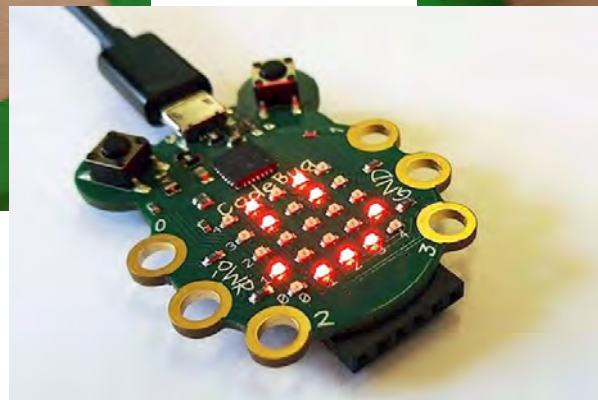
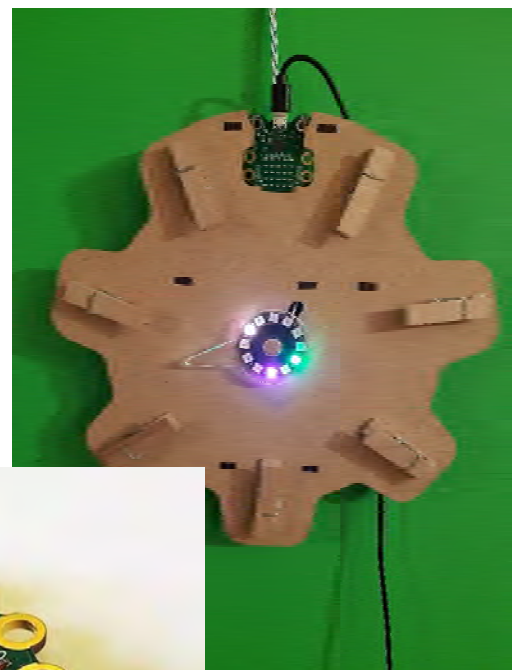
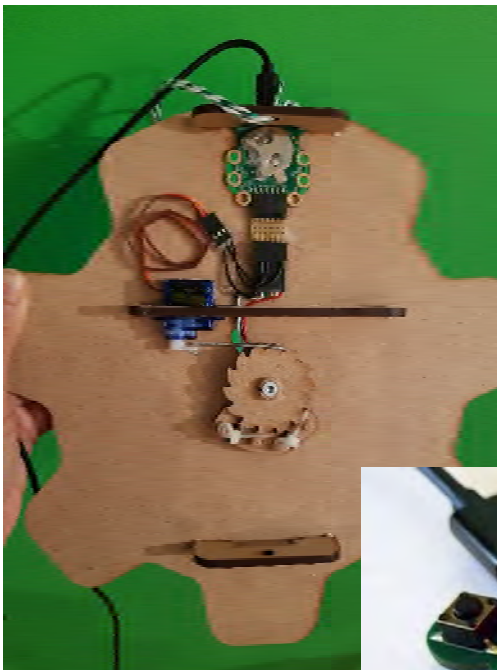


Workshop

WochenUhr

Teil 2: das Programm



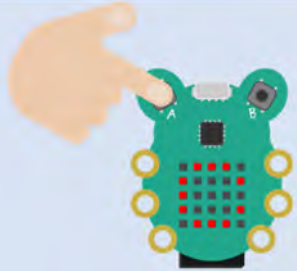
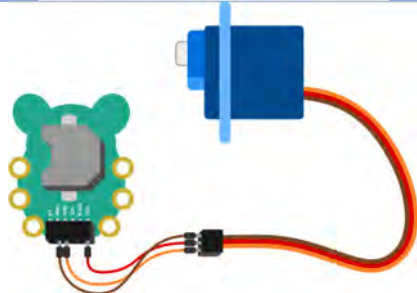





Ziel

Die frisch gebaute WochenUhr testen und sie mit Bewegung und Licht zum Leben erwecken. Dazu schreiben wir drei Programme:

- den Uhrzeiger mit dem Servomotor zu bewegen
- den LED-Ring mit verschiedenen Farben zu beleuchten
- die Tageszeit mit Zählen und Warten zu verfolgen

Für jüngere Kinder können alternativ unplugged Aktivitäten durchgeführt werden, um das fertige WochenUhr-Programm verständlich zu machen.

Inhalt

Kapitel	Aktivität	Eindruck
(1) CodeBug kennlernen + Basisübungen	<ul style="list-style-type: none"> Was ist was beim CodeBug? Wie kann man ihn programmieren? Selber ausprobieren: LED-Matrix mit Symbolen animieren. 	
(2) Steuerung eines Servomotors	<ul style="list-style-type: none"> Erweiterung Port kennenlernen Servomotor einbinden und programmieren: Zeiger des Servomotors bewegen, wenn eine Taste gedrückt wird. 	
(3) LED-Ring leuchten lassen	<ul style="list-style-type: none"> LED-Ring anschliessen und Einstellung für Programm kennenlernen Regenbogen programmieren 	
(4) Eine Uhr Programmieren	<ul style="list-style-type: none"> Sekunden warten und zählen Stunden warten und zählen LEDs als Anzeige leuchten lassen 	
(5) Komplettes Programm	<ul style="list-style-type: none"> WochenUhr fertiges Programm herunterladen Uhrzeit anpassen Anleitung zum Starten zu Hause 	
(6) Unplugged Aktivität Zustand-Spiel	<ul style="list-style-type: none"> Wie läuft das Programm der WochenUhr? Ausprobieren, und Zeichnung bauen 	
(7) Uhrzeit Genauigkeit	<p>Wie genau ist die Uhrzeit der WochenUhr?</p> <ul style="list-style-type: none"> Messen Fragen stellen Ideen sammeln 	

1. CodeBug kennenlernen + Erstes Programm

1.1. CodeBug Platine kennenlernen

Entdecken wir die verschiedenen Teile der Platine des Mini-Computers CodeBug.

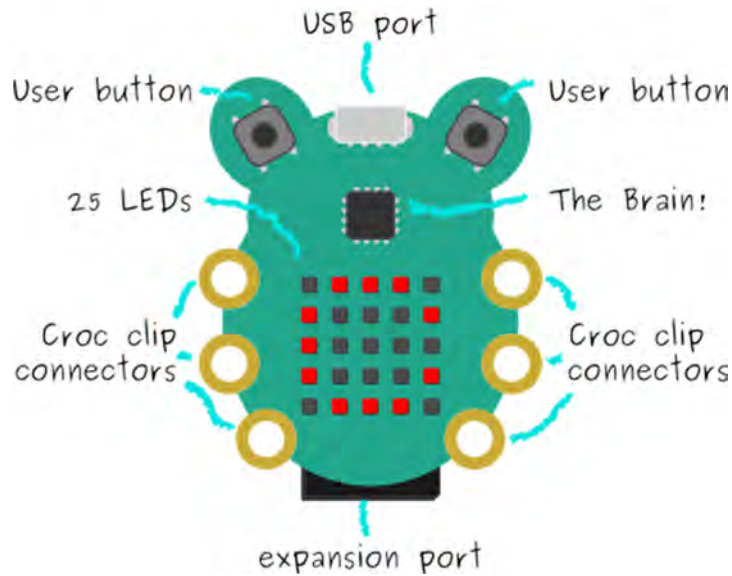


Abbildung 1 CodeBug-Platine

Welches Teil ist das?	Versuchen Sie für jedem Teil in der CodeBug-Figur (Abbildung 1 CodeBug-Platine) eine Tätigkeit zuzuordnen.
USB Port	Hier kann man ein USB-Kabel verbinden, zur Speisung und um Programme herunterzuladen.
	Interaktion durch die Benutzerin oder den Benutzer (man kann ihn drücken).
	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.
	Das kann man anfassen oder Kabel mit einer Krokodilklemme daran fixieren. 
	Hier kann man eine Erweiterung anschliessen. Zum Beispiel einen LED-Ring, oder den Servomotor. 

CodeBug Programmieren

1.2. CodeBug Programmierung-Umgebung

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 2 zeigt einen Schnappschuss der **Programmierungsumgebung**, auf der man landet, wenn man auf «Create» klickt.

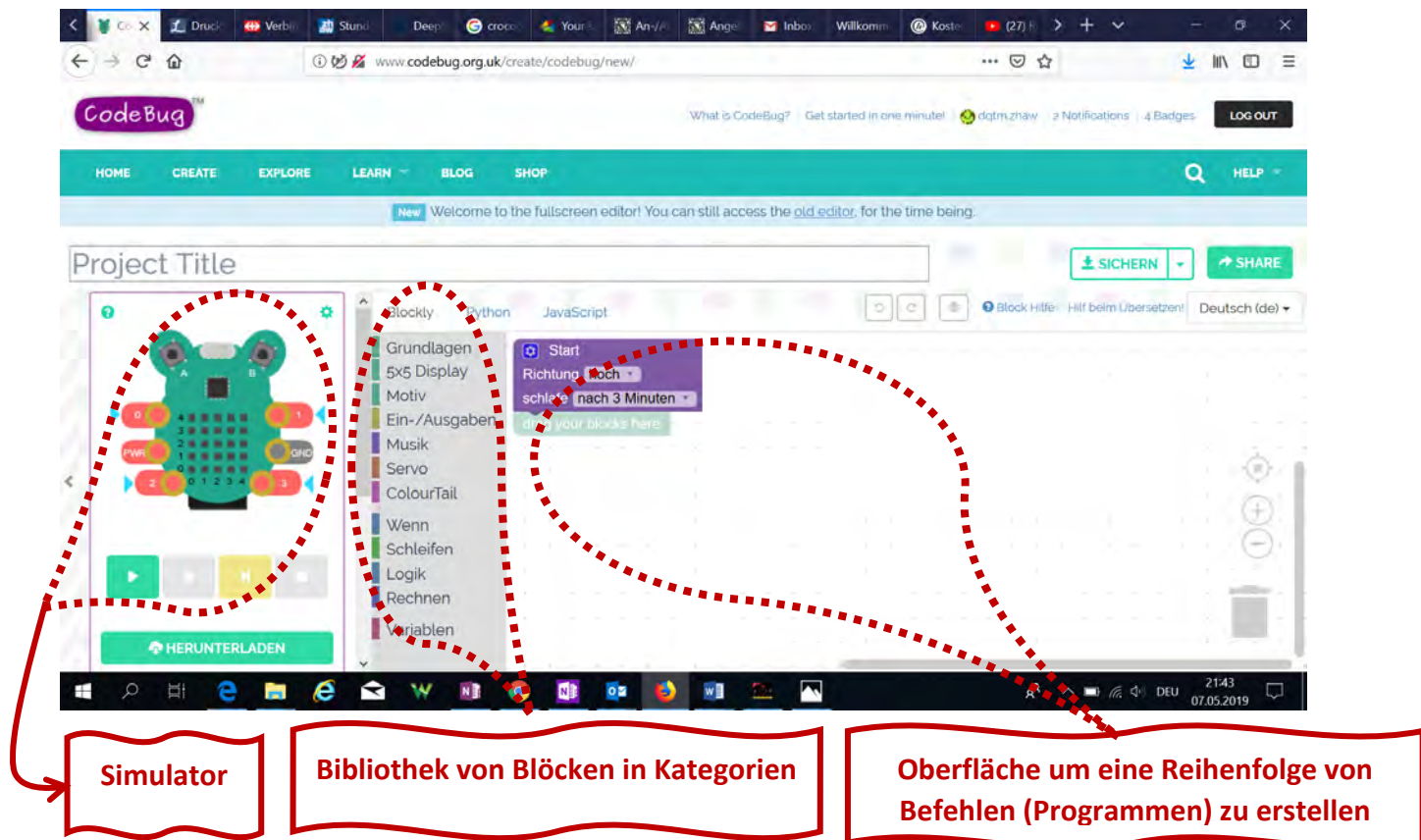


Abbildung 2 CodeBug Programmierungsumgebung

Auf der linken Seite von **Fehler! Verweisquelle konnte nicht gefunden werden.** 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 an.

1.3. Erstes Programm erstellen & herunterladen

Erstes Programm

Du wirst nun dein erstes Programm erstellen. Rechts auf dem Bild siehst du wie das Programm aussehen sollte. Jedes Programm beginnt beim violetten Start-Block. Links vom Start-Block ist ein Menu in dem du die weiteren Blöcke findest.

Der erste Block, den du suchen sollst, heisst «Warte bis Taste ändert». Klicke auf **Grundlagen** um den Block zu finden. Ziehe ihn danach mit der Maus unter den Start-Block.

Die nächsten zwei Blöcke findest du im Menu unter **Motiv**.



Abbildung 3 Erstes Programm

Simuliere dein Programm

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

Danach drücke auf Taste A oder B und siehe was passiert.

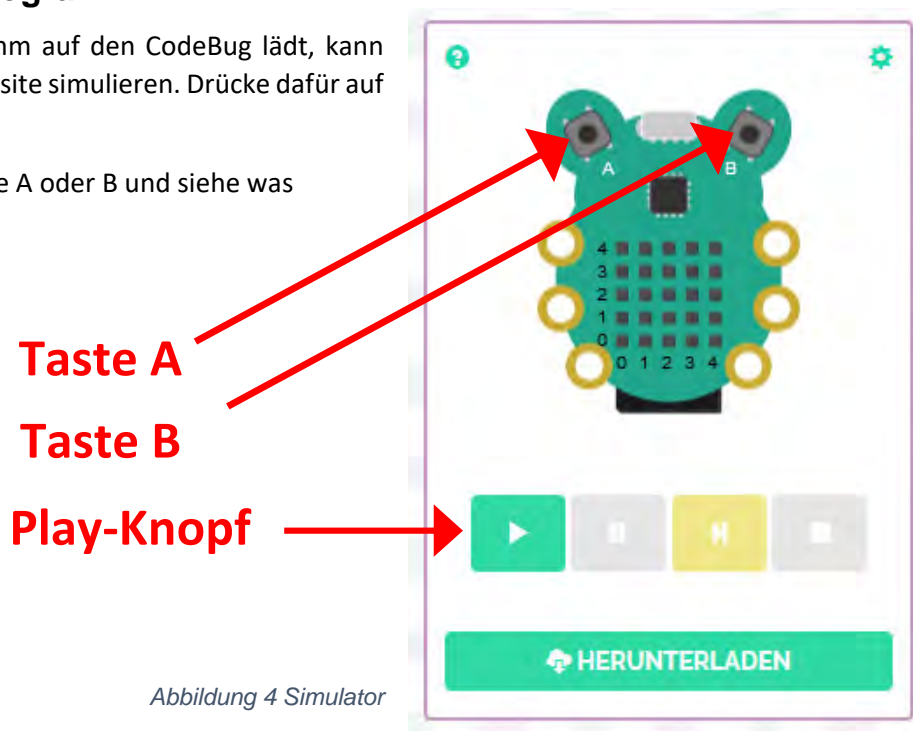


Abbildung 4 Simulator

Lade dein Programm auf den CodeBug

Klicke auf «Herunterladen». Es erscheint die folgende Anleitung. Lese diese durch und lade dann wie beschrieben das Programm auf den CodeBug.



Abbildung 5 Programm herunterladen

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 du Taste A oder B drückst.
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.

Erstelle das folgende Programm

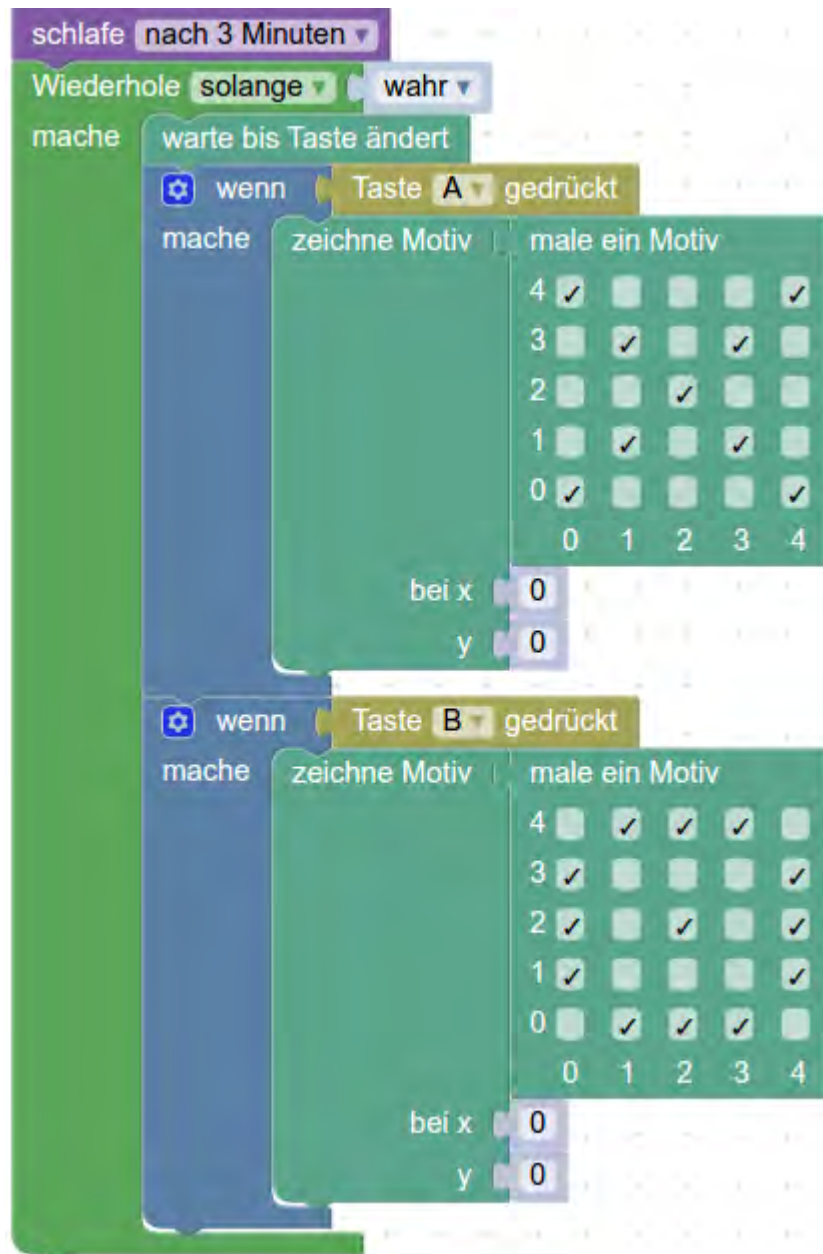


Abbildung 6 Zweites Programm

Simulieren und Herunterladen

Teste das neue Programm auf dem Simulator. Wenn es funktioniert, lade es auf den CodeBug und beantworte die folgenden Fragen:

- o Dieses Programm läuft immer weiter, weil es eine Dauerschleife hat. Wo ist die Dauerschleife, und was muss man alles drin packen?
- o Gibt es andere Typen von Schleifen? Wann würdest du diese anderen anwenden?

Es gibt noch viel mehr zum ausprobieren. Du kannst sehr schönen Ideen hier finden: <http://www.codebug.org.uk/learn/>

2. 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 7 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

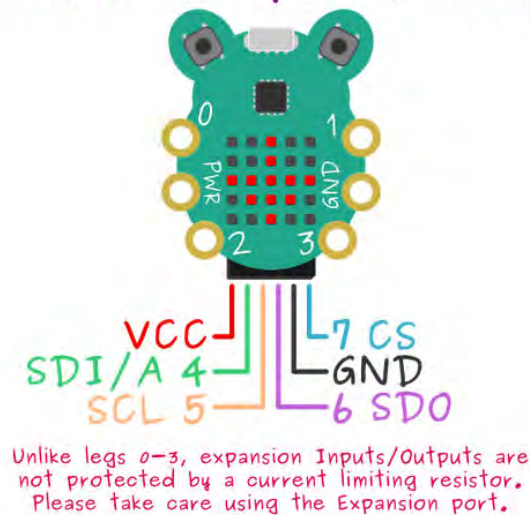


Abbildung 7 CodeBug Erweiterungs-Port und Pin-Belegung Beispiel

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 8 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 9 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.

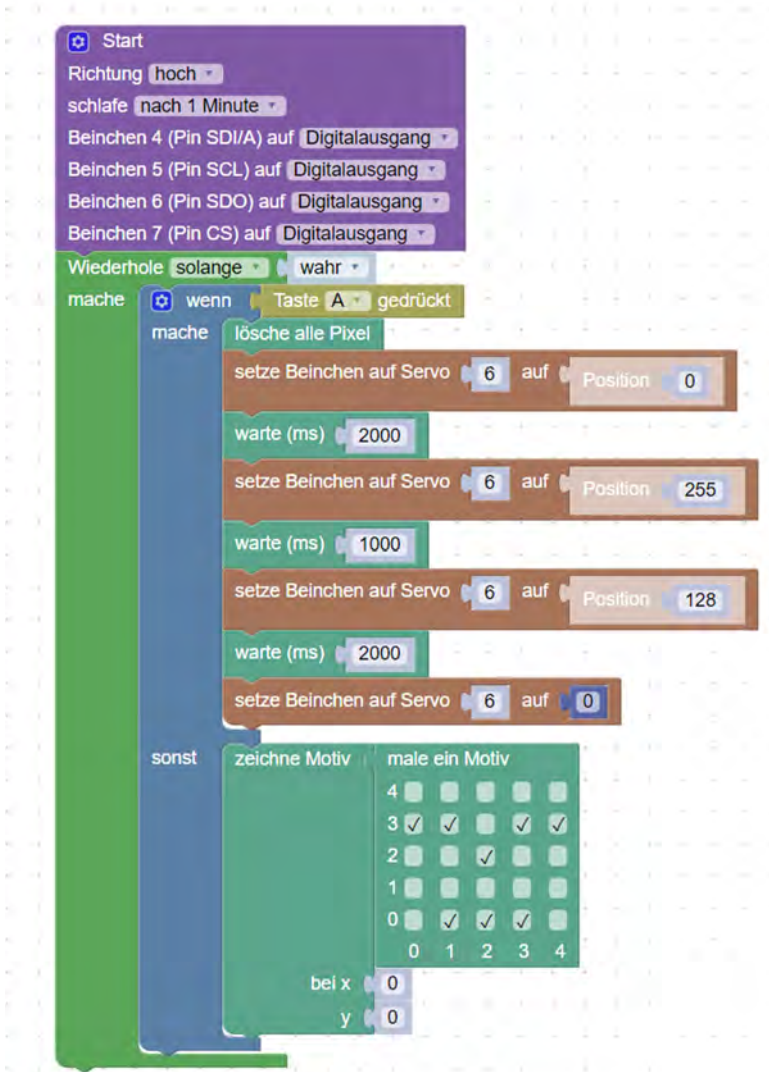


Abbildung 9

Beispiel Programm für Steuerung eines Servomotors verbunden mit

Ground / VCC / SDO
Erde / Speisung / Data

Verbinden Sie jetzt den Servomotor, wie in Abbildung 10, und probieren Sie das Programm oben durchzuführen. Sie können das Programm unter dem folgenden [Link](#)* finden, oder indem Sie in der Suchleiste der CodeBug-Webseite den Projektnamen „test_servomotor“ eingeben.

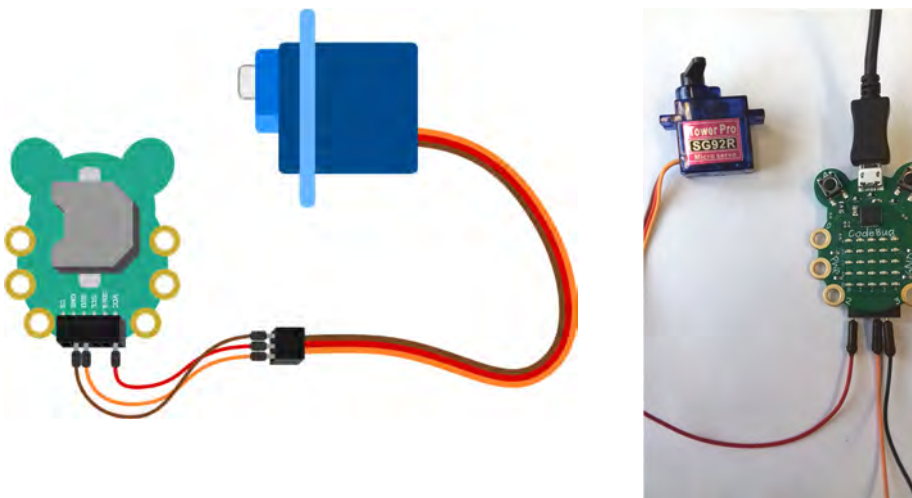


Abbildung 10

Servomotor Anschluss mit dem Erweiterungs-Port

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

3. LED-Ring leuchten lassen

LED-Ring an CodeBug anschliessen und Programm für LED-Ring vorbereiten

1. Schliessen Sie den LED Ring am CodeBug an.

ACHTUNG: Das einzelne Kabel muss beim CodeBug bei der 2 und die beiden Kabel bei der 3 sind.

2. Erstellen Sie ein neues Projekt (Create + NEW CODEBUG PROJECT)
3. Geben Sie dem Projekt den Namen: «LED-Ring – Regenbogen» und Klicken Sie auf «Sichern».
4. Damit Sie den LED Ring programmieren können, müssen Sie den violetten Startblock anpassen. Klicken Sie dazu auf das Zahnrad links oben im Startblock. Aktivieren Sie «schalte den Farbschweif ein». Mit einem Klick auf das Zahnrad verschwindet die Auswahl wieder.
→ Der Startblock hat jetzt eine Zeile mehr: «aktiv ColourStar (als Schweif)».

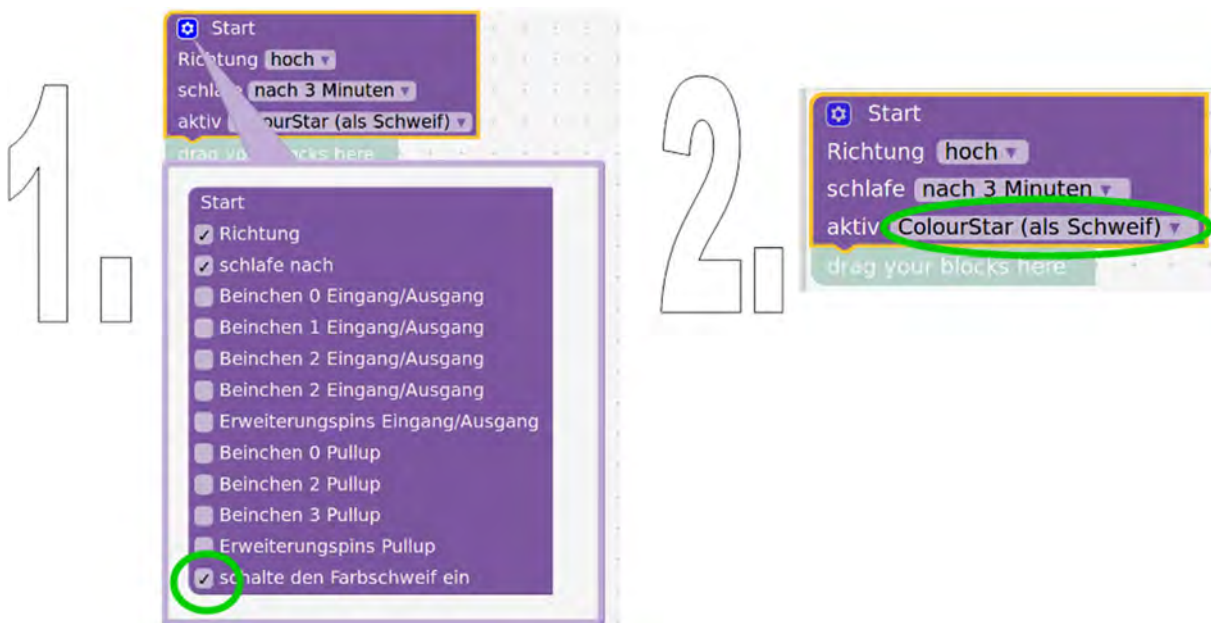
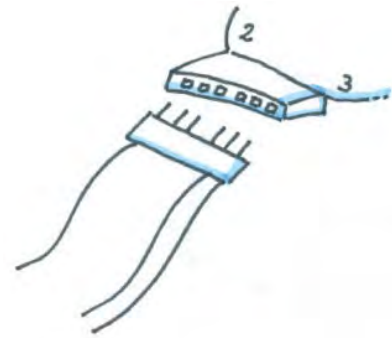


Abbildung 11: LED-Ring braucht eingeschalteten ColourStar-Farbschweif

Aufgabe Regenbogen Programmieren

LED-Ring leuchten lassen

Ihre LED-Ring soll sich in einen Regenbogen verwandeln, wenn man die Taste-A drückt, und alles löschen, wenn man die Taste-B drückt. Erstellen dazu Sie ein Programm, welches auf Ihrem Ring die LEDs in den Farben des Regenbogens erstrahlen lässt.



Tipps

- Benutzt den Befehl <setze Farbepixel # auf Farbe ...> in der Block-Bibliothek unten Kategorie «ColourTail» an.

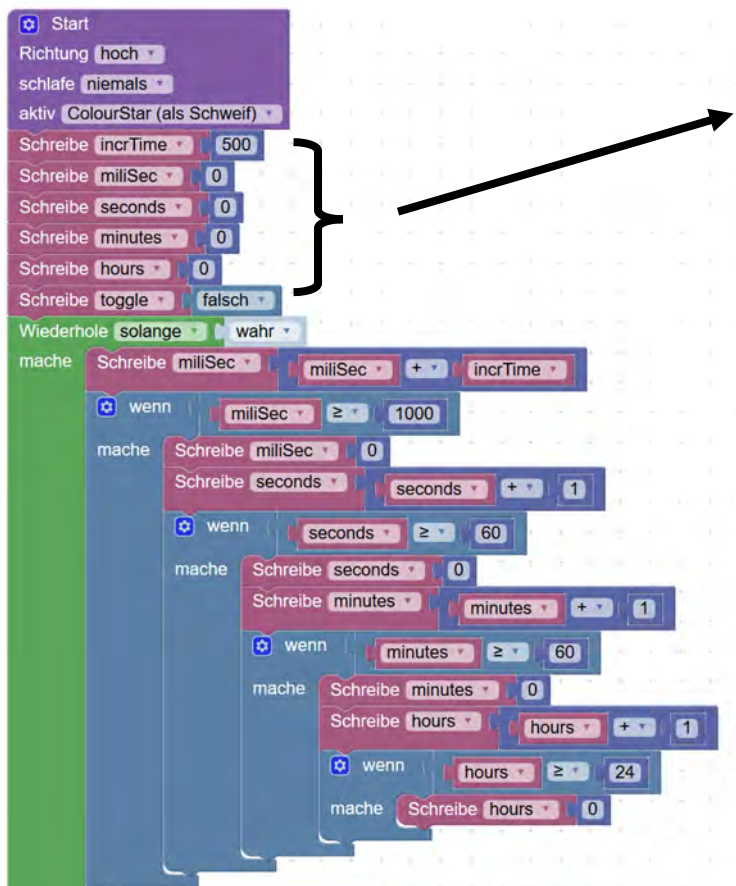


- Denke daran eine Dauerschleife zu benutzen, so dass dein Programm immer wieder läuft.
- Diese Aufgabe kann (wie jedes Programm) mit verschiedenen Lösungen erfüllt werden. Probiere es zuerst selbst aus, und prüfe erst dann die Beispiellösung am Ende dieses Handouts.

4. Eine Uhr Programmieren

Nun wollen wir eine Uhr programmieren! Dazu müssen wir die Zeit verfolgen, indem wir warten und die Sekunden, Minuten und Stunden zählen. Dieses Programm ist schon recht komplex, deshalb wollen wir rückwärtsgehen und eine fertige Lösungsmöglichkeit analysieren. Beantworte die folgenden Fragen zu jedem Teil des Programms.

Dieses Programm findet man unter <https://bit.ly/3e6Kj7s> (Projekt Uhr_Einfach).



Das sind die Variablen des Programms. Jede Variable hat einen Namen und einen Wert. Der Wert kann innerhalb des Programms geändert werden. Die Variable toggle ist von Typ Boolean, und kann entweder falsch oder wahr sein.

FRAGEN

- (A)
Wofür werden die Variablen benutzt?
- (B)
Wie wird der Überlauf gemacht?
Bmk: Überlauf bedeutet zurücksetzen, wenn man über das maximum Wert gezählt hat.
- (C)
Wie werden die Stunden gezählt?
Zweimal 0 bis 12, oder einmal 0 bis 24?

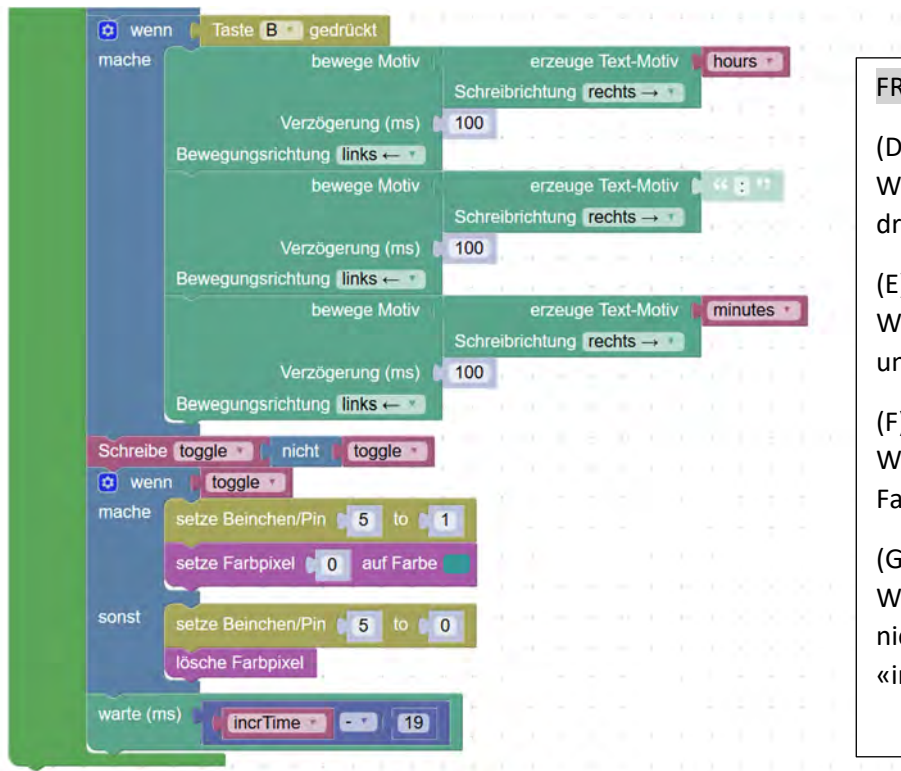


Abbildung 12 Programm einfache Uhr

FRAGEN

(D)

Was passiert, wenn man die Taste B drückt?

(E)

Welche Wert hat toggle in 1ten, 2ten und 3ten Durchgang der Schleife?

(F)

Was passiert dann mit Pin 5 und Farbpixel 0?

(G)

Was könnte sein, warum wartet man nicht genau incrTime, und sonst «incrTime-19» ?

5. Komplettes WochenUhr-Programm

Das komplette WochenUhr Programm kann man unter <https://bit.ly/3nBoNL2> (Project WochenUhr_Servo_12Std_Stunden_Blau) finden. Dieses Programm enthält mehrere Funktionen:

- Uhrzeit zählen
- Uhrzeit anzeigen im LED-Ring: Stunden in blau, Minuten in grün, Sekunden in Rot
- Servomotor bewegen jede 12 Stunden (Zeiger geht vorwärts in Halbtags-Takt)
- Benutzer Bedienung Schnittstelle, um die Uhrzeit anzupassen

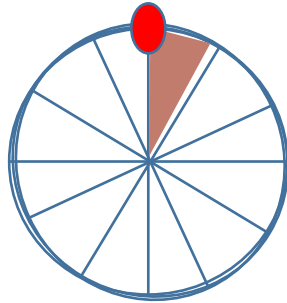
Für Testzweck kann man auch eine Variation des Programms nehmen, wo man den Servomotor jede Minute bewegt <https://bit.ly/2QzrL73> (Project WochenUhr_Servo_1Min_Stunden_Blau).

Lade das fertige Programm auf den CodeBug herunter, und schauen wir uns an, wie die Anzeige auf dem LED-Ring funktioniert und wie man die Uhrzeit einstellt.

Der LED-Ring hat 12 LEDs. Das reicht für die Stunden, aber wir müssen die Minuten und Sekunden etwas kompakter anzeigen. Wir gruppieren die Minuten in 12 5er-Gruppen, so dass die 1. LED leuchtet, solange die Minuten zwischen 0-1-2-3-4 liegen, dann leuchtet die 2. LED für Minuten, die 5-6-7-8-9 entsprechen, und so weiter. Die Sekunden blinken in Rot. Die Tabelle unten fasst die LED-Ring Uhrzeit-Anzeige zusammen.

LED-RING UHRZEIT-ANZEIG

Blau	Stunden
Grün	Minuten 12 Positionen möglich: 1te LED : $0 < \text{Minuten} < 4$ 2te LED : $5 < \text{Minuten} < 9$ 3te LED : $10 < \text{Minuten} < 14$ 4te LED : $15 < \text{Minuten} < 19$ 5te LED : $20 < \text{Minuten} < 24$... 11te LED : $55 < \text{Minuten} < 0$
Rot	Sekunden 12 Positionen (wie die Minuten)



Jedes Mal, wenn man die WochenUhr einschaltet, oder nach ein paar Tagen, wenn die Uhr ein wenig daneben steht, kann man die Zeit einstellen. Das Programm hat einen sogenannten Automaten, der über drei Zustände läuft und zwischen denen die BenutzerIn umschalten kann: Los (die Uhr läuft), Stunden (neuer Wert setzen), Minuten (neuer Wert setzen). Die Abbildung unten zeigt die Verwendung: mit A schaltet man in den nächsten Zustand, und mit B erhöht man den Wert der Stunden oder Minuten.

UHRZEIT ANPASSEN

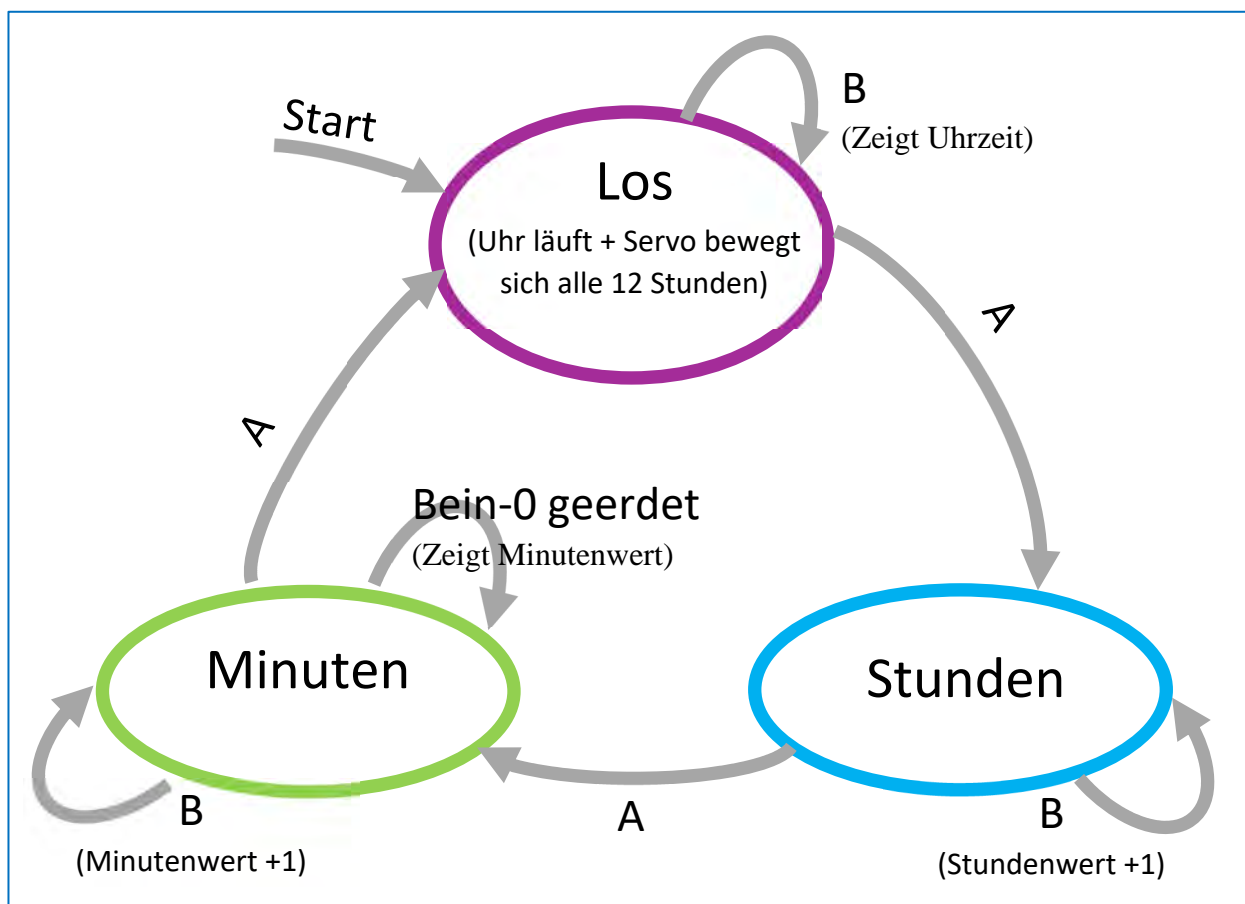
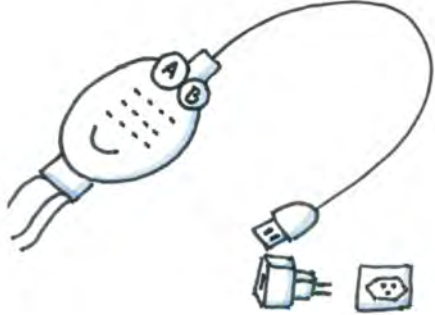


Abbildung 13 Automat für Uhreneinstellung

Anleitung zum Starten zu Hause

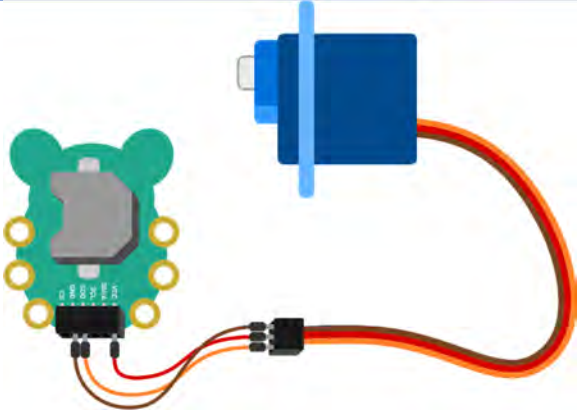
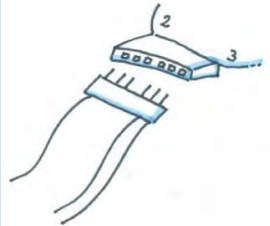
SCHRITTE

(1) USB-Speisung besorgen und WochenUhr anschliessen	
(2) Aktuelle Zeit anpassen	Stunden- und Minuten-Wert Man muss das ab und zu wiederholen, weil die Uhr nicht 100% genau ist.
(3) Wochentag Karte	Gestalten und hängen

Das WochenUhr Programm und Verbindungen

Wenn du zu Hause mit dem CodeBug weiter experimentierst, kannst du deinen CodeBug herausnehmen und neue Programme schreiben und ausprobieren.

Dann, wenn du wieder den CodeBug als WochenUhr benutzen willst, muss du nur das WochenUhr Programm runterladen und die LED-Ring und den Servo nochmals verbinden.

Schritt	Referenz
WochenUhr Programm auf den CodeBug runterladen	Link fürs Programm: https://bit.ly/3nBoNL2 (Project WochenUhr_Servo_12Std_Stunden_Blau)
Servo Motor verbinden	
LED-Ring verbinden	

6. Unplugged Aktivität: Zustand-Spiel

Wie läuft das Programm der WochenUhr?

Unsere WochenUhr hat ein Minicomputer (der CodeBug), wo ein Programm gespeichert wird. Dieses Programm definiert was die WochenUhr machen kann, und wie es gemacht wird.

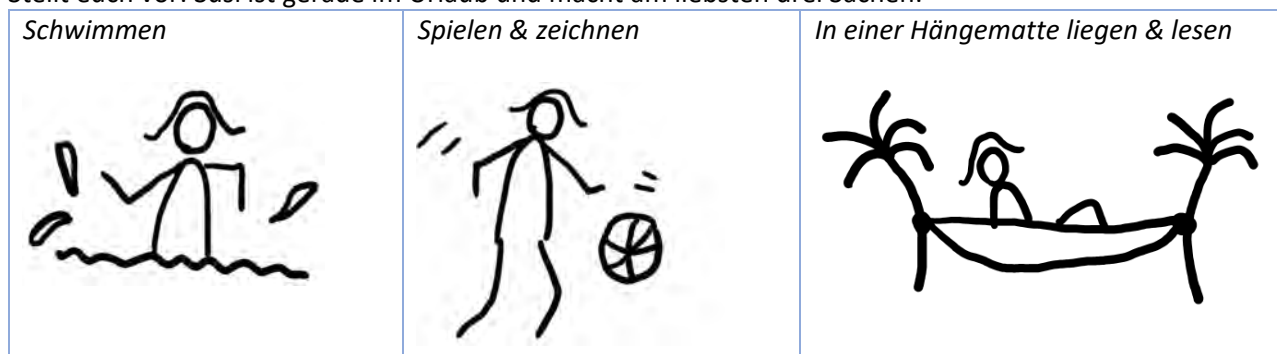


**Microkontroller: Gehirn
des Minicomputers.**

**Hier drin ist das
Programm gespeichert.**

Zum Beispiel: wenn ich Taste A drücke passiert das und das, oder wenn ich Taste B drücke passiert dies und jenes. Und was beschreibt das Programm der WochenUhr? Bevor wir uns das anschauen, müssen wir etwas Neues lernen, den sogenannten **Zustand**.

Stellt euch vor: Susi ist gerade im Urlaub und macht am liebsten drei Sachen:



Morgens steht Susi auf und spielt zuerst. Dann, wenn es wärmer wird, geht sie schwimmen und so weit. Bauen wir das zusammen in einer Zeichnung mit Blasen (Zustände) und Pfeilen (Fragen oder Bedingungen für wechseln).

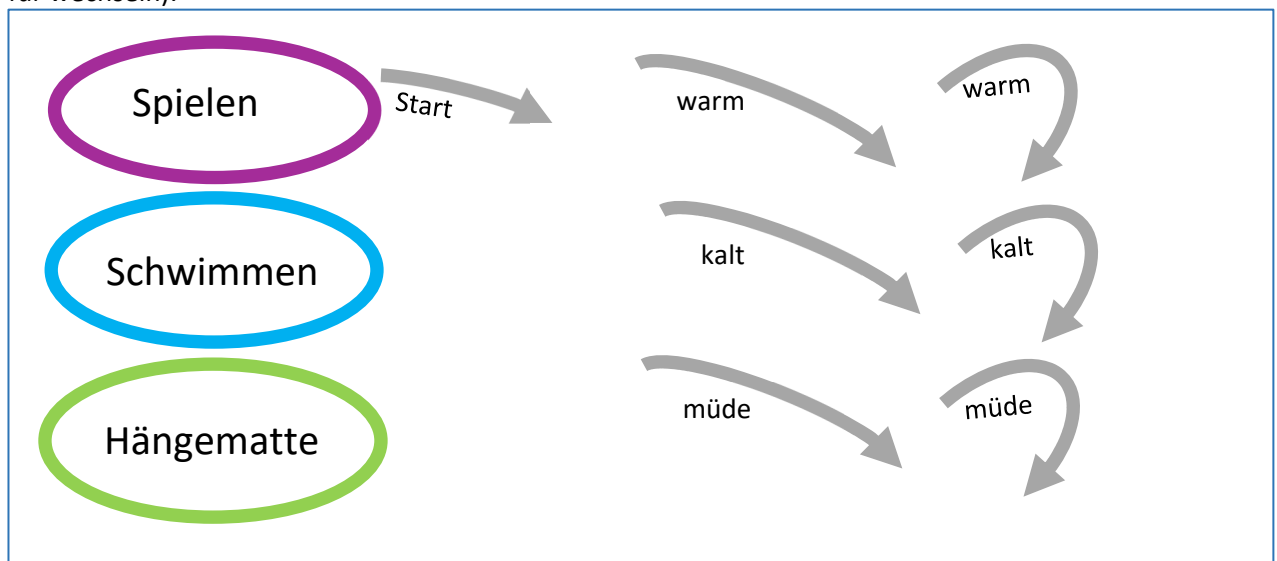


Abbildung 14 Teile für unplugged Spiel

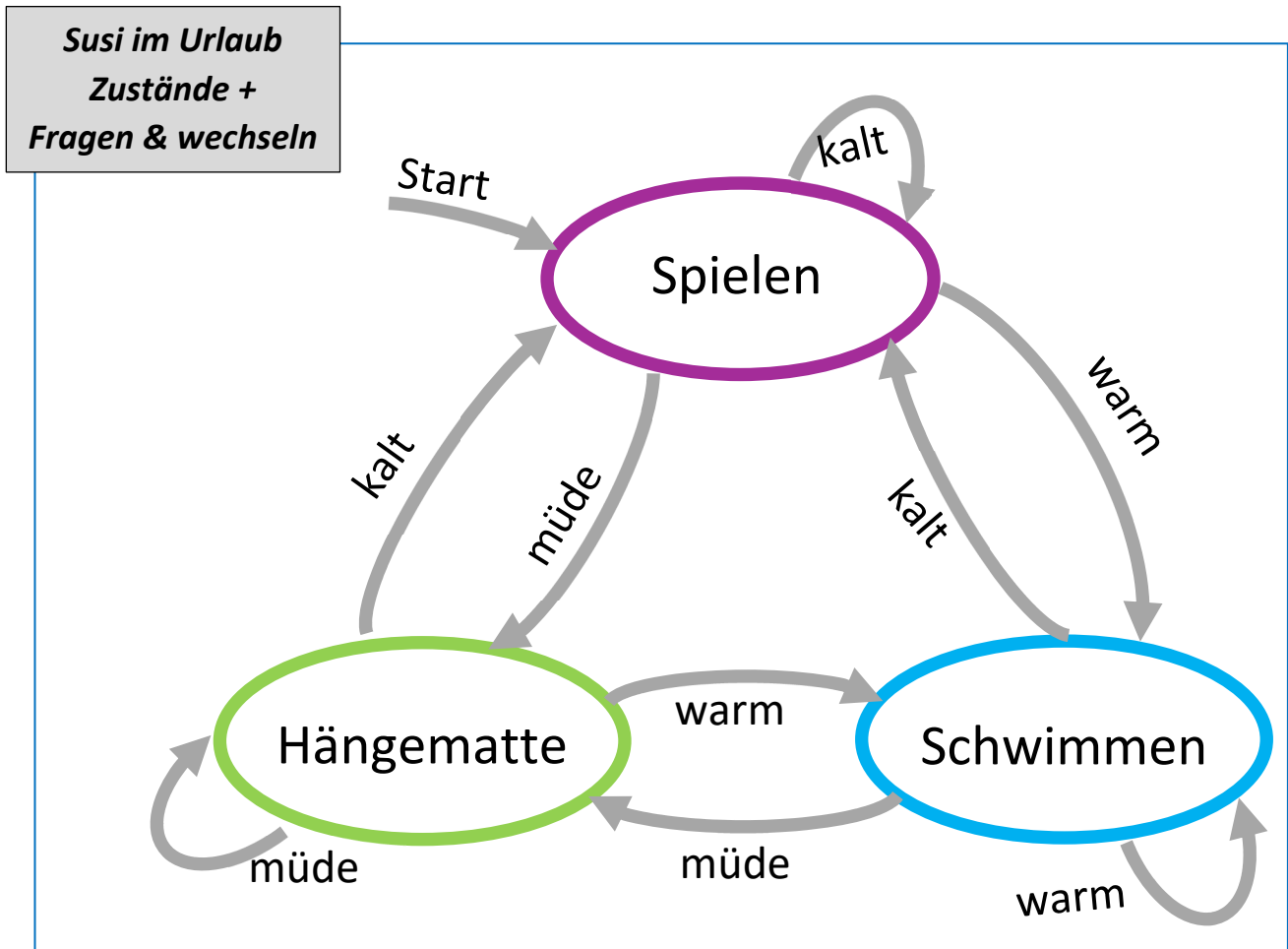


Abbildung 15 Zustandsdiagramm Susi in die Ferien

Schau mal die Zeichnung oben und versuch zusammen mit deiner Nachbarin oder deinen Nachbar zu beantworten:

- Was macht Susi zuerst?
- Wann geht Susi schwimmen?
- Wann und wie lange liegt Susi in der Hängematte?

Die drei Blasen (Spielen, Schwimmen, Hängematte) sind die Zustände von Susi im Urlaub.

Versuch jetzt die Zustände deine WochenUhr herauszufinden.

Tipp: In der WochenUhr wechselt man von einem zu dem nächsten Zustand, indem man die Taste A drückt.

Neben die Zustände und die Fragen (für wechseln) wollen wir auch die Aktionen, die in jedem Zustand passieren können, herausfinden.

Nimm jetzt einen Papier-Set mit farbigen Zustände plus Aktionen-Karten, und baut eine Zeichnung, die zeigt, wie die WochenUhr funktioniert. Zum Beispiel, muss man herausfinden:

- Wann passiert was? Im welchen Zustand kommt jede Aktion?
- Wie ändert man zum neuen Zustand?

7. Uhrzeit Genauigkeit

Wie genau ist die Uhrzeit der WochenUhr?

Leider nicht so genau... Checken wir das zusammen mit einer Messung:

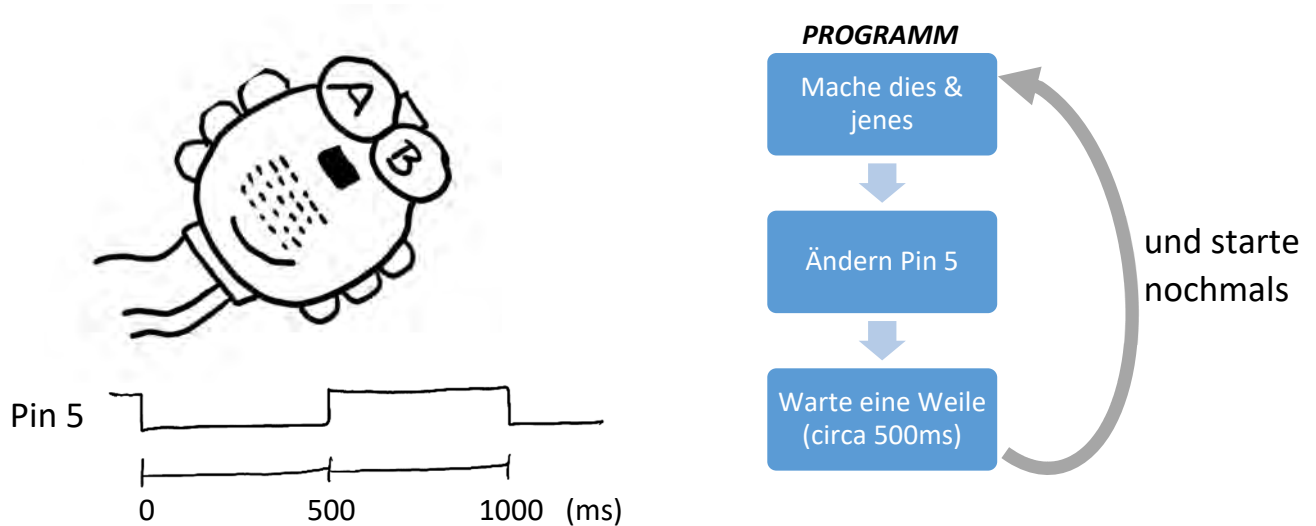


Abbildung 16 Uhrzeit folgen und Genauigkeit messen

Was glaubst du:

- Warum könnte es sein, dass unsere Uhr nicht sehr genau läuft?

Ideen	Antworten

- Wie kann man das verbessern?

Ideen	Antworten

ANHANG

Lösung zur Aufgabe Regenbogen

Auch dieser Auftrag kann (wie jedes Programm) mit unterschiedlichen Lösungen erfüllt werden.

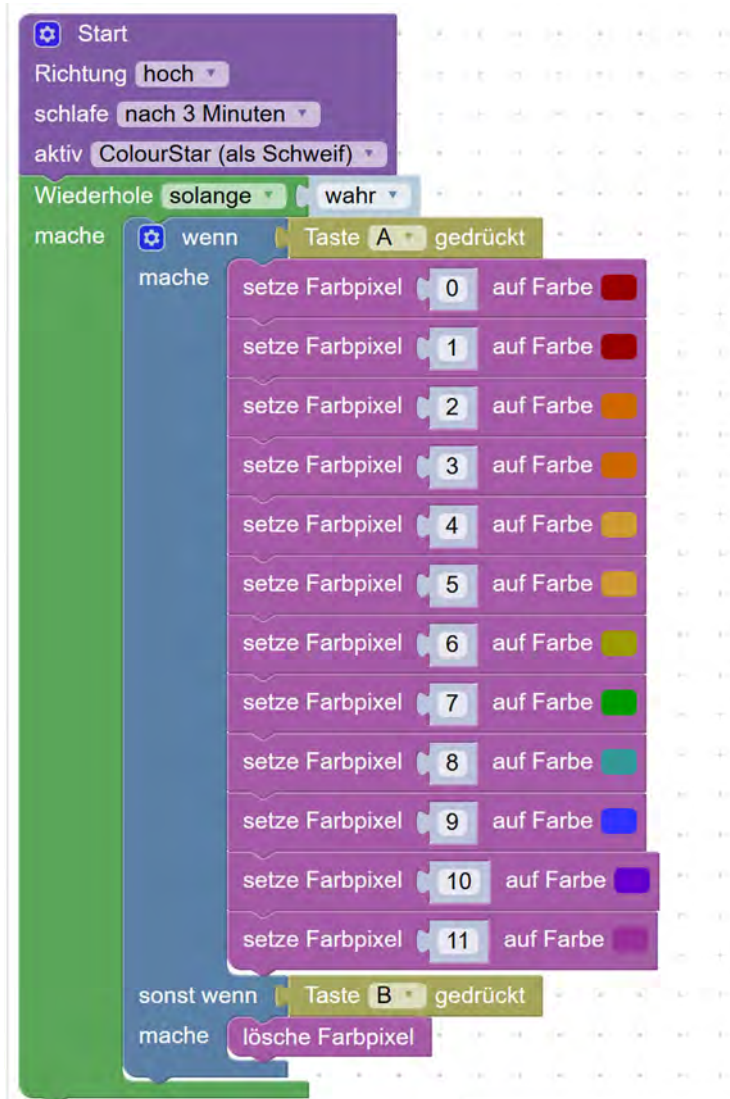


Abbildung 17: Beispiellösung zu Aufgabe – Regenbogen