

Workshop

LightBag Bauen und Programmieren

Informatik- und Werken-Projekte



Teil-1: Check the Bug

Ziel

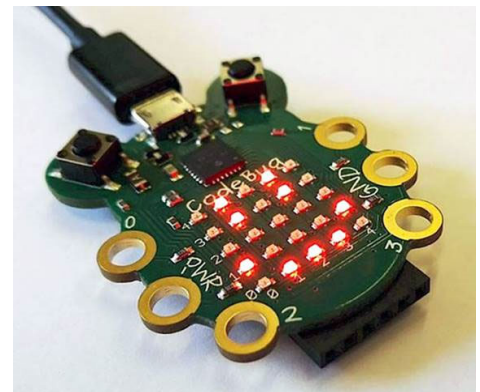
Wir wollen den Mini-Computer CodeBug kennenlernen, indem wir ein einfaches Pflanztopf-Bewässerungssystem aufbauen.

Ist die Erde trocken oder feucht? Falls trocken soll das System langsam Wasser giessen und immer wieder messen, um nicht zu übertreiben.

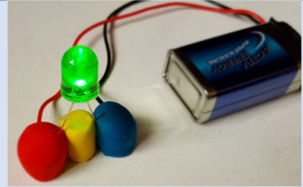
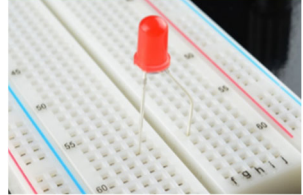
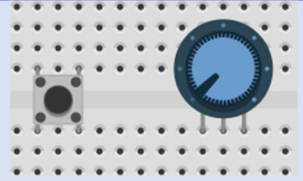
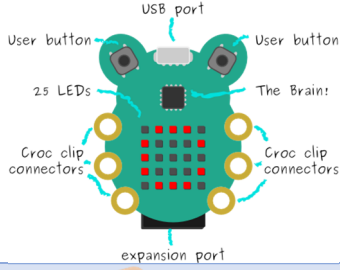
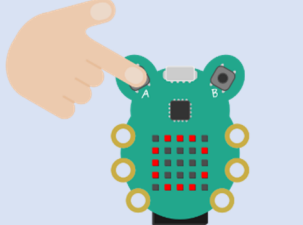
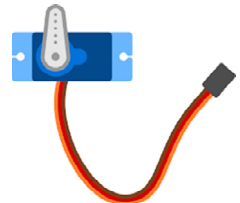
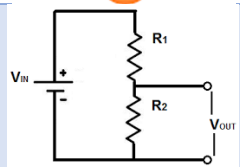
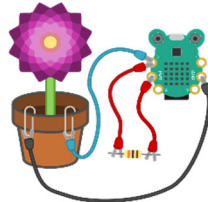
Kann ein Computer das machen? Oder besser ein Mini-Computer? Hat dieser Hände und Augen, oder wie können wir das System genau aufbauen?

Daran wollen wir in den nächsten Stunden zusammenarbeiten!

Wir beginnen mit Grundlagen über Stromkreise und wie wir sie steuern können. Dann stellen wir den Minicomputer als eine flexible Möglichkeit vor, einen Schaltkreis zu steuern. Und schließlich lernen wir die elektronischen Komponenten kennen, die wir für den Bau des Pflanzentopf-Giessers benötigen.

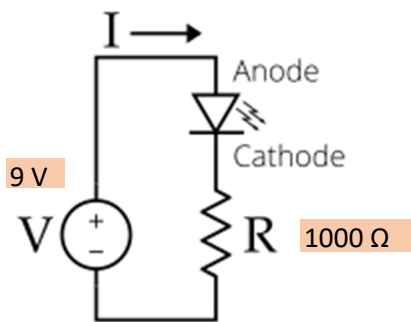


Inhalt

Stichwort	Aktivität	Eindruck
1. Grundlage Stromkreis	<ul style="list-style-type: none"> ○ Stromkreis mit Knetmasse (leitende und nichtleitende) ○ LED + Batterie-Schaltkreis bauen 	
2. Steckbrett	<ul style="list-style-type: none"> ○ Steckbrett oder Prototyp-Brett kennenlernen ○ LED mehrere Farben+ Widerstand 	
3. Steuerung in Schaltkreis	<ul style="list-style-type: none"> ○ Knopf im Schaltkreis einbauen ○ Weitere Ideen erkunden (z.B. mit Potentiometer) 	
Steuerung mit Mini-Computer 4. CodeBug kennenlernen	<ul style="list-style-type: none"> ○ CodeBug Platine kennenlernen ○ Was ist was: Programmierungsumgebung, Programm und Simulator? ○ Sich registrieren in Programmierung-Umgebung 	
5. CodeBug Programmieren Basisübungen	<ul style="list-style-type: none"> ○ LED-Matrix mit Symbolen und Texten animieren. ○ 1. Programm schreiben + Simulieren + Herunterladen ○ 2. Programm mit Bedingungen und Schleifen ○ Programm speichern und Aufgaben 	
6. Erweiterung Steuerung eines Servomotors	<ul style="list-style-type: none"> ○ CodeBug Erweiterung Port kennenlernen ○ Servomotor einbinden und programmieren ○ Zeiger des Servomotors bewegen, wenn eine Taste gedrückt wird. 	
7. Erweiterung Messung einer analogen Spannung	<ul style="list-style-type: none"> ○ Messung eines analogen Eingangssignals ○ Spannungsteiler mit zwei Widerständen 	
8. Abschluss Experiment	<ul style="list-style-type: none"> ○ Einfachen Feuchtigkeitsdetektor ausprobieren ○ Einfaches Pflanztopf-Bewässerungssystem aufbauen 	

1. Grundlage Stromkreis

Wie war das noch einmal im Physik-Unterricht, mit der Geschichte über den Stromkreis?



V : Spannung in Volt [V]

I : Strom in Ampere [A]

R : Widerstand in Ohm [Ω]

Grundregel: $V = R \cdot I$

Abbildung 1 Einfache LED-Schaltung *

a) Ordnen Sie in Abbildung 1 die folgenden Elemente zu:	<ul style="list-style-type: none"> Spannungsquelle Schutzwiderstand für die Strombegrenzung LED = Light Emitting Diode
b) Welches von diesen Elementen ist äquivalent zu einer Batterie?	<div>Batterieclip</div> <div>9V Batterie</div>
c) Wieso ist der folgende Aufbau mit Knetmasse [†] äquivalent zum Stromkreis von Abbildung 1? Wo ist der Schutzwiderstand?	<div>Grund-Schaltung</div> <div>Erweiterung als leuchtender Vogel</div>
d) Macht die Richtung des LEDs einen Unterschied, wenn man den Stromkreis aufbaut? Welches Bein muss man in Richtung «+» und welches in Richtung «-» einstecken?	<div>Anode +</div> <div>Kathode -</div> <div>Verweisquelle konnte nicht gefunden werden.</div> <div>Stromrichtung</div> <div>Anode (+)</div> <div>Kathode (-)</div> <div>Fehler!</div>

Tabelle 1 Stromkreis Grundlagen und einfache LED-Schaltungen

Experimentieren Sie und vergleichen Sie Ihre Antworten mit einer Nachbarin oder einem Nachbarn; notieren Sie neue Fragen, die entstanden sind.

* Bem.: Viele Figuren waren ursprünglich auf Englisch und verwenden die Notation V für die Spannung (anstelle von U auf Deutsch). Aus Gründen der Kompatibilität wird daher V für Spannung verwendet.

[†] Hinweis: folgende Websites haben detaillierte Anweisungen über die Vorbereitung und Experimente mit leitender Knetmasse: <https://tudu.org/projekt/elektro-knete> ; <https://www.kleine-ingenieure.de/leitende-knete/>

2. Steckbrett oder Prototyp-Brett

Mit älteren Schülerinnen und Schülern wollen Sie eventuell ein Prototyp-Brett verwenden. Schauen wir mal, wie das funktioniert.

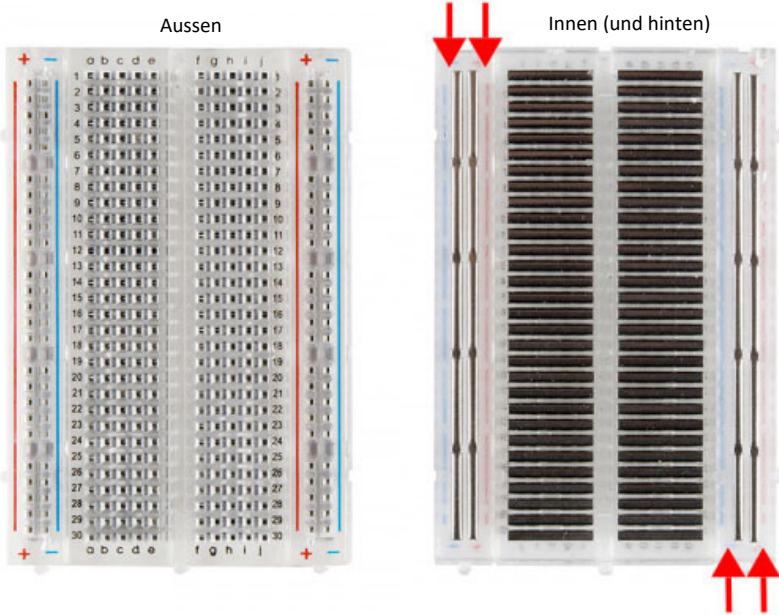
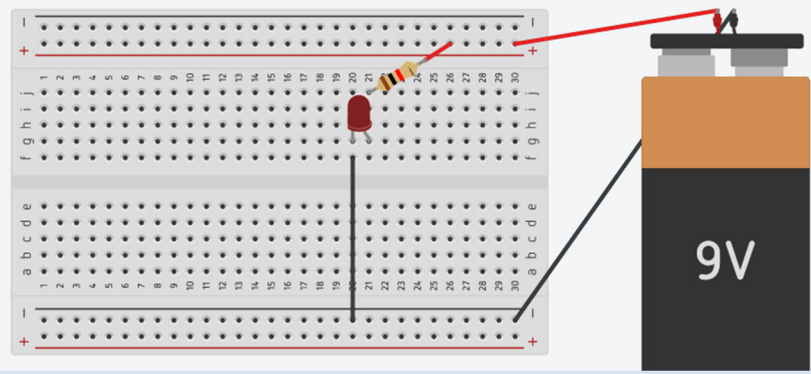
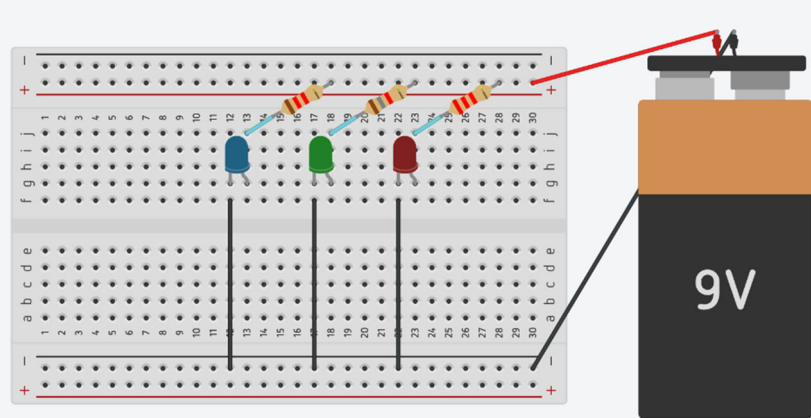
<p>Es gibt Versorgungs-schienen (in rot und blau markiert), und Querschienen.</p> <p>Auf den Bildern sehen Sie die äusseren und inneren Ansichten.</p> <p>a) Welche Löcher der Querschienen sind elektrisch miteinander verknüpft?</p>	
<p>b) Ist die Schaltung nebenan gleich wie die Schaltung in der Abbildung 1 (Seite 3)?[‡]</p>	
<p>c) Was ist in dieser Schaltung anders?</p> <p>d) Sind die LEDs parallel oder in Reihe geschaltet?</p>	

Tabelle 2 Steckbrett kennenlernen

Bitte probieren Sie aus, diskutieren Sie und stellen Sie Fragen!

[‡] Diese Schaltung kann simuliert werden unter:

<https://www.tinkercad.com/things/f0Z8sQtvjIP-einfacherledstromkreis>

3. Steuerung im Schaltkreis

Jetzt wollen wir den Schaltkreis steuern, zum Beispiel sollte das LED nur dann leuchten, wenn man eine Taste drückt.

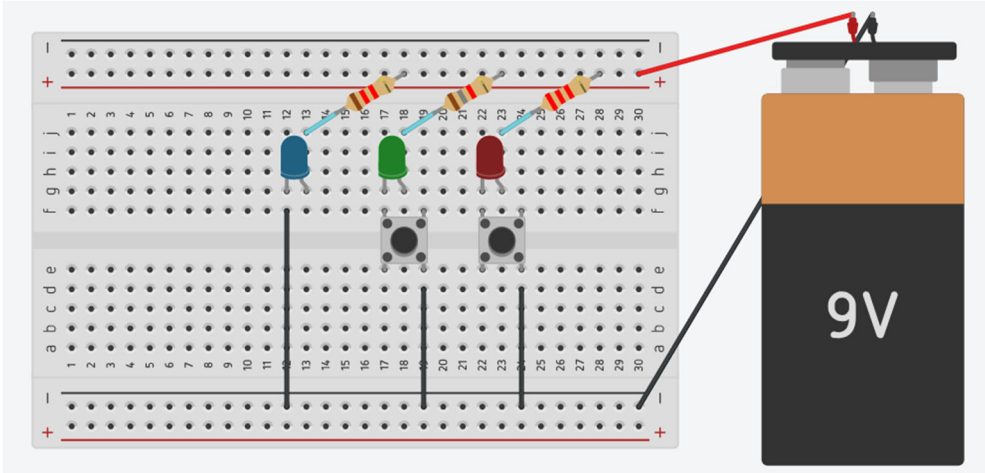
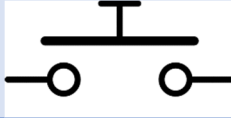
<p>So könnte man die Schaltung mit Drucktasten auf einem Steckbrett aufbauen. §</p>	
<p>Aufgabe</p>	<p>Zeichnen Sie den Schaltplan dieser Schaltung.</p>
<p>Hinweis 1</p>	<p>Der Schaltplan ist eine abstrahierte Zeichnung der Schaltung, wie in Abbildung 1 (Seite 3).</p>
<p>Hinweis 2</p>	<p>Das Symbol für eine Drucktaste in einem Schaltplan sieht so aus:</p> 
<p>Frage</p>	<p>Das Symbol der Drucktaste hat 2 Beine (links und rechts), aber das Bauteil hat 4 Beine (siehe Hinweis 2 oben).</p> <p>Was denken Sie? Welche Beine sind immer miteinander verbunden, und welche Beine werden erst beim Betätigen der Taste verbunden?</p>

Tabelle 3 Einfacher LED-Stromkreis mit Knopf

Bitte probieren Sie aus, diskutieren Sie und stellen Sie Fragen!

§ Diese Schaltung können Sie simulieren unter:

<https://www.tinkercad.com/things/iMu24pv5vIO-einfacherledstromkreismitknopf>

Steuerung mit Mini-Computer

4. CodeBug Platine kennenlernen

Noch besser wäre: eine Steuerungsmöglichkeit zu haben, die man nach Bedarf anpassen kann. Schauen wir mal, ob der CodeBug Mini Computer für diese Aufgabe geeignet ist. Zuerst entdecken wir die verschiedenen Teile der Platine des Mini-Computers CodeBug.

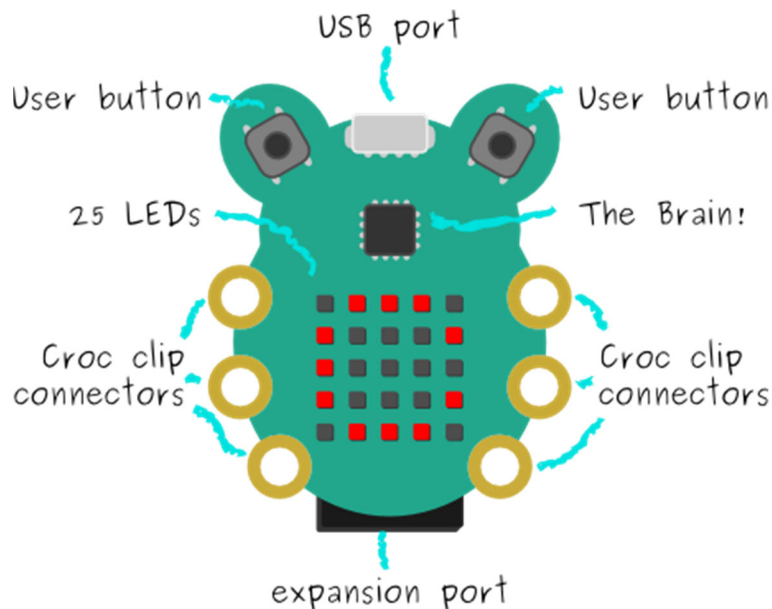
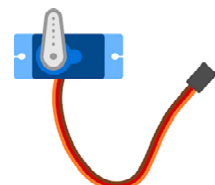


Abbildung 2 CodeBug-Platine

Welches Teil ist das?	Versuchen Sie für jedes Teil in der CodeBug-Figur (Abbildung 2) eine Tätigkeit zuzuordnen.
USB Port	Hier kann man ein USB-Kabel verbinden, zur Speisung und um Programme herunterzuladen.
	Hier kann man drücken. So kann beispielsweise die Benutzerin oder der Benutzer mit dem Programm interagieren.
	Mikrocontroller, so zu sagen das Gehirn des Mini-Computers. Hier wird das Programm (eine Reihenfolge von Befehlen) gespeichert und durchgeführt.
	Anzeige in der Form einer Matrix mit 5x5 LEDs.
	Hier kann man anfassen oder Kabel mit einer Krokodilklemme fixieren.
	Hier kann man eine Erweiterung anschliessen. Zum Beispiel einen Servomotor.



4.1. Account für CodeBug Programmierung-Umgebung

Falls Sie noch nicht mit dem CodeBug gearbeitet haben, wollen wir jetzt einen Account kreieren, damit Sie Ihre eigenen Programme entwickeln und speichern können.


Registrierung ausfüllen

Schritte	Inhalt
Browser starten	Die Benutzeroberfläche zum Programmieren des CodeBugs befindet sich auf einer Website. Um diese zu erreichen müssen wir einen Browser starten. Mögliche Browser sind: Firefox, Google Chrome, Microsoft Edge, Safari Achtung: Verwende nicht Internet Explorer!
CodeBug Website öffnen	Gehen Sie auf die Website: www.codebug.org.uk
Registrierung öffnen	Klicken Sie auf den Knopf «Register» ganz rechts oben auf der Website
Hinweis	Für die Registrierung benötigen Sie eine E-Mail-Adresse, deren Posteingang Sie abrufen können (wegen der Aktivierungsmail).


Füllen Sie das Formular auf der Website wie folgt aus:


- (1) Benutzername eingeben (frei wählen)
- (2) E-Mail Adresse
- (3) Passwort eingeben (frei wählen)
- (4) Passwort nochmals eingeben
- (5) Aktivieren
- (6) Deaktivieren
- (7) Account erstellen


WICHTIG: Notieren Sie sich Ihre Login Daten auf der Rückseite dieses Blattes.




Register new user

1
 

2
 

3
 

4
 

☒ 5 I agree to the [Privacy Policy](#) and the [Terms & Conditions](#).

☐ 6 I would like to receive emails with news and updates about CodeBug
Don't worry, we won't spam you or sell your details.

7 **CREATE AN ACCOUNT**

Ihre Anmeldedaten

Benutzername:

E-Mail:

Passwort:

Link in Bestätigungsmail anklicken

CodeBug schickt Ihnen zum Bestätigen Ihrer Registrierung eine Mail. Öffnen Sie ihren Posteingang, finden Sie die Bestätigungsemail (eventuell am Spam), und klicken Sie den Link an.

Jetzt ist Ihre Registrierung bestätigt und Sie sind bereits eingeloggt.

Bereit zum Loslegen!

4.2. Rund um der Programmierungsumgebung

Um den Mini-Computer zu programmieren, müssen wir ein Programm schreiben und dieses auf die Platine herunterladen. Die Programme schreiben wir in einem Web-Browser (z.B. Firefox) auf der Website www.codebug.org.uk. Abbildung 3 zeigt einen Schnappschuss der **Programmierungsumgebung**, auf der man landet, wenn man auf «Create» klickt.

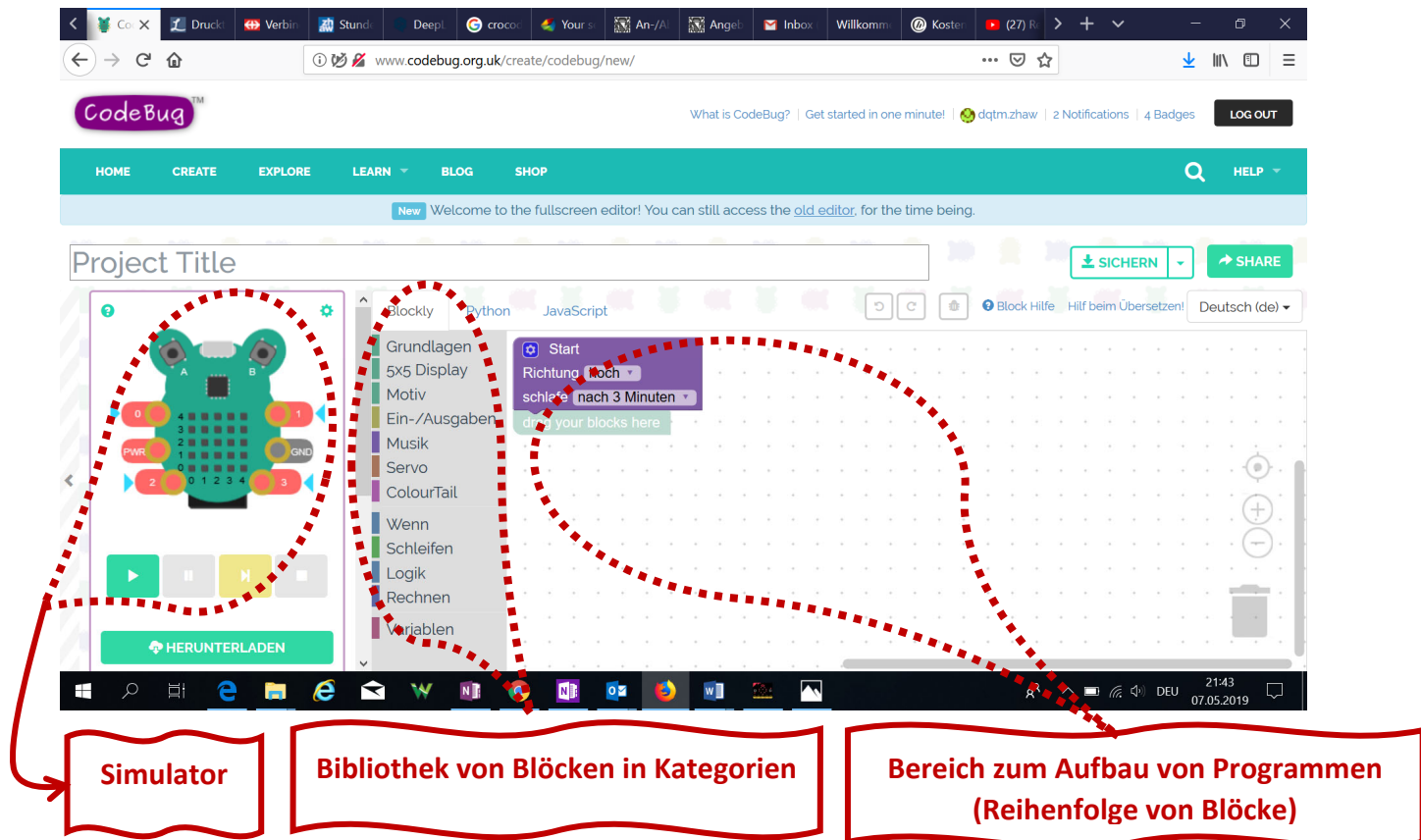


Abbildung 3 CodeBug Programmierungsumgebung

Auf der linken Seite von Abbildung 3 CodeBug Programmierungsumgebung finden Sie den Simulator, mit dem Sie das Programm simulieren und mögliche Fehler oder Verbesserungen erkennen können, bevor Sie es auf die Platine herunterladen.

In der Mitte haben Sie die Bibliothek, unterteilt in verschiedene Kategorien. Dort finden Sie alle Blöcke, die Sie zum Aufbau von Programmen benötigen.

Auf der rechten Seite befindet sich der Bereich zum Aufbau des Programms. Hier können Sie die ausgewählten Blöcke aus der Bibliothek ziehen, loslassen und sie in der gewünschten Reihenfolge miteinander verbinden.

Jetzt ist es wieder Zeit zum Ausprobieren. Wir fangen mit ein paar Basisübungen.

5. CodeBug Programmieren: Basisübungen

5.1. Erstes Programm erstellen & herunterladen

Erstes Programm

Sie werden nun ihr erstes Programm erstellen. Rechts auf dem Bild sehen Sie wie das Programm aussehen sollte. Jedes Programm beginnt beim violetten Start-Block. Alle weiteren Blöcke finden Sie in der Bibliothek.

Der erste Block, den Sie suchen sollen, heisst «Warte bis Taste ändert». Klicken Sie auf **Grundlagen** um den Block zu finden. Ziehen Sie ihn danach mit der Maus unter den Start-Block.

Die nächsten zwei Blöcke finden Sie im Menu unter **Motiv**.



Simulieren Sie Ihr Programm

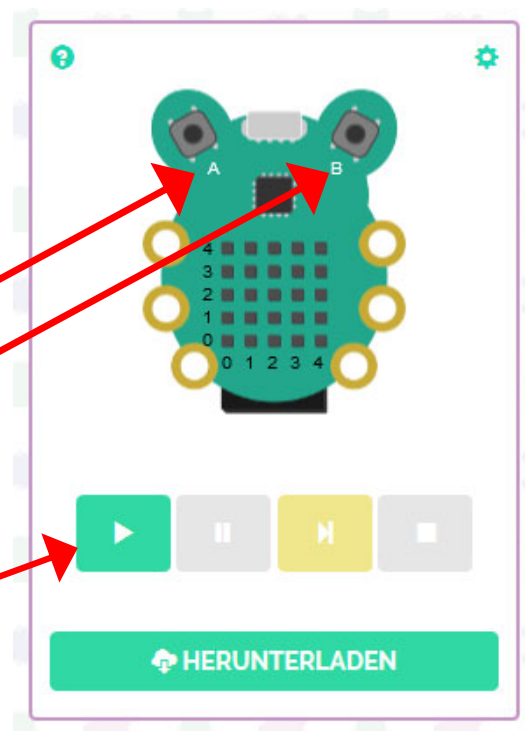
Bevor man ein Programm auf den CodeBug lädt, kann man dieses auf der Website simulieren. Drücken Sie dafür auf den Play-Knopf.

Danach drücken Sie auf Taste A oder B und sehen Sie was passiert.

Taste A

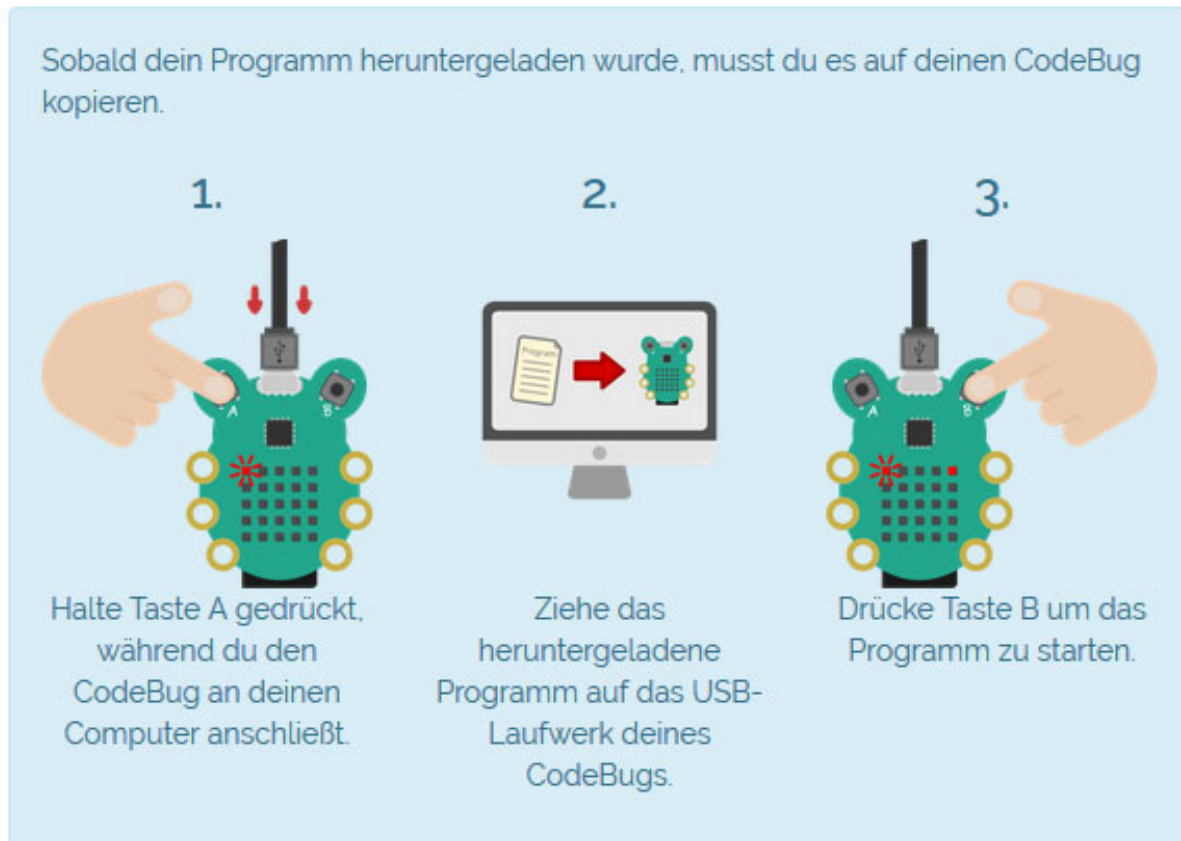
Taste B

Play-Knopf



Laden Sie Ihr Programm auf den CodeBug

Klicken Sie auf «Herunterladen». Es erscheint die folgende Anleitung. Lesen Sie diese durch und laden Sie dann wie beschrieben das Programm auf den CodeBug.



Testen Sie Ihr Programm jetzt auf dem CodeBug-Board und klären Sie eventuelle Schwierigkeiten oder Fragen mit den WorkshopleiterInnen.

Als nächsten Schritt wollen wir ein Programm schreiben, bei dem das in der LED-Matrix angezeigte Motiv je nach gedrückter Taste variiert.

Folgen Sie bitte den Anweisungen im Abschnitt 5.2 Zweites Programm: Zwei Motive.

5.2. Zweites Programm: Zwei Motive

Wir wollen ein Programm erstellen, mit welchem wir zwei verschiedene Motive auf den LED's leuchten lassen können. Wir werden die Tasten verwenden, um zwischen den zwei Motiven zu wählen.

Verwendete Blöcke	Im Menu	Beschreibung
warte bis Taste ändert	Grundlagen	Das Programm wartet bei diesem Block bis man Taste A oder B drückt.
wenn	Wenn	Wenn die Bedingung im Puzzle Stück wahr ist, wird der Code bei «mache» ausgeführt. Ansonsten wird es übersprungen.
Taste A/B gedrückt	Ein-/Ausgaben	Ist wahr, wenn die Taste A/B gedrückt ist.
zeichne Motiv	Motiv	Zeichnet das Motiv, welches im Puzzle Stück gegeben wird.
male ein Motiv	Motiv	Erstellt ein Motiv bei dem die LED's leuchten, bei welchen ein Häkchen gesetzt wurde.
Wiederhole solange wahr	Schleifen	Ist eine Dauerschleife. Alles innerhalb dieser Dauerschleife wiederholt sich für immer.

Erstellen Sie das folgende Programm



Simulieren das Programm

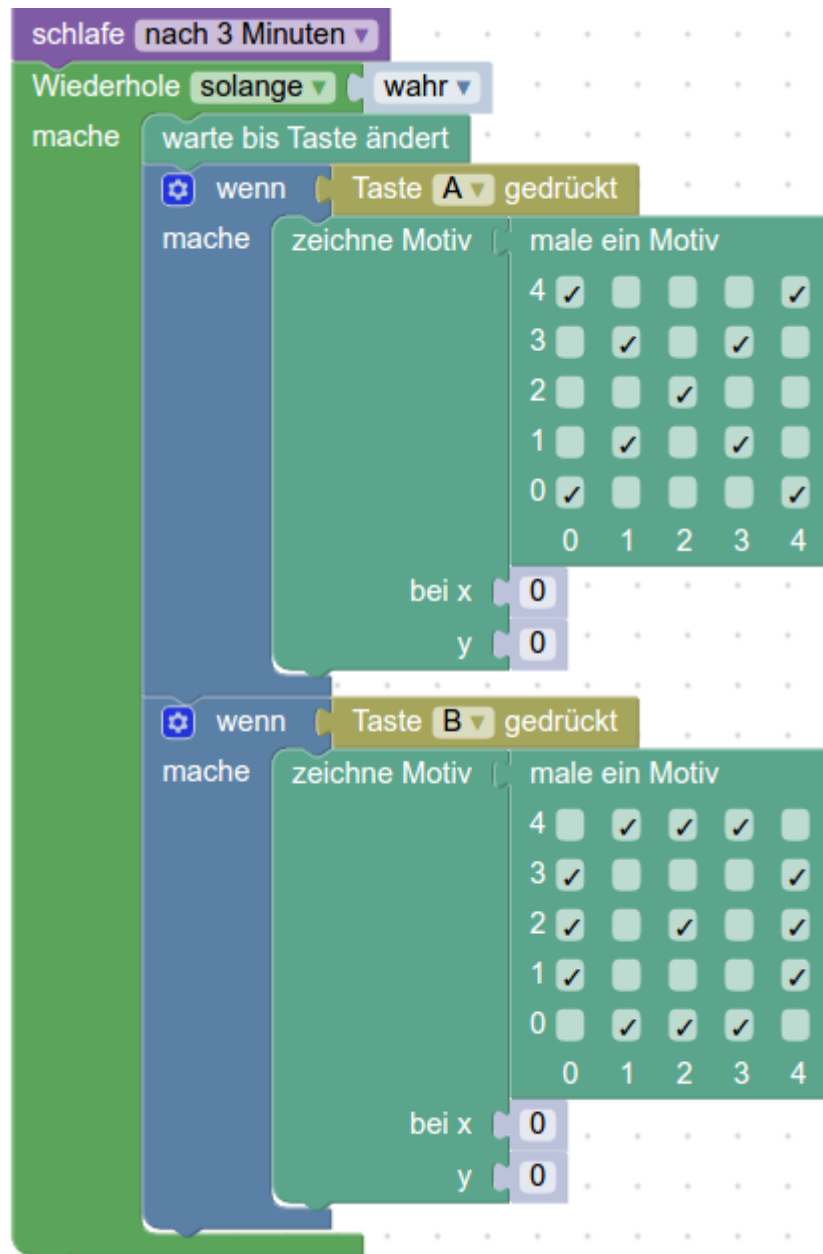
Drücken Sie beim Simulator auf den Play Knopf und danach auf Taste A. Drücken Sie nochmals den Play Knopf und danach die Taste B.

Warum müssen Sie immer wieder den Play Knopf drücken, um zwischen den Motiven zu wechseln?

Programm verbessern

Das Programm endet sobald man eine Taste (A oder B) gedrückt hat. Wir wollen jedoch, dass das Programm immer weiterläuft und wir immer wieder zwischen den Motiven wechseln können mit den Tasten.

Dafür müssen wir unser Programm in eine Dauerschleife packen.



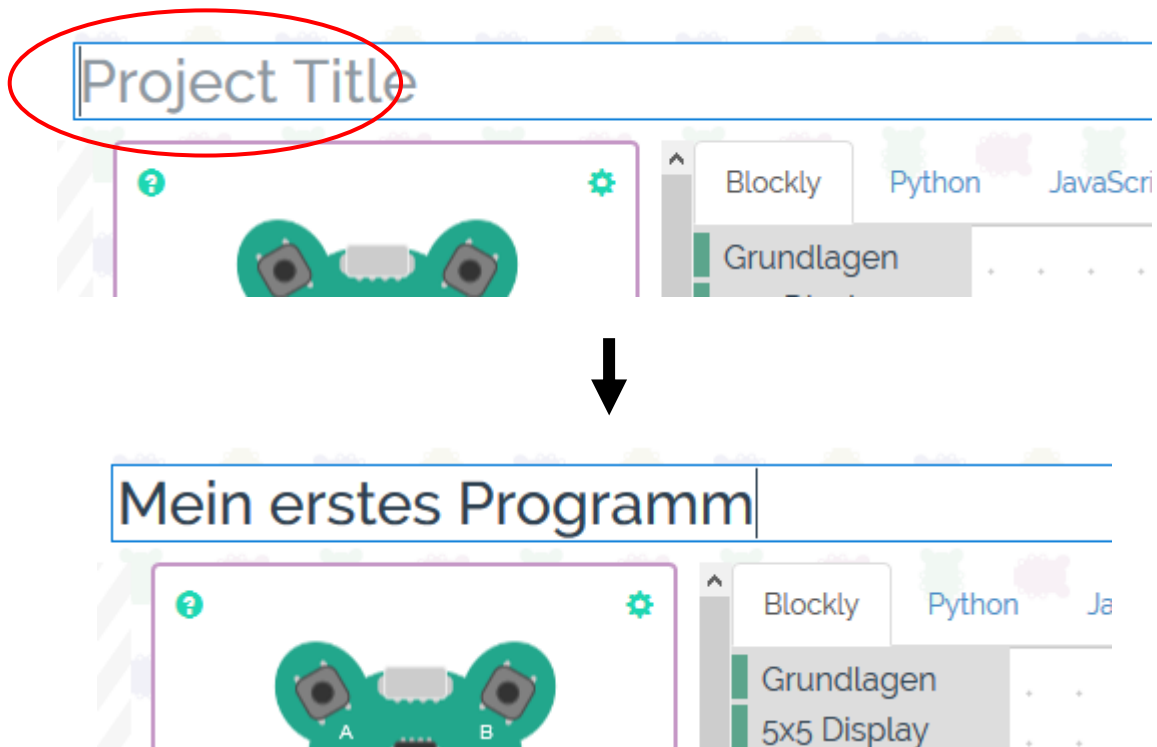
Simulieren und Herunterladen

Testen Sie das neue Programm auf dem Simulator. Wenn es funktioniert, laden Sie es auf den CodeBug.

5.3. Projekt speichern und neues Projekt beginnen

Damit ihr Programm nicht verloren geht, wenn Sie die Website verlassen, muss das Projekt abgespeichert werden.

- (1) Klicken Sie in die Titelleiste des Projektes und geben den Titel ein: **Mein erstes Programm**



- (2) Drücken Sie auf den Knopf «**SICHERN**»



Um ein neues Projekt zu beginnen, klicken Sie auf «**Create**» und danach auf «**+ NEW CODEBUG PROJECT**»

Sie können auch aus Ihrem bisherigen Projekt ein neues machen. Klicken Sie dafür auf das kleine Dreieck bei «**Sichern**» und wählen «**Save as**». Dann können Sie einen neuen Projekttitel eingeben.

5.4. Aufgaben zur CodeBug Basisübung

Aufgabe 1 - Smiley

Beim CodeBug soll ein lächelnder Smiley aufleuchten, wenn Sie Taste A drücken. Wenn Sie Taste B drücken soll ein trauriges Gesicht erscheinen.

Benötigte Blöcke	Im Menu	Beschreibung
warte bis Taste ändert	Grundlagen	Das Programm wartet bei diesem Block bis Sie Taste A oder B drücken.
wenn	Wenn	Wenn die Bedingung im Puzzle Stück wahr ist, wird der Code bei «mache» ausgeführt. Ansonsten wird es übersprungen.
Taste A/B gedrückt	Ein-/Ausgaben	Ist wahr, wenn die Taste A/B gedrückt ist.
zeichne Motiv	Motiv	Zeichnet das Motiv, welches im Puzzle Stück gegeben wird.
male ein Motiv	Motiv	Erstellt ein Motiv bei dem die LED's leuchten, bei welchen ein Häkchen gesetzt wurde.
Wiederhole solange wahr	Schleifen	Ist eine Dauerschleife. Alles innerhalb dieser Dauerschleife wiederholt sich für immer.

Aufgabe 2 - Name

Beim CodeBug soll wieder ein lächelnder Smiley aufleuchten, wenn Sie Taste A drücken. Aber wenn Sie Taste B drücken, soll **Ihr Name** auf den LED's erscheinen. Suchen Sie dafür im Menu **Motiv** nach Blöcken, die Ihnen möglicherweise bei dieser Aufgabe behilflich sein könnten.

TIPPS:



- (1) Block «bewege Motiv»: Nimmt ein Motiv und lässt es über die LED Anzeige laufen. Besonders hilfreich, wenn nicht das ganze Motiv innerhalb der LED-Anzeige Platz hat.
- (2) Block «Erzeuge Textmotiv»: Macht aus einem Text ein Motiv für die LED-Anzeige.

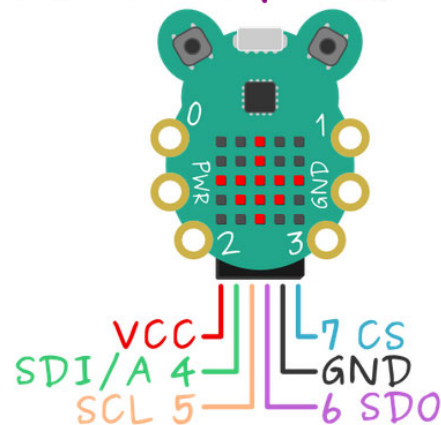
Versuchen Sie, das Programm zu bauen, zu simulieren, herunterzuladen und auf dem Board zu testen. Anschließend können Sie Ihr Programm mit der Lösung (im Anhang) überprüfen.

6. Erweiterung: Steuerung eines Servomotors

Die untere Seite der CodeBug Platine hat eine Buchsenleiste, an der man weitere Hardware-Komponenten anbinden kann. Wir wollen einen Servomotor anschliessen und diesen mit einem Programm steuern.

Abbildung 4 zeigt die Details über die Ein/Ausgänge des Erweiterungs-Ports, und eine mögliche Pin-Belegung. Bitte achten Sie darauf, dass diese Ein/Ausgänge keine internen Schutzwiderstände haben.

Extension Inputs/Outputs



(x)

Abbildung 4 CodeBug Erweiterungs-Port und Pin-Belegung Beispiel

Um die Erweiterungs-Ports zu benutzen, müssen Sie im Start-Block durch drücken des blauen Zahnrads, die Option „Erweiterungspins Eingang/Ausgang“ anklicken. Setzen Sie danach die Beinchen 4 bis 7 als Digitalausgang. Wenn Sie nochmals auf das blaue Zahnrad drücken, schliesst sich das Menü wieder.

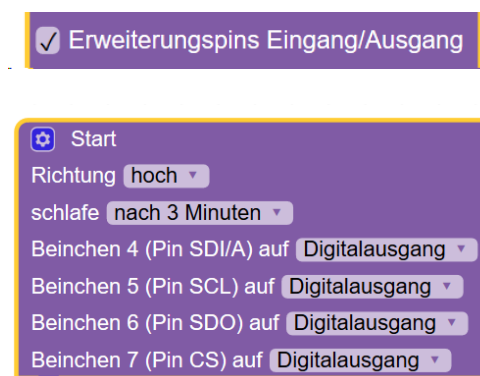


Abbildung 5 Start-Block-Konfiguration für die Benutzung des Erweiterungs-Ports

So weit so gut, aber jetzt müssen wir ein Programm für die Steuerung des Servomotors schreiben und ausprobieren. Analysieren Sie den Code, der in Abbildung 6 gezeigt wird.

Welche Blöcke kennen Sie bereits und welche sind neu? Stellen Sie Fragen an die Workshop-LeiterInnen, und diskutieren Sie mit anderen Teilnehmenden, um das ganze Programm zu verstehen.

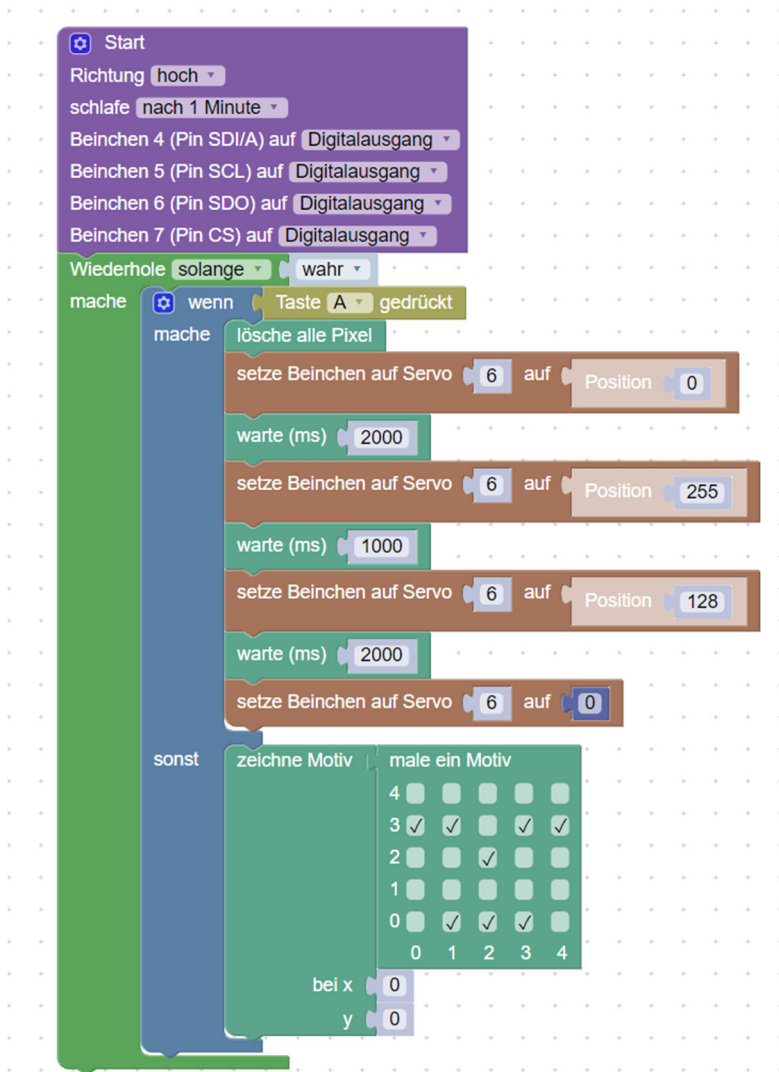


Abbildung 6

Beispiel Programm für Steuerung eines Servomotors verbunden mit

Ground / VCC / SDO
Erde / Speisung / Data

Verbinden Sie jetzt den Servomotor, wie in Abbildung 7, und probieren Sie das Programm oben durchzuführen. Sie können das Programm unter dem folgenden [Link](https://tinyurl.com/y5ggnh9j)* finden, oder indem Sie in der Suchleiste der CodeBug-Webseite den Projektnamen „test_servomotor“ eingeben.

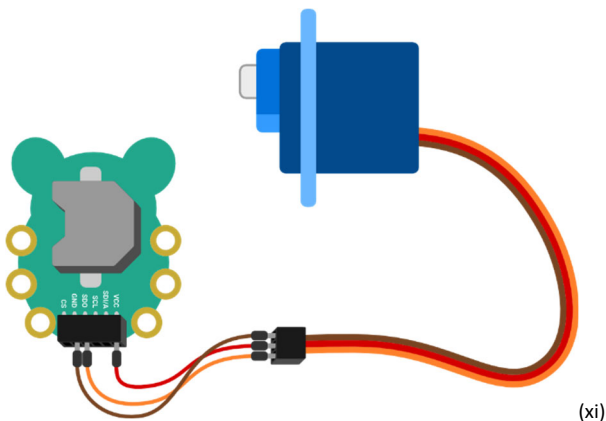
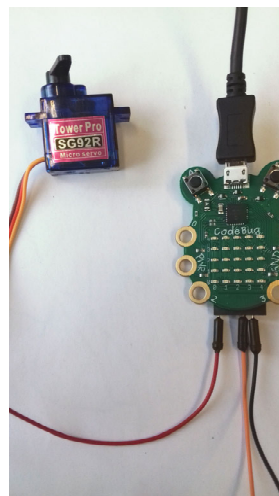


Abbildung 7

Servomotor Anschluss mit dem Erweiterungs-Port

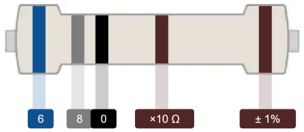
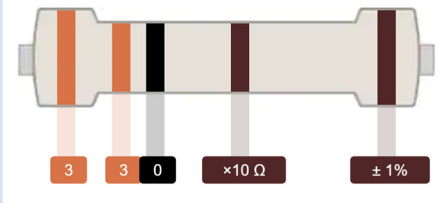
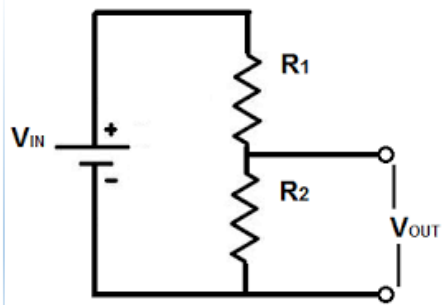
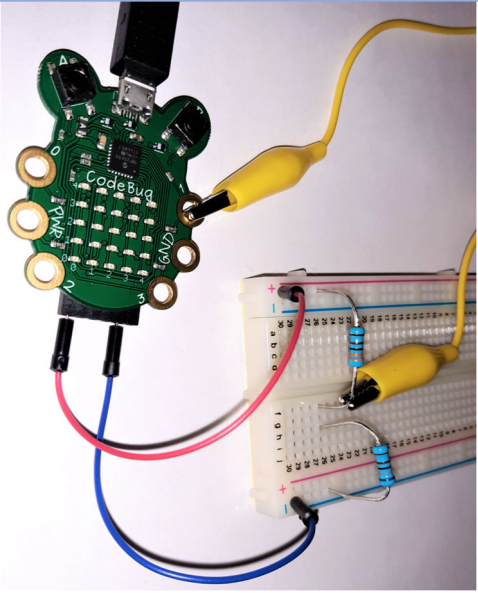


* <https://tinyurl.com/y5ggnh9j>

7. Erweiterung: Messung einer analogen Spannung

Spannungsteiler mit Widerständen und Potentiometer

Um die Feuchtigkeit der Erde zu überprüfen, werden wir den Widerstand der Erde mit einem bekannten Widerstand vergleichen. Wie können wir einen solchen Vergleich mit dem CodeBug-Minicomputer realisieren? Am besten baut man einen sogenannten Spannungsteiler und misst die Spannung in der Mitte des Teilers (als Analogwert). Wir wollen diese Idee zunächst mit zwei bekannten Widerständen ausprobieren.

<p>Wert eines Widerstandes herauslesen</p> <p>Verwenden Sie einen Web-Rechner, um das Farbcodesystem zu bewerten[†]. Wir haben im Material Widerstände mit 5 Ringe. Halten Sie den Widerstand mit dem Toleranz-Ring nach rechts und geben Sie z. B. die Farbe der 5 Ringe ein:</p>	<div> <div> Number of Bands <input type="radio"/> 4 Band <input checked="" type="radio"/> 5 Band <input type="radio"/> 6 Band </div> <div> Resistor Parameters 1st Band of Color: Blue 6 2nd Band of Color: Grey 8 3rd Band of Color: Black 0 Multiplier: Brown $\times 10 \Omega$ Tolerance: Brown $\pm 1\%$ </div> <div> Output  </div> </div> <p>Resistor value: 6.8k Ohms 1%</p>
<p>Welche Wert hat den folgende Widerstand?</p> <p>R = _____</p>	
<p>Spannungsteiler mit Festwiderständen</p> <p>Bauen Sie einen Spannungsteiler mit zwei Widerständen. Nehmen Sie $R_1=3.3k\Omega$ und $R_2=6.8k\Omega$. Als Versorgung (V_{in}) nehmen wir VCC (für die Plus-Seite) und GND (für die Minus-Seite) vom CodeBug. Den mittleren Punkt V_{out} verbinden Sie über ein Krokodilkabel mit dem Beinchen/Pin-1.</p>	<div> <div> $R_1 = 3.3k\Omega$ $R_2 = 6.8k\Omega$ $V_{in} = 5V$ $V_{out} = ?$ </div> <div>  </div> <div>  </div> </div>
<p>Schätzen Sie den Wert von V_{out}</p>	<p>Berechnen Sie den Spannungswert von V_{out} anhand der Werte von R_1, R_2 und V_{in}</p> <p>$V_{out} =$ _____</p>

[†] Der Toleranzring ist ein wenig weiter entfernt als die anderen. Ein Beispiel für einen Web-Rechner finden Sie unten:
<https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code>, oder in
<https://tinyurl.com/whryjxss>

Die Spannung V_{out} wird im CodeBug als analoger Eingang gemessen und in einen digitalen Wert mit 8 Bits umgewandelt. Dies bedeutet einen Wert zwischen 0 und 255, der proportional zum Bereich 0V bis 5V ist. Welchen Wert erwarten Sie z.B. bei einem Eingang von ca. 3.37V zu lesen?

Analoge Spannung	Analog zu Digital Umwandlung	Digitaler Wert mit 8 Bits Auflösung
5V	entspricht	255
0V		0

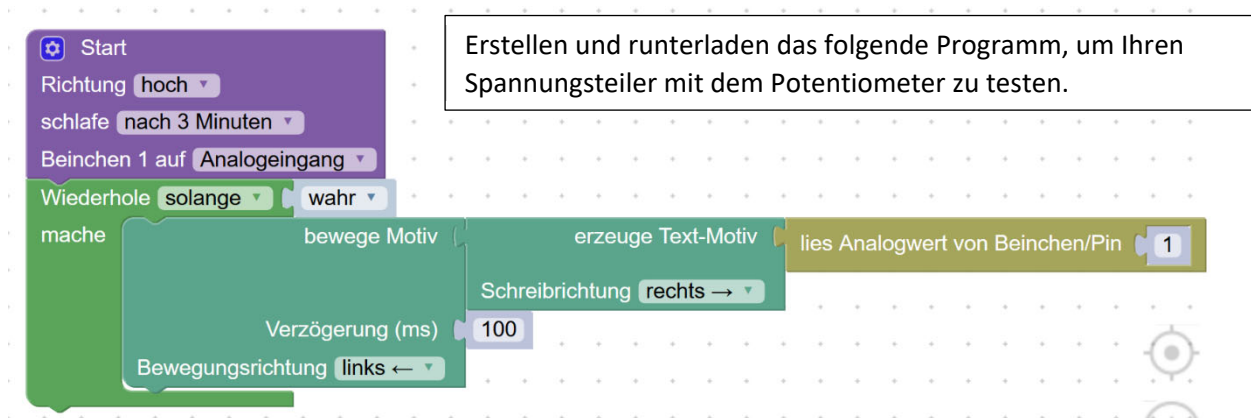


Abbildung 8 Test-Programm: Analog-Wert einlesen und anzeigen ‡

Als nächstes ersetzen wir R2 durch ein Potentiometer. Der mittlere Punkt des Potentiometers kann als variabler Widerstand verwendet werden. Weitere Details zum Potentiometer finden Sie im Anhang.

Sie haben ein Potentiometer, dessen Widerstand zwischen 0 und 10kOhm variiert. Welchen Bereich erwarten Sie als Analogwert auszulesen? Testen Sie es mit dem CodeBug und dem Programm aus Abbildung 8.

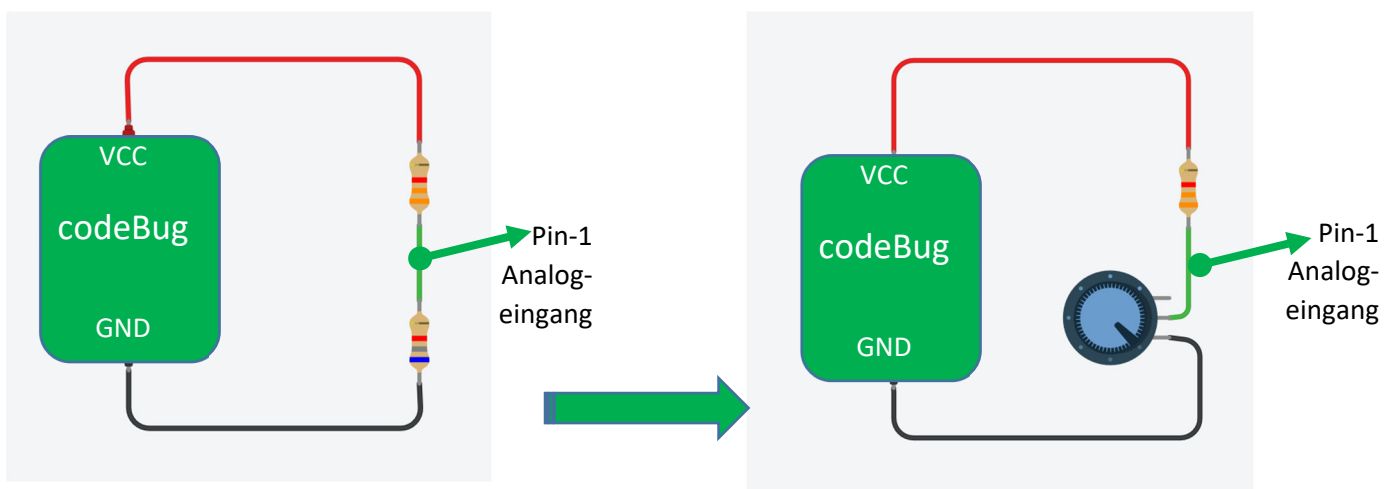


Abbildung 9 Spannungsteiler mit zwei Widerständen, oder mit einem Widerstand und einem Potentiometer

‡ Dieses Programm können Sie auch unter <https://tinyurl.com/2maudy6r> finden.

Hinweis: bei einem Spannungsteiler mit zwei Widerständen ist das Verhältnis der Widerstände proportional zu den entsprechenden Spannungsabfällen. Verwenden Sie Widerstände größer als 1kOhm, um die Stromaufnahme des Spannungsteilers unter 5mA zu begrenzen (bei $V_{cc}=5V$).

Hinweis: Sie können auch einen Spannungsteiler nur mit dem Potentiometer aufbauen. Dazu verwenden Sie dann die 3 Pins des Potentiometers. In diesem Fall würde der Analogwert den gesamten Bereich von 0 bis 5V oder von 0 bis 255 variieren. Können Sie das erklären?

Das Potentiometer im Workshop Material hat einen Widerstand von 10kOhm von ganz links nach ganz rechts. Das mittlere Bein, nimmt einen Punkt irgendwo in der Mitte dieses Widerstandes auf. Die Position dieses Abgreifpunktes kann mit der Welle des Potentiometers variiert werden. Auf diese Weise haben wir in diesem mittleren Abgreifpunkt eine Spannung zwischen 0V und 5V. Im Anhang III finden Sie eine detaillierte Beschreibung des Potentiometers.

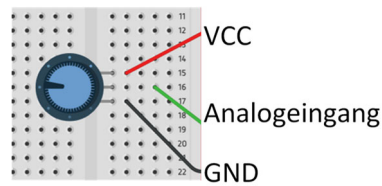


Abbildung 10 Spannungsteiler mit Potentiometer

Der Aufbau von Abbildung 9 auf der rechten Seite mit einem Widerstand und einem Potentiometer ist für uns hier praktisch, weil wir das Potentiometer später durch den Pflanztopf ersetzen werden, um die Feuchtigkeit des Bodens zu messen.

8. Abschlussexperiment: Feuchtigkeit messen & Bewässerung aktivieren

Dieses Experiment basiert auf dem Projekt [micro:bit plant water pump](https://www.youtube.com/watch?v=jANCdtKJAKY)[§] (Abbildung 11). Im Video können Sie den Aufbau mit Benutzung eines micro:bit Mini-Computers anschauen. Wir können dasselbe mit einem CodeBug implementieren.

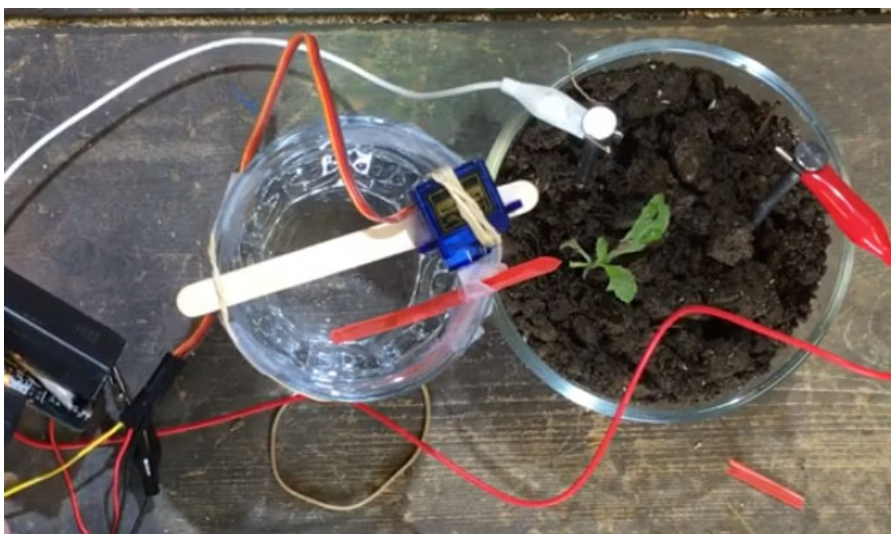


Abbildung 11 Einfaches Pflanztopf Bewässerungssystem mit Servomotor und Mini-Computer

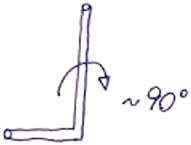
[§] <https://www.youtube.com/watch?v=jANCdtKJAKY>

Die Grundidee des Bewässerungssystems ist:

- Der Mini-Computer misst regelmässig die Feuchtigkeit der Erde
- Falls diese zu niedrig ist, soll der Servomotor aktiviert werden, um Wasser zu giessen
- Der Arm des Servomotors ist mit einem Strohhalm verbunden
- Servomotor und Strohhalm sind über einem Gefäss mit Wasser montiert

Wir haben bereits mehrere Elemente kennengelernt, die man für den Aufbau dieses Experiments braucht. Was uns noch fehlt ist: die folgende drei Schritte zu probieren und zusammen zu bringen.

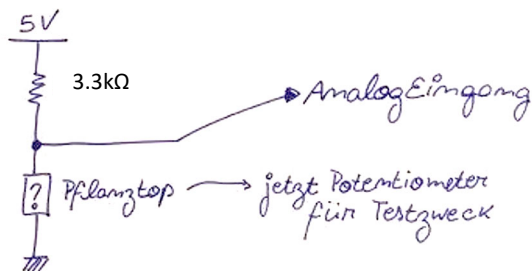
Schritt-1 : Servo
 ↳ Position
 ↳ Winkel



$0^{\circ}-180^{\circ} \Leftrightarrow 0-255$

Binden Sie den Strohhalm an den Arm des Servomotors und experimentieren Sie damit, wie groß der Winkel sein muss, um Wasser einzufüllen und auszugießen.

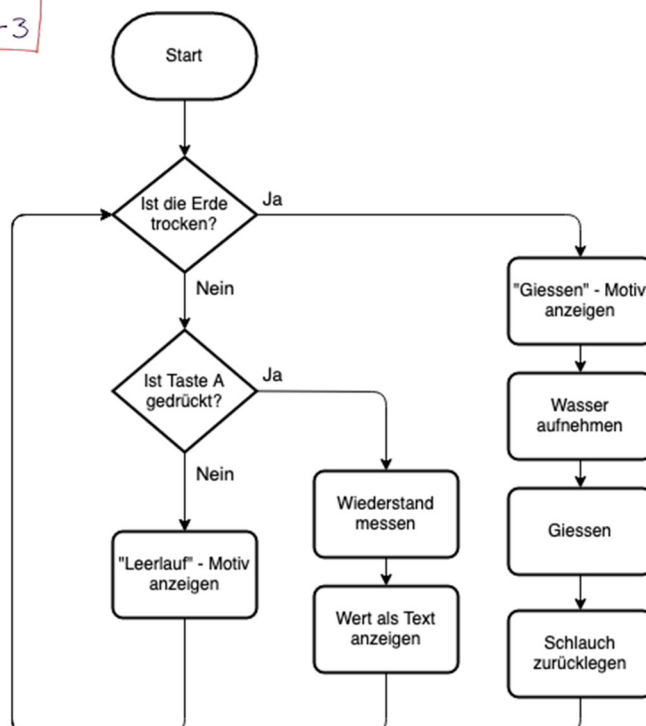
Schritt-2 : Feuchtigkeit/Widerstand messen



Ersetzen Sie das Potentiometer (in der Schaltung von Abbildung 9) durch einen Pflanzentopf und versuchen Sie, den Widerstand der Erde zu messen.

Zu Testzwecken können Sie auch zwei Stücke Baumwolle nehmen, ein trockenes und ein nasses (mit Salzwasser).

Schritt-3



Schreiben Sie ein Programm, das wiederholt einen Analogwert einliest. Und je nach eingelesenem Wert entscheidet, ob der Boden bewässert werden soll oder nicht.

Ein möglicher Ablauf ist hier in Form eines Flussdiagramms aufgezeichnet.

Dokumentieren Sie Ihren Versuch (Foto oder Video vom Aufbau und Link zum Code) und senden Sie uns bitte per E-Mail.

Wir werden Ihre verschiedenen Lösungsvarianten sammeln, an alle verteilen und offene Fragen gerne beantworten.

Eine Lösungsvariante von uns finden Sie im Anhang.

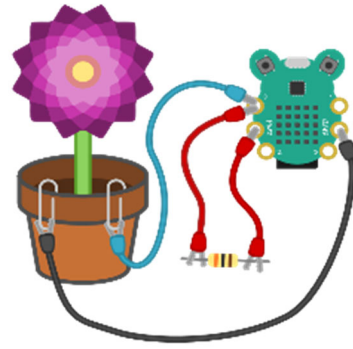
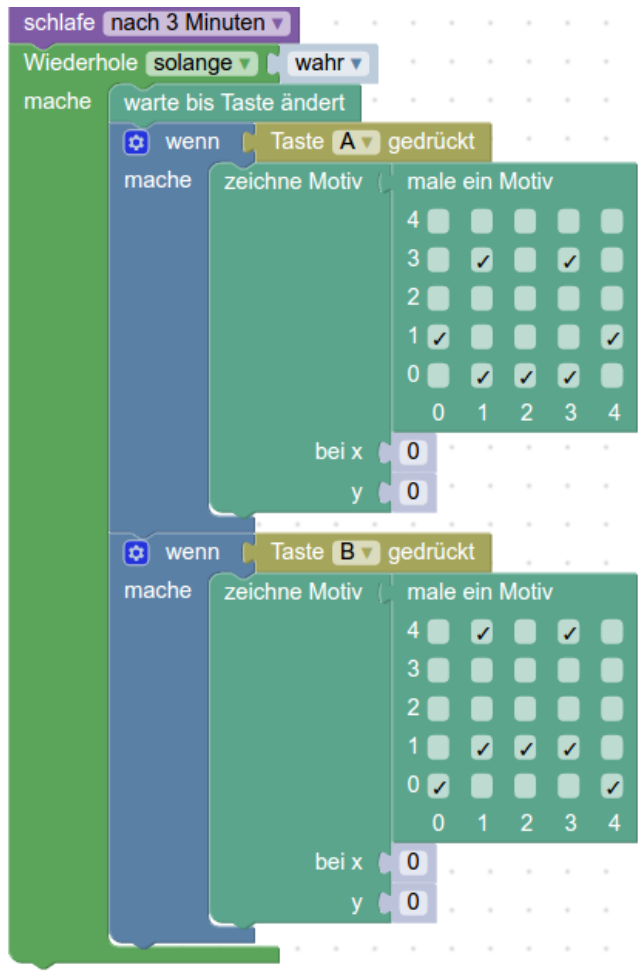


Abbildung 12 Feuchtigkeit messen im Vergleich mit bekanntem Widerstand

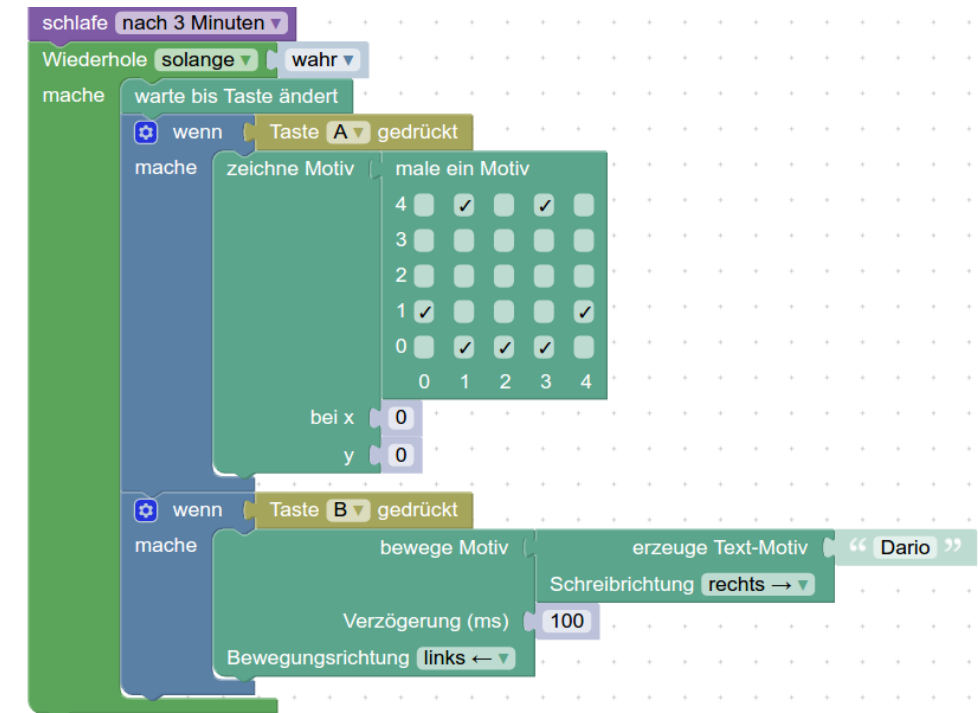
ANHANG

I. Lösungen zum Aufgaben Basisübungen

Lösung von Aufgabe 1 – Smiley

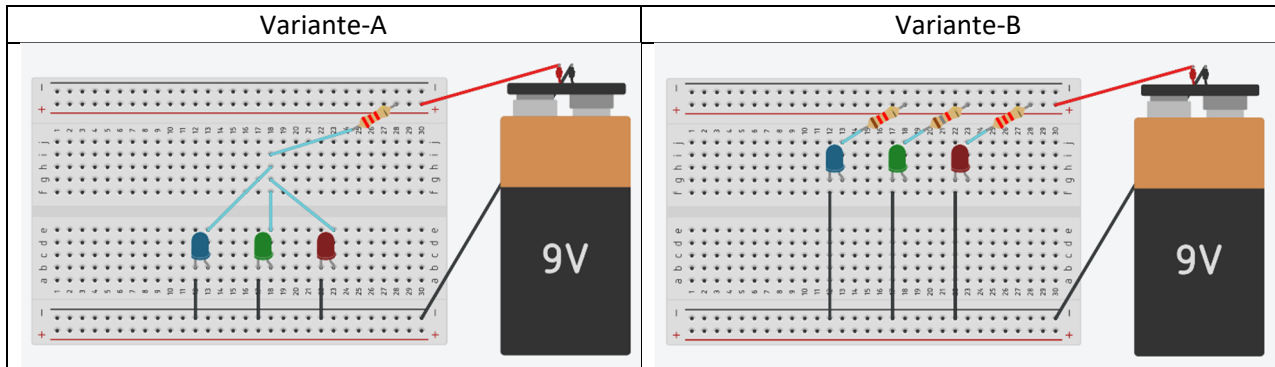


Lösung von Aufgabe 2 - Name



II. Herausforderung Aufgabe zum Abschnitt: Prototyp-Brett

Was ist der Unterschied, wenn Sie die Schaltung mit den 3 LEDs, wie die beiden Varianten unten aufbauen? Welche Farbe leuchtet am stärksten bei Variante-A? Warum?



Hintergrundinformationen

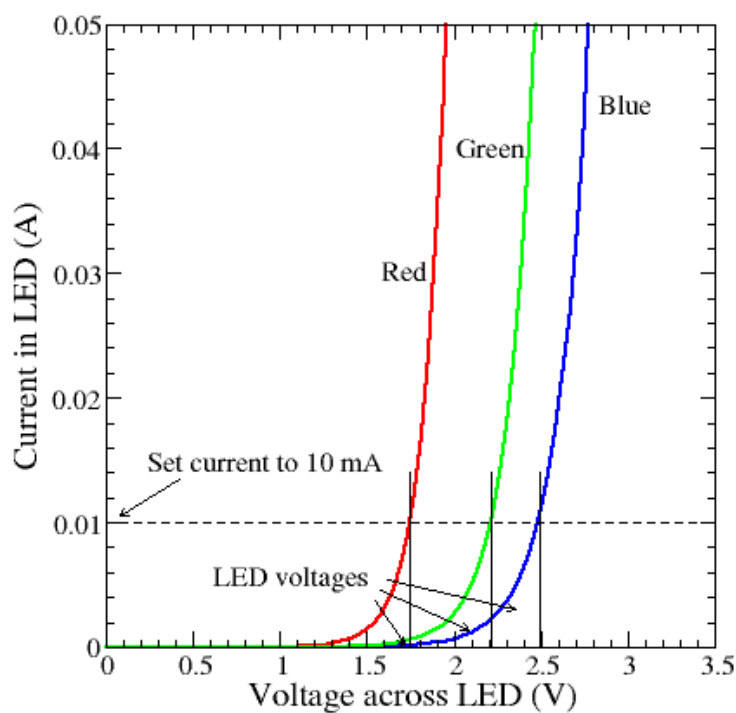


Abbildung 13 Beispiel von LED-Kennlinie Strom gegen Spannung (xii)

Bem.: Die Schwellenspannung kann für verschiedene LED-Typen variieren. Bei einer vergleichbaren LED-Fertigung bleibt jedoch die Tendenz, dass Blau eine höhere Schwellenspannung als Grün und Rot aufweist.

Wenn Sie sich für weitere Details oder Aufgaben mit Benutzung des Steckbretts interessieren (zum Beispiel mit dem Potentiometer), fragen Sie bitte die Workshop Leiterin. Wir werden uns freuen Ihnen weitere Unterlagen/Ideen zukommen zu lassen.

III. Komponente: Potentiometer

Potentiometer


What it Does:

Produces a variable resistance dependant on the angular position of the shaft.

Identifying:

They can be packaged in many different form factors, look for a dial to identify.

No. of Leads:

3

Things to watch out for:

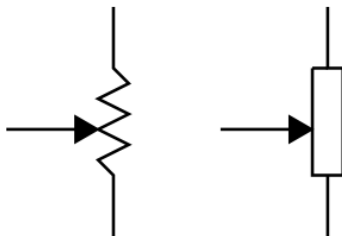
- Accidentally buying logarithmic scale.

More Details:

<http://ardx.org/POTE>

Abbildung 14 Potentiometer-Komponentenübersicht

In einem Schaltplan wird ein Potentiometer durch eines der beiden folgenden Symbole dargestellt:



Das Potentiometer kann als variabler Widerstand oder als Spannungsteiler arbeiten.

Abbildung 15 Potentiometer-Schaltplansymbol



Abbildung 16 Potentiometer-Aufbau

Wenn Sie 2 der 3 Beine verbinden, erhalten Sie einen variablen Widerstand:

Tipp: Nehmen Sie entweder die 2 oberen oder die 2 unteren Beine.



Abbildung 17 Potentiometer-Anwendung als variabler Widerstand

Wenn Sie alle 3 Beine verbinden, funktioniert es als Spannungsteiler:

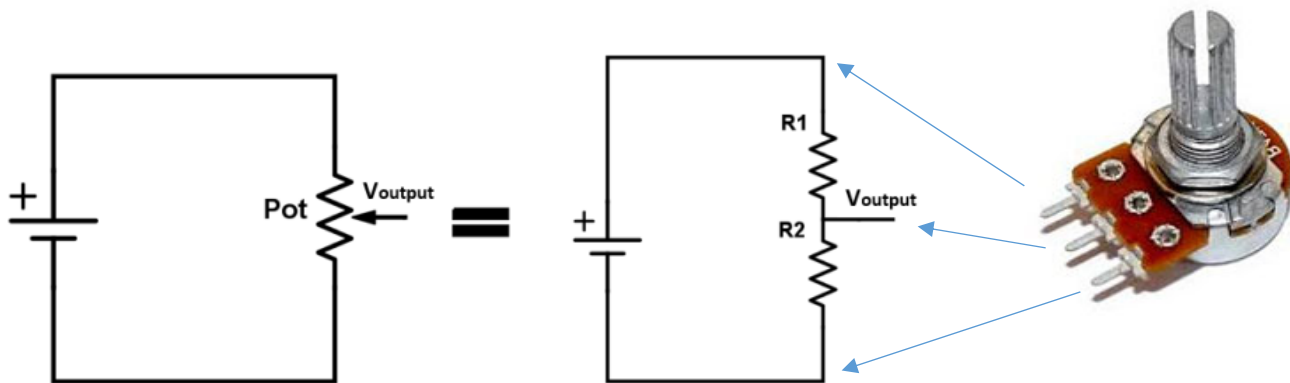


Abbildung 18 Potentiometer-Anwendung als Spannungsteiler

IV. Komponente: Widerstand (für Durchsteckmontage)

Der Widerstandswert wird oft auf dem Bauteil mit 4, 5 oder 6 farbigen Streifen angezeigt. Die Farbecodierung für Widerstände mit 4 und 5 Ringe ist unten gezeichnet.

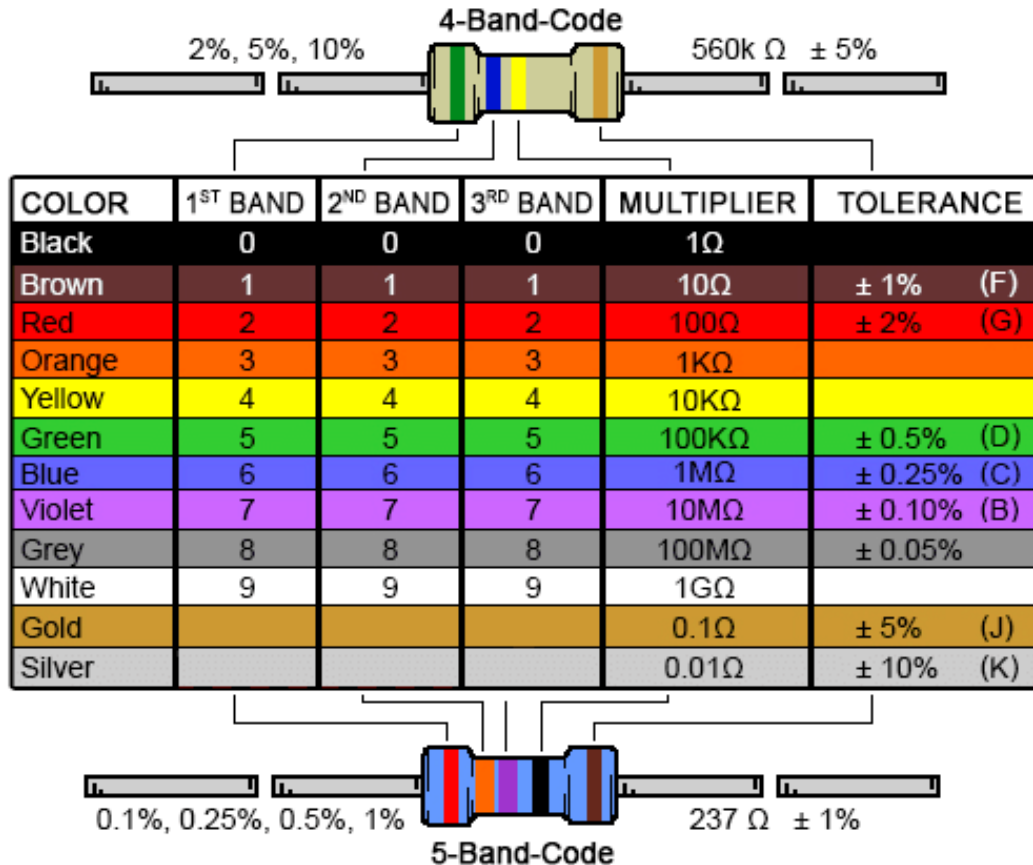
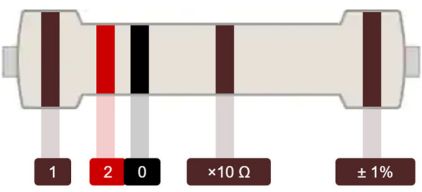
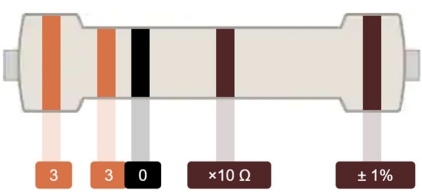
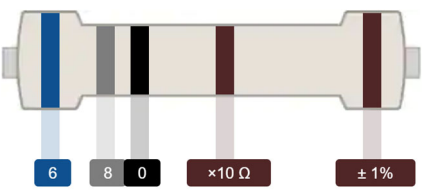


Abbildung 19 Farbecodierung für Widerstände mit 4 und 5 Ringe


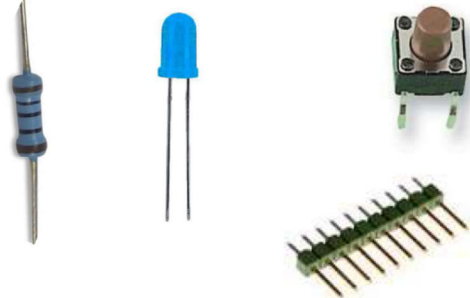
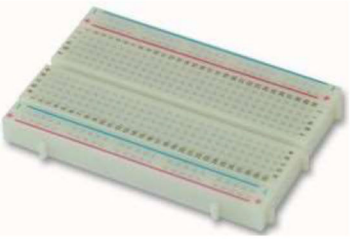
In unserem Workshop-Material haben wir Widerstände mit 5 Ringe, mit den folgenden Werte:


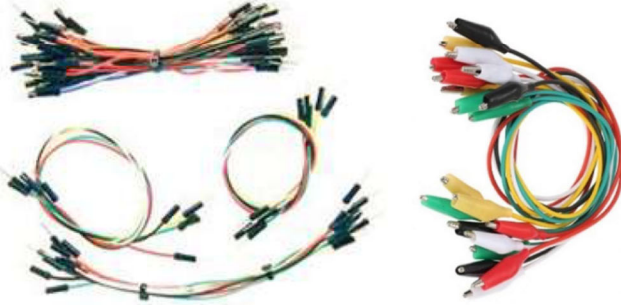

 <p>Resistor value: 1.2k Ohms 1%</p>	 <p>Resistor value: 3.3k Ohms 1%</p>	 <p>Resistor value: 6.8k Ohms 1%</p>
--	--	--

V. Material-Liste (BOM : Bill of Material)

Name	Menge	Technische Beschreibung	Preis CHF	Beispiel Lieferant CH/EU
Codebug	1	Programmierbares Entwicklungsboard	13.60	Farnell
Steckplatine	1	Lötfreie Steckplatine 400 Verbindungspunkte	2.55	Farnell
Batterieanschluss	1	Batteriehalter, Lasche, bedrahtet, (9V)	0.74	Farnell
9V Batterie	1	Batterie, 12er-Pack, 9 V, Schnappverbindung	24.45	Farnell
Schaltdrähte	10	Überbrückungsdrähte, 65-teiliger Satz	3.14	Farnell
Krokodilklemmen				
Testkabel	5	Krokodilklemmen Kabel 10 Stk. farbig	5.00	Shop of Things
Servomotor	1	Micro Servo 9g - 180° For Arduino and RPi	6.00	Farnell
Drehpotentiometer	1	Drehpotentiometer, 10 kOhm, Toleranz 20%	1.01	Farnell
Taster	2	Taster, oben betätigen, Durchsteckmontage	0.19	Farnell
LED Blau 5 mm T-1 3/4	2	LEDs für Durchsteckmontage – einfarbig	0.27	Distrelec
LED Grün 5 mm T-1 3/4	2	LEDs für Durchsteckmontage – einfarbig	0.19	Distrelec
LED Rot 5 mm T-1 3/4	2	LEDs für Durchsteckmontage – einfarbig	0.09	Distrelec
Widerstände (3 Werte) 1k2, 3k3, 6k8 Ohm	je 2	Widerstände für Durchsteckmontage Toleranz ca.1%, 250-600mW	ca. 0.12	Farnell
Stiftleiste	2x3 (1x6)	Stiftleiste gerade, 1-reihig, Raster 2.54mm	1.22	RS Components
Wasserbehälter, Strohalm	1	Im Fachhandel erhältlich		Supermarket
Elastik, Holzstab	je 2	Im Fachhandel erhältlich		Bastelshop
Nagel, Pflanztopf	---	Im Fachhandel erhältlich		Bastelshop

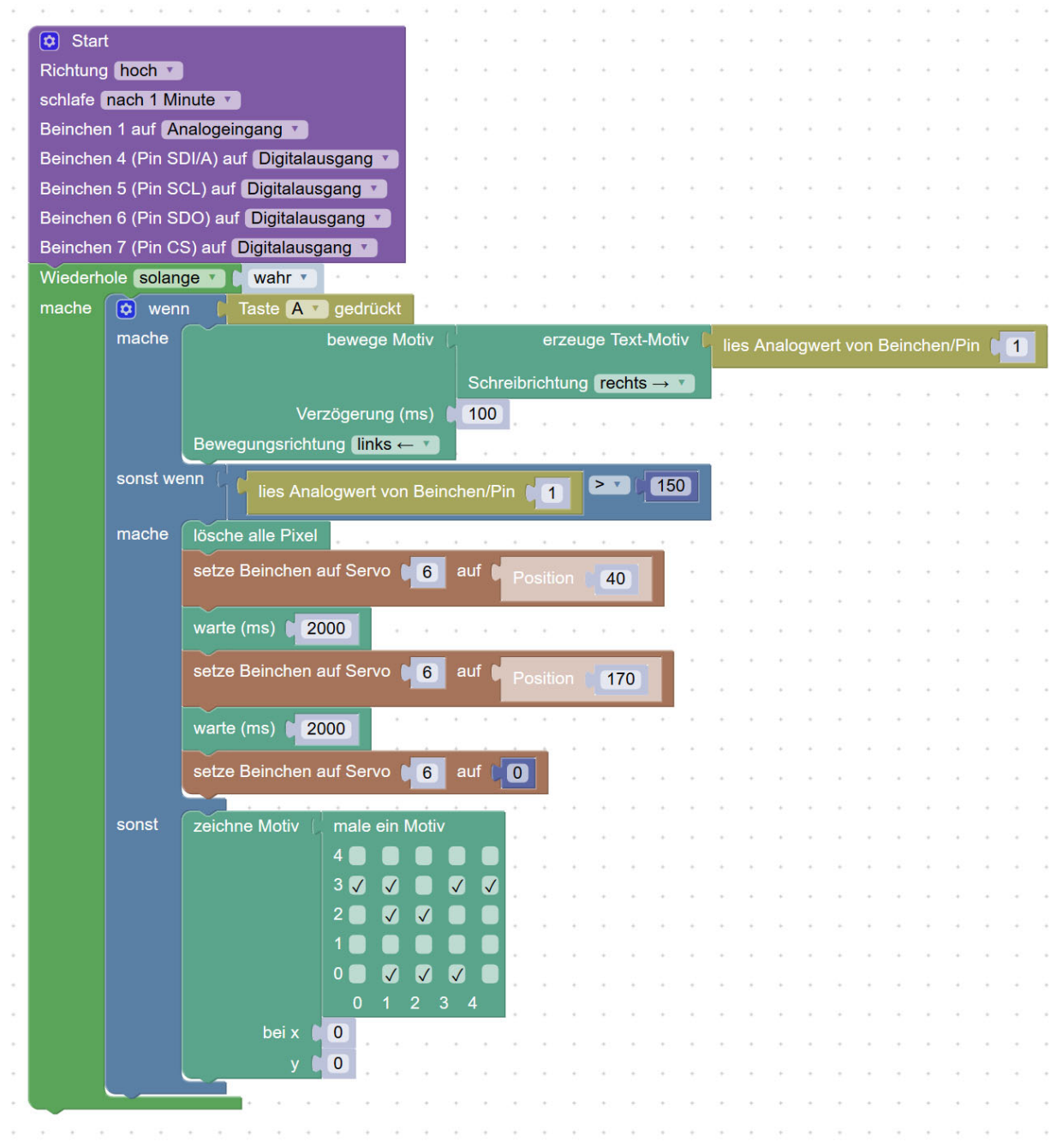
Bitte beachten: Preise können sich schnell ändern.

Codebug	Widerstände + LEDs + Taste + Stiftleiste	Steckplatine
		

Batterie + Halter	Schaltdrähte + Kokodilklemmen-Kabel	Servomotor
		

VI. Pflanztopf Giesser Programm für CodeBug

Mögliche Lösung



Link für Projekt in <http://www.codebug.org.uk> :

http://www.codebug.org.uk/explore/codebug/yqWhNRUjsUqCZBMZZfeQLN/project_plantpot_watering_step3/

VII. Quellenverzeichnis

Die Bilder aus eigener Gestaltung, und aus colourbox.com (über ZHAW Lizenz) werden nicht explizit gekennzeichnet. Weitere Quellen für die einzelnen Bilder werden unten gelistet.

Bildnachweis für Inhaltsverzeichnis

- i. <https://www.kleine-ingenieure.de/leitende-knete/stromkreise/>
- ii. <http://www.codebug.org.uk/learn/activity/60/what-is-codebug/>
- iii. <http://www.codebug.org.uk/learn/course/getting-started-codebug/activity/2/scrolling-name-badge/>
- iv. <http://www.codebug.org.uk/learn/step/562/introduction/>
- v. <http://www.learningaboutelectronics.com/Articles/Voltage-divider-resistor-calculator.php>
- vi. <http://www.codebug.org.uk/learn/activity/94/plant-pot-water-detector/>

Bildnachweis für Abschnitt Grundlage Stromkreis)

- vii. en.wikipedia.org/wiki/Light-emitting_diode
- viii. http://www.linetec.nl/electronics/leds/led_1.html

Bildnachweis für Abschnitt Steckbrett

- ix. <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard>

Bildnachweis für Abschnitt Steuerung Servomotor

- x. <http://www.codebug.org.uk/blog/codebugs-expansion-port>
- xi. <http://www.codebug.org.uk/learn/step/563/wiring-up-the-servo/>

Bildnachweis für Anhänge (Herausforderungsaufgabe, Komponenten)

- xii. https://people.eecs.berkeley.edu/~boser/courses/49_sp_2019/N_gpio.html
- xiii. <https://www.electronicshub.org/resistor-color-code/>
- xiv. <https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code>