man verhindern kann, dass jemand gewinnt, wenn derjenige dauerhaft den Taster drückt? Kann man das Spiel noch komplexer machen?