



# CALLIOPE

## Lehrerhandreichung

Hans Haase, André Hilbig, Ludger Humbert,  
Dorothee Müller und Philipp Rumm



*Bergische Universität Wuppertal*

<http://ddi.uni-wuppertal.de/>



Revision 0.901

letzte Änderungen: 16. Februar 2017 von *haase*

aktuelle Version: <http://uni-w.de/a5>

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>2</b>
<b>1 Aufbau dieser Handreichung</b>	<b>3</b>
<b>2 Informatische Bildung ab der Grundschule</b>	<b>5</b>
2.1 Was ist Informatik? . . . . .	5
2.2 Die Phänomenbereiche der Informatik . . . . .	6
2.3 Die Fachgebiete der Informatik . . . . .	7
2.4 Der informatische Modellierungskreislauf . . . . .	8
2.5 Zu erwerbende Kompetenzen . . . . .	9
<b>3 Grundlegende Sachinformation</b>	<b>11</b>
3.1 Algorithmen . . . . .	11
3.1.1 Sequenzen . . . . .	11
3.1.2 Kontrollstrukturen . . . . .	11
3.2 Technische Informatik . . . . .	12
3.3 Codierung . . . . .	14
3.3.1 Das Binärsystem . . . . .	14
<b>4 Materialien für den Unterricht</b>	<b>16</b>
Übersicht über die Unterrichtseinheiten . . . . .	17
Kurzanleitung <i>Calliope mini</i> . . . . .	18
Unterrichtseinheit 1: Technische Informatik – Schaltungen . . . . .	19
Unterrichtseinheit 2: Algorithmen – Pflanzenbewässerung . . . . .	22
Unterrichtseinheit 3: Algorithmen – Keksdose . . . . .	32
Unterrichtseinheit 4: Algorithmen – Quizbuzzer . . . . .	35
Unterrichtseinheit 5: Codierung – Multimediales Büchlein . . . . .	40
Unterrichtseinheit 6: Technische Informatik – Ampel . . . . .	47
<b>5 Verwendungsnachweis</b>	<b>51</b>

## Vorwort

### Informatisierung → Digitalisierung

Informatik und Informatiksysteme als Unterrichtsgegenstände für Grundschülerinnen und Grundschüler? Können Schülerinnen und Schüler dieser Altersgruppe überhaupt verstehen, was Informatik ist?

**Diese Fragen beantworten wir mit Ja!**

Informatik wird von vielen Menschen als Geheimwissenschaft angesehen. Dieser Vorstellung setzen wir mit Ideen, die auch mittels CALLIOPE MINI umsetzbar sind, Beispiele – bereits für Grundschulkindern – entgegen, die spannende, herausfordernde und wissensgenerierende Erfahrungen bereithalten.

Um diese Beispiele umsetzen zu können, muss die Grundschullehrerin nicht Informatik studiert haben, sollte aber die vorliegende Handreichung in Ruhe durchlesen. Dann besteht die Chance, dass die Kontexte für unsere Beispiele auch zu informatischer Bildung beitragen, die der Schlüssel für die digitale Welt ist.

### Informatik – wir sind das I in M\_I\_N\_T

Dass Digitalisierung – als Folge der Informatisierung – gestaltet werden kann, erleben die Schülerinnen und Schüler, wenn sie die hier vorgelegten Elemente durch Eigentätigkeit erproben, damit experimentieren und so Gestaltungskompetenz entwickeln. Dies ist die Schlüsselkompetenz, die nötig ist, um zukünftigen Herausforderungen aktiv begegnen zu können und Freude – gepaart mit Erfolgserlebnissen – daran zu haben, die eigene Zukunft im wahrsten Sinne des Worte gestaltend in die eigenen Hände zu nehmen.

Die bei den Schülerinnen und Schülern entwickelte informatische Selbstkompetenz in der Gestaltung von Informatiksystemen ist für unsere gesellschaftliche Weiterentwicklung von zentraler Bedeutung, denn informatische Mündigkeit setzt informatisches Wissen voraus.

### Hier wird gearbeitet – Danke

Stunden der Diskussion, des Abwägens und der Besprechung zur Dokumentation, an deren zielführender Gestaltung maßgeblich Frau Dr. DOROTHEE MÜLLER beteiligt war, führten zu den hier in exzellenter Form vorliegenden Unterrichtsideen und Unterrichtsmaterialien.

Vorarbeiten, die diese Form der Dokumentation möglich gemacht haben, wurden wesentlich von ANDRÉ HILBIG (Informatik- und Physiklehrer an der Gesamtschule Uellendahl-Katernberg, Wuppertal) im Zusammenhang mit seiner Arbeit im Fachgebiet Didaktik der Informatik an der Bergischen Universität Wuppertal geleistet – ihm gilt unser Dank für diese durchdachte, ausgezeichnete Basis.

Bei der Detailgestaltung ergeben sich immer wieder Fragen, wie bestimmte Elemente umzusetzen sind, die grafisch gestaltet werden müssen – wir wissen, dass die beste Typografie so zu gestalten ist, dass sie nicht negativ auffällt. Der Umsetzung dieser Anforderung in besonderem Maß verpflichtet, hat Herr PHILIPP RUMM einen wichtigen Anteil an der Erstellung dieser Handreichung.

Für die Realisierung dieser Handreichung wird das Satzsystem  $\text{\LaTeX}$  genutzt. So entstehen qualitativ hochwertige, typografisch wertvolle Materialien. Mein Dank gilt Herrn HANS HAASE, der als wissenschaftliche Hilfskraft wesentlich zu der Erstellung dieser Dokumentation beigetragen hat.

Neben den direkt im Projektzusammenhang tätigen Personen möchte ich mich bei Frau BRIGITTE SCHULTZ für die professionelle Unterstützung in allen formalen Belangen, die den Fortgang des Projekts betreffen, bedanken.

Last not least – gilt mein Dank Prof. Dr. ANDREAS FROMMER, dem Prorektor für Studium und Lehre der Bergischen Universität Wuppertal für die unterstützende Begleitung bei allen Entscheidungen der Universitätsleitung zur Ermöglichung der Umsetzung.


Die Ergebnisse sind überzeugend – Danke!

LUDGER HUMBERT  
Didaktik der Informatik  
Bergische Universität Wuppertal

### Lizenzhinweis

Dieses Dokument steht unter der folgenden Creative-Commons-Lizenz: .

Unter <https://creativecommons.org/licenses/by-sa/4.0/deed.de> sind Details zu der Lizenz dokumentiert.

Um eine Bearbeitung unter Einhaltung der Lizenz zu ermöglichen ist der Quellcode hier hinterlegt: 



# 1 Aufbau dieser Handreichung

In dieser Handreichung finden sich fachdidaktische Hinweise zur Umsetzung von informatischen Gegenständen – also sowohl von Informatikinhalt als auch von Methoden der Informatik. Die Phänomenbereiche der Informatik sowie der informatische Modellierungskreislauf werden als fachdidaktische Grundlage dargestellt. Mithilfe des *Calliope mini* werden Themenfelder aufgeschlossen. Hierbei wird zu jedem Teilbereich je ein kurzer Sachinput vorgestellt, der durch praktische Umsetzungsvorschläge und Kopiervorlagen für die Schüler- und Lehrerhand ergänzt wird.

## Allgemeine Hinweise zur unterrichtlichen Umsetzung

Die Lernaufgaben sind so ausgerichtet, dass die Schülerinnen und Schüler Ideen entwickeln, wie informatische Prinzipien funktionieren. Die Schülerinnen und Schüler erproben das Material im Sinne der Binnendifferenzierung angeleitet oder frei.

**Wissen vernetzen** Die Schülerinnen und Schüler äußern im Rahmen des Unterrichts ihre Vorerfahrungen und Vermutungen, aber auch ihre Fragen. Die Lehrkraft ermöglicht ihnen die aktive Vernetzung ihres Wissens, indem sie Anreize zur Aktivierung gibt.

**Fachsprache nutzen** Die verwendeten Fachbegriffe, wie »Algorithmus«, »Kontrollstrukturen« etc. sind für die Kinder bisher abstrakte Begriffe, die mit Bedeutung gefüllt werden müssen. Hierzu empfiehlt es sich, im Rahmen einer Wandzeitung einen Wortspeicher anzulegen, der es den Schülerinnen und Schülern ermöglicht, die eingeführten Fachwörter zu sichern, zu erinnern und anzuwenden.

**Problemlösendes Lernen** Die Schülerinnen und Schüler werden narrativ mit einer Problemstellung konfrontiert. Sie entwickeln Strategien zur Lösung des Problems und wenden dabei die informatische Modellierung an.

**Entdeckendes Lernen** Den Schülerinnen und Schülern wird die Gelegenheit gegeben, sich mit der Sache auseinanderzusetzen. Sie stellen Vermutungen zum Sachverhalt an und entwickeln Lösungsstrategien.

**Anwendungs- und Strukturorientierung** Die Schülerinnen und Schüler werden mit der Strukturwissenschaft Informatik vertraut gemacht, indem sie zu einzelnen fachbezogenen Themen Fragestellungen bearbeiten sowie Lösungen entwickeln und anwenden.

**Vernetzung verschiedener Darstellungsformen** Die Schülerinnen und Schüler lernen fachübergreifend Darstellungsformen zu nutzen und zu bewerten.

(Vgl. Fricke u. a., 2016)

## Die Unterthemen im Überblick

**Pflanzenbewässerung** Die Schülerinnen und Schüler machen sich ihr Wissen über Leitfähigkeit zunutze, um anhand trockener/feuchter Erde entscheiden zu können, ob eine Pflanze Wasser benötigt. Darüber hinaus entwickeln die Schülerinnen und Schüler einen Algorithmus, der in regelmäßigen Abständen prüft, ob eine Pflanze bewässert werden muss.

**Keksdose** Die Schülerinnen und Schüler entwickeln eine Möglichkeit zu überprüfen, wie oft Kekse aus einer Keksdose entnommen werden. Dabei machen sie sich die informatische Modellierung zunutze und erkennen, wo Vor- und Nachteile ihrer jeweiligen Modellierung liegen.

**Multimediales Heftchen** Die Schülerinnen und Schüler lernen Möglichkeiten der Codierung kennen und machen sich dieses Wissen zu nutze, um ein Heftchen mit multimedialer Ergänzung zu basteln.

## 2 Informatische Bildung ab der Grundschule

Schülerinnen und Schülern sollten so früh wie möglich – und damit schon ab der Grundschule – die verschiedenen Facetten der Informatik kennenlernen. Damit wird es ihnen ermöglicht, ein intuitives Verständnis für Informatik zu entwickeln. Die Aneignung zumindest dieses basalen Verständnisses ist eine der Voraussetzungen für ein selbstbestimmtes Leben in unserer heutigen, von Informatik durchdrungenen Gesellschaft.

### 2.1 Was ist Informatik?

Die Bezeichnung *Informatik* ist eine Vereinigung der Worte *Information* und *Automation*. Damit bezeichnet *Informatik* die »automatische Informationsverarbeitung« (nach Ernst, 2015, S. 1).

Der Informatik-Duden hält die folgende Definition bereit:

»**Informatik (Computer Science)**: Wissenschaft von der systematischen Verarbeitung von Informationen, besonders der automatischen Verarbeitung mithilfe von Digitalrechnern« (Claus und Schwill, 2006).

Möchte man sich jedoch nicht auf die Digitalrechner und informatischen Systeme, also PCs, Laptops, Smartphones, Tablets etc. beschränken und sich stattdessen mit den unserer Sichtweise auf die Wirklichkeit zugrunde liegenden informatischen Phänomenen und Strukturen befassen, so greift eher die Beschreibung des englischsprachigen Begriffs *Informatics* durch Ernst:

»Der Begriff *Informatics* ist [im Englischen] ebenfalls geläufig, wird aber gewöhnlich etwas umfassender verwendet, beispielsweise auch für die Informationsverarbeitung in biologischen oder sozialen Systemen« (Ernst, 2015, S. 1).

Wir schlagen daher die folgende allgemeine Definition vor:

Definition Informatik	
<p style="text-align: center;">Informatik ist die Wissenschaft von der systematischen und automatisierten Verarbeitung von Information.</p> <p style="text-align: right;">(Fricke u. a., 2016)</p>	

Doch was ist eigentlich Information? Information ist ein schwer zu fassender Begriff, der in verschiedenen Kontexten unterschiedlich konnotiert wird. Für unsere Zwecke verwenden wir die folgende Zuordnung:

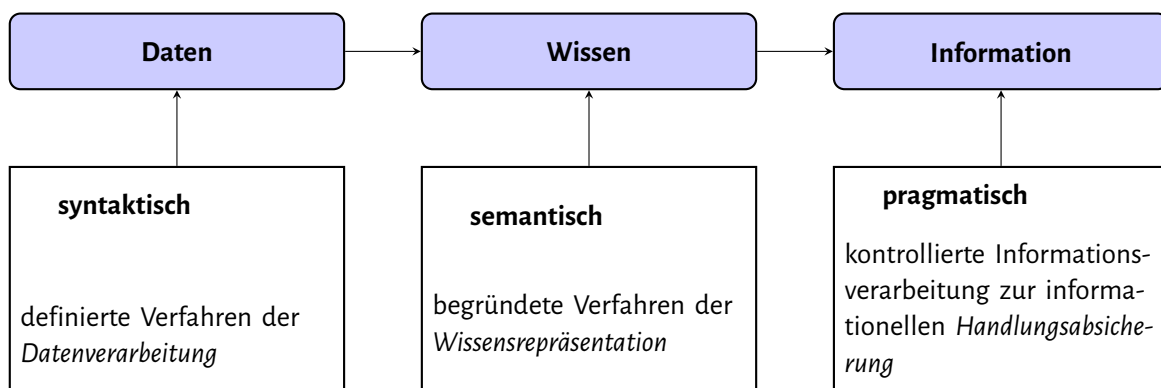


Abbildung 1: Information, Abbildung Pieper und Müller, 2013 (vgl. Fuhr, 2000).

## 2.2 Die Phänomenbereiche der Informatik

Die Informatik umgibt den Menschen. In manchen Bereichen unserer Lebenswelt ist uns dies bewusst, denn dort arbeiten wir an oder mit Computern oder anderen Informatiksystemen. An anderer Stelle vermuten wir Informatik im Hintergrund, z. B. an der Kasse im Supermarkt oder beim Aufzug. In wiederum anderen Kontexten sind wir völlig verblüfft, dass sich auch hier Informatik versteckt. So etwa beim Schriftspracherwerb als Form von Codierung oder beim *Hüpfekästchen*-Spielen auf dem Pausenhof als Anwendung von Algorithmen.

Humbert und Puhmann haben 2004 in einem gedanklichen Modell drei Phänomenbereiche der Informatik skizziert:

Phänomenbereiche der Informatik
<ol style="list-style-type: none"> <li>1. Informatikphänomene im direkten Zusammenhang mit Informatiksystemen,</li> <li>2. Informatikphänomene mit indirektem Bezug zu Informatiksystemen und</li> <li>3. Informatikphänomene ohne jede Beteiligung von Informatiksystemen.</li> </ol> <p style="text-align: right; margin-top: 20px;">(Humbert und Puhmann, 2004)</p>

Die hier vorgestellten Materialien lassen sich verschiedenen Phänomenbereichen zuordnen. Das Unterrichtsbeispiel zur Pflanzenbewässerung lässt sich z. B. je nach Betrachtungsweise allen drei Bereichen zuordnen.

- Wird der *Calliope mini* als Feuchtigkeitssensor eingesetzt, so ist dies dem Phänomenbereich 2 zuzuordnen.
- Wenn man nun den *Calliope mini* in den Vordergrund rückt und es um die Programmierung der Bewässerungsanlage geht, ist es dem Phänomenbereich 1 zuzuordnen.
- Wohingegen man den *Calliope mini* auch weglassen kann und den Algorithmus von Hand durchführt, indem man die Feuchtigkeit z. B. durch Berühren der Erde bestimmt. In diesem Fall ist das Unterrichtsbeispiel dem Phänomenbereich 3 zuzuordnen.

## 2.3 Die Fachgebiete der Informatik

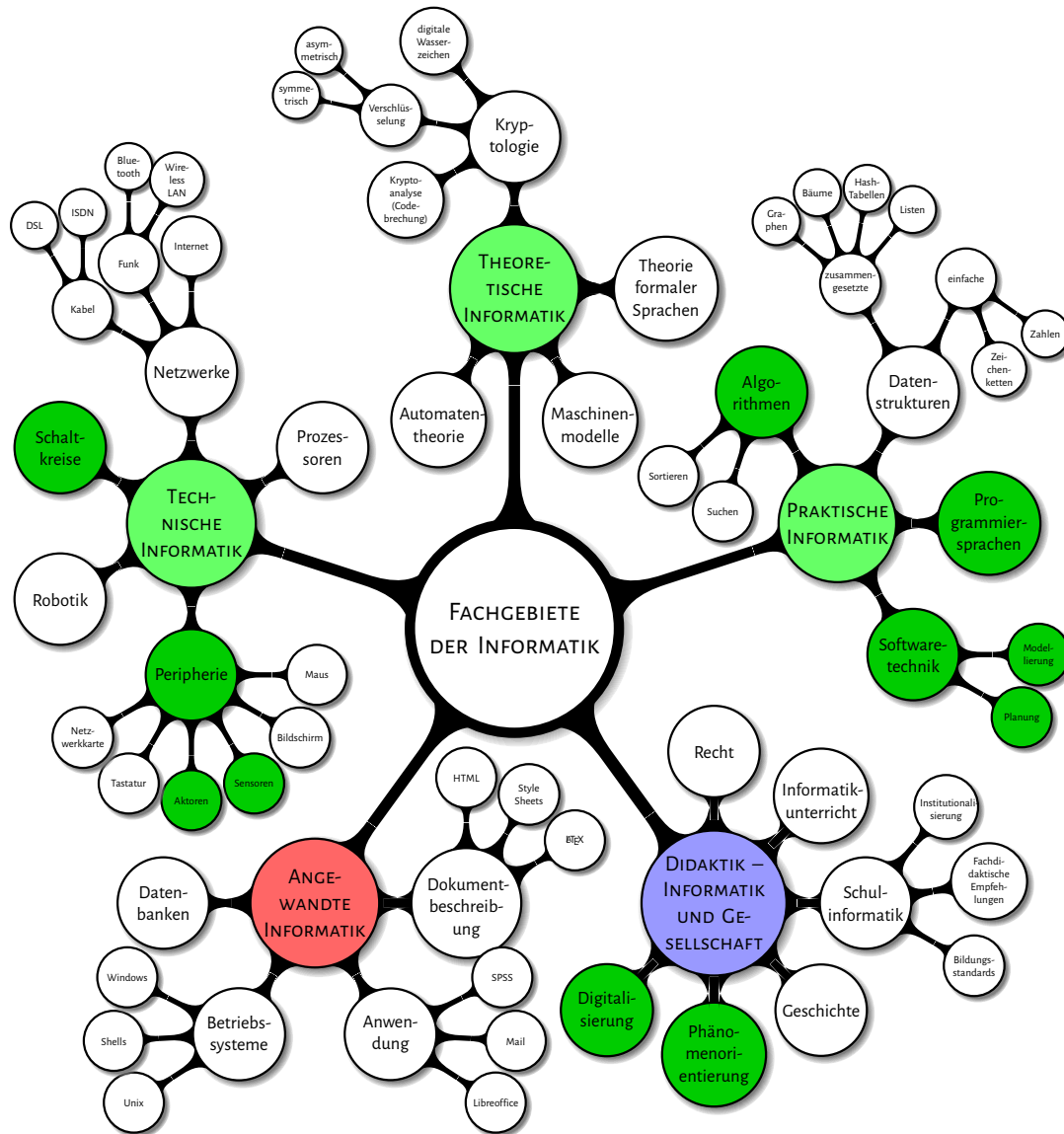


Abbildung 2: Die Fachgebiete der Informatik, Anpassung durch Haase (vgl. Humbert, 2016).

Die Informatik ist in sechs Fachgebiete unterteilt. Die *Technische Informatik*, die *Theoretische Informatik* und die *Praktische Informatik* werden zur sogenannten *Kerninformatik* (in Abb. 2 hellgrün) zusammengefasst. Dem gegenüber stehen die *Angewandte Informatik* sowie *Informatik und Gesellschaft* und die *Didaktik der Informatik*, letztere sind in Abb. 2 zu einem Oberpunkt zusammengefasst. *Calliope mini* lässt sich entsprechend Abb. 2 verschiedenen Bereichen aus *Technischer* und *Praktischer Informatik* zuordnen. Die Zuordnungen der Inhalte des vorliegenden Moduls sind *dunkelgrün* hervorgehoben.

Abb. 2 ist als Übersicht zu verstehen. Sie lässt sich erweitern und ausdifferenzieren.



## 2.4 Der informatische Modellierungskreislauf

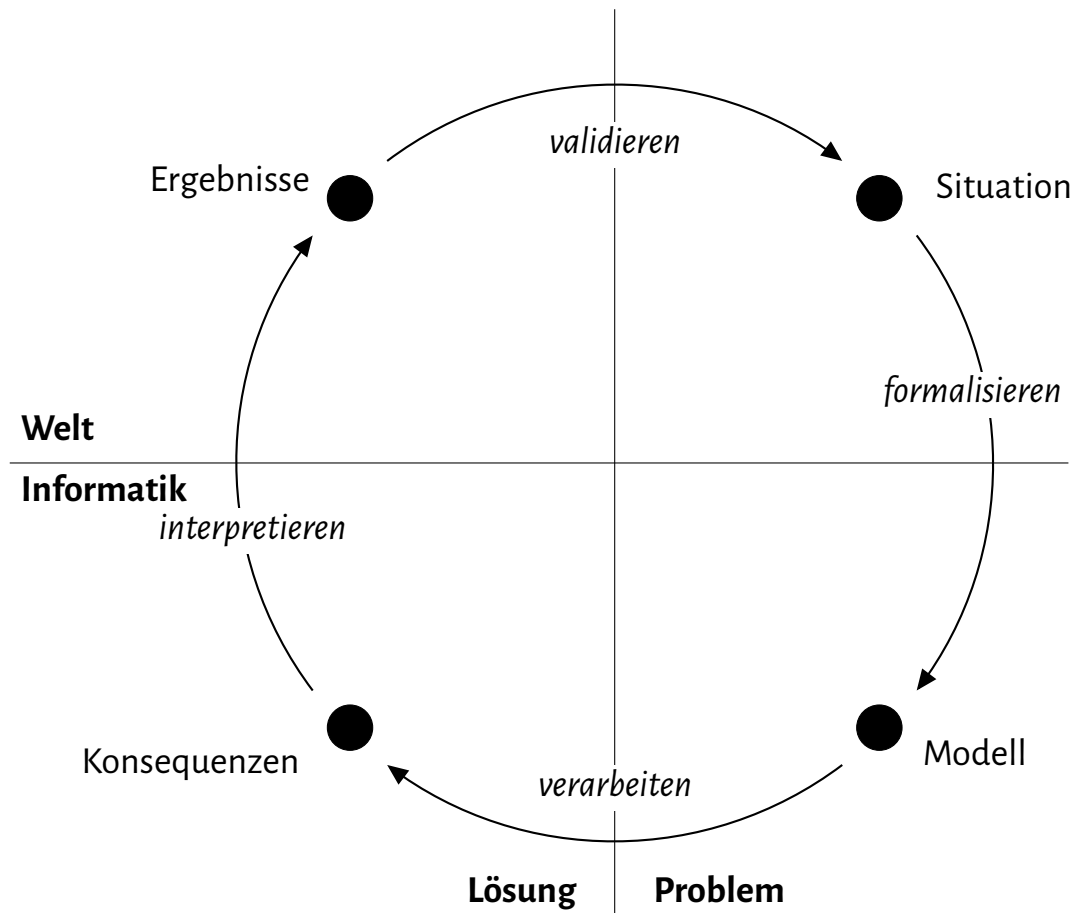


Abbildung 3: Modellierungskreis der Informatik (Humbert, 2006, S. 14).

Im informatischen Modellierungskreis (vgl. Abb. 3) lernen die Schülerinnen und Schüler, wie eine Problemstellung der Alltagswirklichkeit aus Sicht der Informatik bearbeitet wird.

Der Kreislauf unterscheidet horizontal die Ebenen *Welt* und *Informatik*, sowie vertikal die Ebenen *Problem* und *Lösung*. Hieraus ergeben sich die vier Kreissektoren mit den Zuständen *Situation*, *Modell*, *Konsequenzen* und *Ergebnisse*.

Jeder Übergang von einem Zustand zum nächsten entspricht einem festgelegten Arbeitsschritt.

(Vgl. Fricke u. a., 2016)

## Beispielhafte Anwendung des Modellierungskreislaufs

### 1. Situation

Der Zustand *Situation* stellt in der wirklichen Welt eine Problemstellung dar.

**Eine Pflanze muss regelmäßig bewässert werden. Dies soll in Zukunft automatisiert geschehen.**

### 2. Modell

Im Übergang zum nächsten Zustand wird das Problem *formalisiert*. Somit entsteht ein *Modell*. Ein Modell ist als gestaltetes Abbild der Realität zu verstehen und umfasst die zur Problemlösung notwendigen Elemente.

**Um eine Bewässerungsanlage bauen zu können, muss zunächst überlegt werden, wann eine Pflanze Wasser benötigt und wann nicht. Die Modellierung ergibt, wenn die Erde keinen Strom leitet, ist sie trocken und die Pflanze benötigt Wasser, wenn die Erde hingegen feucht ist, leitet sie den Strom und die Pflanze benötigt kein Wasser.**

### 3. Konsequenzen

Das so entstandene Modell kann *verarbeitet* (oder *abgearbeitet*) werden. Aus der Verarbeitung ergeben sich Konsequenzen.

**Es wird ein Informatiksystem entwickelt, das die Leitfähigkeit der Erde überprüft und diese ggf. bewässert.**

### 4. Ergebnisse

Durch *Interpretieren* der Konsequenzen gelangt man zurück in horizontale Ebene der *Welt*. Die *Ergebnisse* stellen eine *Lösung* in der Lebenswirklichkeit dar.




**Es stellt sich heraus, dass es eine Verzögerung gibt, bis die Leitfähigkeit hergestellt ist und in der Zeit der Topf mit Wasser schon übergelaufen ist. Eine bessere Methode muss her. Der Kreislauf beginnt von Neuem...**

## 2.5 Zu erwerbende Kompetenzen

Die in dieser Handreichung vorgelegten Beispielszenarien liefern Vorlagen für die kreative Gestaltung weiterer Herausforderungen zum spielerischen Erwerb von Kompetenzen, die an den Interessen von Schülerinnen und Schülern orientiert werden. Sie unterstützen die Entwicklung der folgenden Kompetenzen in Anlehnung an *DIG-COMP: A Framework for Developing and Understanding Digital Competence in Europe* (Ferrari, 2013, S. 15, 19, 25, 28, 32).



### Problemlösendes Lernen

Die Schülerinnen und Schüler ...

-  messen innerhalb eines Phänomens bzw. Experiments Daten, indem sie die Sensoren eines Informatiksystems geeignet nutzen.
-  prüfen formulierte Hypothesen, indem sie gemessene Daten unter Verwendung von geeigneten Ausgaben eines Informatiksystems sinnvoll darstellen und interpretieren.
-  formulieren und diskutieren Fragestellungen zu technischen Phänomenen unter Berücksichtigung der zugrundeliegenden Modellierung, indem sie Sensoren und Aktoren von Informatiksystemen zur Entdeckung und zur Prüfung nutzen.



### Modellieren & Implementieren

Die Schülerinnen und Schüler ...

-  modellieren für einfache Problemstellungen Algorithmen, indem sie Abläufe formulieren und darstellen.
-  implementieren/erweitern Programme für/auf/mit gegebene/n Informatiksysteme/n, um deren Funktionsumfang zu erweitern.





### Phänomenorientierung

Die Schülerinnen und Schüler ...

-  untersuchen Auswirkungen von Informatiksystemen an beispielhaften informatischen Phänomenen für ihren Alltag.
-  benennen Nutzen und Gefahren informatischer Phänomene.

### Problemlösen & Gestalten

Die Schülerinnen und Schüler ...

-  beschreiben Modellierungen gegebener Informatiksysteme, indem sie informatische Problemstellungen und Phänomene entdecken und untersuchen.
-  erklären technische Phänomene, indem sie Informatiksysteme zum Entdecken und Lösen von Problemstellungen einsetzen.
-  setzen Informatiksysteme kreativ ein, um technische Alltagsphänomene hypothesengeleitet zu untersuchen.
-  entwickeln Problemlösungen innerhalb von Alltagsphänomenen, indem sie ein Modell erstellen und mit Hilfe eines Informatiksystems umsetzen und testen.

Um die hier formulierten Kompetenzen erfolgreich vermitteln zu können, sind sinnstiftende und ganzheitliche Kontexte unabdingbar. Einzelne Unterrichtsbeispiele müssen daher Teil von übergreifenden Unterrichtseinheiten – z. B. durch ihre Verbindung mit Kontexten und Inhalten des Sachunterrichts – sein. Dadurch können vor allem problemlösende und gestaltende Kompetenzen erreicht werden. So werden – über das einfache und spielerische Erleben von technischen Artefakten hinaus – informatische Kompetenzen erreicht.

Außerdem entstehen sinnvolle Synergieeffekte mit natur- und gesellschaftswissenschaftlichen Fachbereichen, da dort behandelte Themen durch die Nutzung von informatischen Mitteln vertieft und erlebt werden. Bspw. wird das Phänomen der Farbmischung durch Verwendung von Informatiksystemen für Schülerinnen und Schüler nicht nur haptisch mit Pinseln, sondern auch in Bezug auf technische Geräte wie Farbdrucker erleb- und erklärbar (vgl. Sellin, 2016). Wird das Informatiksystem zur Messung bzw. Zählung von Schritten, Toren und anderen sportlichen Aktivitäten genutzt, können *Fitnessarmbänder* erklärt und gesellschaftlich diskutiert werden.

## 3 Grundlegende Sachinformation

### 3.1 Algorithmen

Das Wort Algorithmus ist vom Namen des arabischen Gelehrten al-Chwarizmi (in Umschrift des arabischen Alphabets al-Ḥwārizmī geschrieben) abgeleitet. Al-Chwarizmi lebte ungefähr im 8. Jahrhundert n. Chr. Er war ein Universalgelehrter, der sich unter anderem mit Mathematik beschäftigte.

Formal betrachtet ist ein Algorithmus eine wohldefinierte Rechenvorschrift, die eine Größe oder eine Menge von Größen als *Eingabe* verwendet und eine Größe oder eine Menge von Größen als *Ausgabe* erzeugt. Somit ist ein Algorithmus eine Folge von Rechenschritten, die die Eingabe in die Ausgabe umwandeln (vgl. Cormen u. a., 2001). Algorithmen kennt jeder: z. B. die schrittweise Installationsanleitung für einen neuen Router oder auch die Musiknoten für ein bestimmtes Lied.

Algorithmen können sehr unterschiedlich formuliert sein – es beginnt mit einfachen wörtlichen, wenig abstrakten Darstellungsformen. Um Algorithmen auf Informatiksystemen – also auch auf dem *Calliope mini* zum Ablauf zu bringen, müssen sie formalen Kriterien genügen – also syntaktisch korrekt sein. Dies wird über die Eingabemöglichkeit der Algorithmen mit Hilfe der Editoren sichergestellt.

Damit ist noch keine semantische Richtigkeit sichergestellt, dass heißt: ob der Algorithmus tut, was er soll, bleibt der Einschätzung der Schülerin oder des Schülers (und der Lehrkraft) überlassen.

Algorithmen werden im Wesentlichen aus den beiden grundlegenden Strukturen: Sequenz und Kontrolle (des Ablaufs) zusammengestellt.

#### 3.1.1 Sequenzen

Sequenzen von Aktionen (Anweisungen) sind Aktionen, die – »ohne wenn und aber« – einfach hintereinander ausgeführt werden. Sind mehrere Aktionen für einen bestimmten Kontext zusammenzufassen, so kann gesagt werden, dass diese gemeinsam einen **Block** darstellen, der ggf. an verschiedenen Stellen eines Algorithmus genutzt werden soll. In dem Fall ist es auch möglich, einem Block einen Namen – genauer: **einen Bezeichner** – zu geben.

#### Ein Beispiel für eine Sequenz:

zeige traurigen Smiley  
stelle Bewässerung an

#### 3.1.2 Kontrollstrukturen

Algorithmen zeichnet aus, dass nicht nur eine lineare Abfolge von Aktionen durchgeführt wird, sondern – in Abhängigkeit von Bedingungen – Aktionen wiederholt ausgeführt werden können aber auch die Möglichkeit besteht, während des Ablaufs zwischen verschiedenen Aktionen zu wählen. Die Konstruktion solcher Algorithmen benötigt **Kontrollstrukturen**, mit denen die **Kontrolle** über den Ablauf beschrieben wird. Insbesondere werden die beiden oben genannten Elemente als *Schleifen* beziehungsweise als *Verzweigung* bezeichnet. Diese beiden Kontrollstrukturen ermöglichen es, beliebig komplexe Abläufe zu beschreiben.

Auch solche Strukturen finden sich beispielsweise in Musiknoten wieder. Schleifen kennen wir dort als Wiederholungszeichen  $\text{||}$ , genauso wie Verzweigungen, wenn wir z. B. beim zweiten Durchlauf etwas anderes spielen sollen, als beim ersten.

**Ein Beispiel für eine Schleife:**

**solange** *der Topf nicht überläuft* **tue**  
 | füge 100 ml Wasser hinzu

**Ein Beispiel für eine Verzweigung:**

**wenn** *Erde feucht* **dann**  
 | zeige fröhlichen Smiley  
**sonst**  
 | zeige traurigen Smiley

Die beiden Strukturen können auch kombiniert werden. Das kann dann z. B. wie folgt aussehen:

**Ein Beispiel für eine Kombination aus Schleife und Verzweigung:**

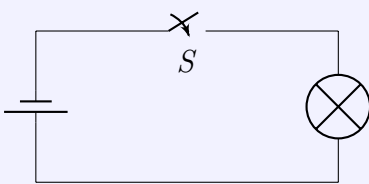
**solange** *Wasser vorhanden* **tue**  
 | **wenn** *Erde feucht* **dann**  
 | | warte 30 Minuten  
 | **sonst**  
 | | füge 100 ml Wasser hinzu

### 3.2 Technische Informatik

Die *Technische Informatik* befasst sich unter anderem mit dem Aufbau von Schaltkreisen, Computern, Mikrocontrollern und der zugehörigen Peripherie (siehe Abb. 2 S. 7).

Wir nähern uns der *Technischen Informatik* zunächst über einen einfachen Stromkreis aus Batterie, aufgeklebten Leiterbahnen, einem Schalter<sup>1</sup> und einer Glühlampe (wahlweise auch LED), die zum Leuchten gebracht wird.

**Beispiel einfacher Stromkreis**



S(chalter)	Lampe
0	0
1	1

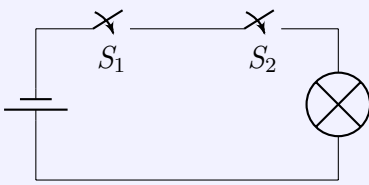
Wenn der Schalter geöffnet ist, fließt kein Strom und die Lampe kann nicht leuchten. Ist der Schalter hingegen geschlossen, leuchtet auch die Lampe.

Dieser Stromkreis wird nun nach und nach weiter ausgebaut, so dass wir im nächsten Schritt logische Schaltungen kennenlernen. So lernen wir als nächstes die Reihenschaltung mit zwei hintereinander angebrachte Schaltern kennen. Diese Schaltung entspricht einem *logischen und* (Symbol  $\wedge$ ), denn der Strom fließt nur, bzw. die Lampe leuchtet nur, wenn beide Schalter geschlossen sind.

<sup>1</sup>Wir haben uns an dieser Stelle für den Begriff Schalter und nicht dem beim *Calliope mini* verwendeten Taster entschieden, da dies der für den naturwissenschaftlichen Unterricht gebräuchliche Begriff ist. Nähere Erläuterungen folgen im Verlauf dieses Unterkapitels.



### Beispiel Reihenschaltung

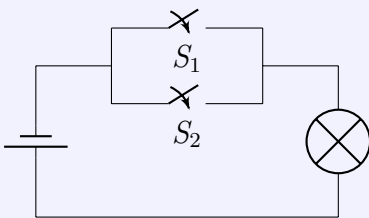


$S_1$	$S_2$	Lampe
0	0	0
0	1	0
1	0	0
1	1	1

0 bedeutet, dass kein Strom fließt, 1 bedeutet, dass Strom fließt. Man sieht, dass bei der *und*-Schaltung (Symbol  $\wedge$ ) nur Strom fließt, wenn auch durch beide Schalter,  $S_1$  und  $S_2$ , Strom fließt, diese somit geschlossen sind.

Die zweite logische Schaltung, der wir uns widmen, ist das sogenannte *logische oder* (Symbol  $\vee$ ). Dies entspricht einer Parallelschaltung. Hierzu wird eine Leiterbahn in zwei parallele Leiterbahnen aufgeteilt, auf der jeweils ein Schalter angebracht ist. Dies hat zur Folge, dass die Lampe immer dann leuchtet, wenn wenigstens ein Schalter geschlossen ist.

### Beispiel Parallelschaltung



$S_1$	$S_2$	Lampe
0	0	0
0	1	1
1	0	1
1	1	1

Nur im Fall, wenn beide Schalter,  $S_1$  und  $S_2$ , geöffnet sind, leuchtet die Lampe nicht. In allen anderen Fällen ist wenigstens ein Schalter geschlossen und die Lampe leuchtet.

Der nächste Schritt sieht dann so aus, dass ein Mikrocontroller wie *Calliope mini* all diese Dinge auf einem Chip vereint und wir die Schaltungen dort nicht mehr von Hand kleben müssen, sondern den Mikrocontroller nach unseren Vorstellungen programmieren können.<sup>2</sup>

Im Abschnitt zu den verschiedenen Schaltungen wurde bislang immer von Schaltern gesprochen. Auf dem *Calliope mini* befinden sich aber Taster. Worin besteht eigentlich der Unterschied? Schalter haben, wie in den Abbildungen dargestellt, zwei persistente Zustände: Sie sind entweder geschlossen oder geöffnet. Wenn sie geschlossen werden, bleiben sie geschlossen bis sie wieder geöffnet werden und andersherum. Im Alltag begegnen sie uns z. B. an Steckerleisten, die über einen Kippschalter zum Ein- und Ausschalten verfügen. Dem gegenüber stehen Taster: Diese sind erst einmal ständig geöffnet, d. h. der Stromkreis ist nicht geschlossen. Erst wenn man sie drückt, schließt sich der Stromkreis. Allerdings auch nur so lange, wie man sie tatsächlich drückt. Beim Loslassen öffnet sich der Taster wieder. Ein Alltagsbeispiel für die Anwendung eines Tasters ist der Klingelknopf: die Türklingel ertönt, während man den Klingelknopf drückt. Taster finden eine immer größere Verbreitung. So sind moderne »Lichtschalter« heutzutage meist keine Schalter mehr, sondern Taster, die einen Impuls geben, mit dem das Licht an- oder ausgeschaltet wird.

Auch bei einem Mikrocontroller wie dem *Calliope mini* ist es möglich, Peripherie anzuschließen. Hierzu gehören *Sensoren* und *Aktoren*. Mit einem Sensor erhalten wir ein Eingangssignal, auf das wir entsprechend reagieren können. So ist z. B. unser selbst gebauter Feuchtigkeitssensor (siehe Seite 22) ein Sensor, der uns im Falle von feuchter (leitender) Erde eine *logische 1* als Signal gibt<sup>3</sup>. Darauf können wir z. B. mit einem lachenden Smiley als Ausgabe auf dem Display reagieren. Alternativ können wir aber auch einen *Aktor* verwenden, um darauf zu reagieren. Ein Aktor könnte z. B. ein Motor sein, aber auch ein steuerbares Ventil, das bei trockener, also nicht leitender Erde unsere Pflanze entsprechend bewässert.

<sup>2</sup>Im Allgemeinen wird immer nur ein Schaltungstyp verwendet. Dies sind in der Regel NAND- oder NOR-Schaltungen, da sich aus ihnen alle anderen Schaltungen zusammensetzen lassen.

<sup>3</sup>Genau genommen ist das Signal nicht digital. Ab einer gewissen Eingangsspannung wertet der Mikrocontroller das Eingangssignal als *logische 1*. Alternativ können wir das Signal auch analog verwenden, dann ließe sich zwischen staubtrockener, ein bisschen feuchter und nasser Erde unterscheiden.

### 3.3 Codierung

Unter Codierung versteht man die Umwandlung einer Darstellungsform in eine andere. So können z. B. die Buchstaben einer Nachricht in Morsezeichen umgewandelt werden. Die so codierte Nachricht ist nicht geheim, denn der Algorithmus für die Codierung, also wie die Nachricht codiert wird, ist allgemein bekannt.

Somit handelt es sich bei jeder Codierung um eine Konvention (vgl. Strecker, 2009).

Streng genommen ist auch das gesprochene Wort »Hund« bereits eine Codierung. Denn das mentale Abbild eines Hundes wird von uns mit einem sprachlichen Code, also dem verbal gesprochenen Wort »Hund« codiert.

Das gesprochene Wort »Hund« kann dann wiederum als Schriftbild codiert werden. Somit ist der Hund bereits doppelt codiert. Wenn das Wort »Hund« dann z. B. in einen Morse-Code übersetzt wird, ist der Hund sogar dreifach codiert.

Kinder lernen daher bereits während des Spracherwerbs zu codieren. Ab dem ersten Schuljahr lernen sie eine weitere Art der Codierung: indem sie die Schriftsprache erwerben, codieren sie Laute in Buchstaben bzw. Phoneeme in Grapheme.

Auch im Alltag kommen Kinder stetig mit den unterschiedlichsten Codierungen in Kontakt. So kennen sie möglicherweise im Rahmen von Spionage-Themen in Kinderzeitschriften bereits das Morse- Alphabet oder haben im Aufzug und anderen Bereichen des öffentlichen Lebens Blindenschrift wahrgenommen.

Aber auch in ihrer unmittelbaren Umgebung kommen sie mit Codierungen in Kontakt, ohne dies bewusst zu bemerken. Sie nehmen z. B. Verkehrsschilder wahr ohne unmittelbar deren Bedeutung, also Codierung, zu kennen. Nachdem sie einzelne Schilder im Rahmen der Radfahrausbildung kennengelernt haben, kennen sie die Codierung und können sich verkehrskonform nach ihnen richten.

Auch Noten für ein Instrument sind eine Art von Codierung, mit der Kinder bereits im Grundschulalter in Kontakt kommen. Darüber hinaus gibt es in der kindlichen Lebenswirklichkeit noch Unmengen weiterer Codierungen, wie etwa QR-Codes oder Strichcodes, um nur zwei Beispiele zu nennen, die im Rahmen des Unterrichts thematisiert werden können.

(vgl. Fricke u. a., 2016)

In dem vorliegenden Modul *Codierung – Multimediales Büchlein* geht es um die binäre Codierung von Zahlen, diese wird benötigt, damit der *Calliope mini* weiß, welche Seite gerade aufgeschlagen ist.

#### 3.3.1 Das Binärsystem

Eine mögliche Codierung ist das Binärsystem. Doch warum überhaupt müssen Zahlen für Informatiksysteme kodiert werden? Informatiksysteme arbeiten intern nur mit den beiden Zustände an und aus bzw. 0 und 1. Also werden alle Zahlen zur Basis 2 dargestellt. Das ist für uns ungewohnt, stellen wir im Dezimalsystem doch alle Zahlen zur Basis 10 dar. Wenn wir uns nun anschauen, wie die Darstellung zur Basis 2 geschieht, hilft es – für das Verständnis – erst einmal zu verstehen, wie unsere gewöhnliche Darstellung zur Basis 10 funktioniert.

Schreiben wir die Zahl 768, ist für uns automatisch klar, dass es sich dabei um die Zahl Siebenhundertachtundsechzig handelt. Die letzte (dritte) Ziffer sind die Einer, die vorletzte (zweite) Ziffer die Zehner und die vorderste Ziffer die Hunderter. Man könnte also auch schreiben

$$7 \cdot 100 + 6 \cdot 10 + 8 \cdot 1 = 768_{10}$$

Wenn man dies nun in Potenzen angibt, ergibt sich

$$7 \cdot 10^2 + 6 \cdot 10^1 + 8 \cdot 10^0 = 768_{10}$$

Das Binärsystem funktioniert analog zur Basis 2. Die Zahl  $101101_2$  lässt sich genauso aufschreiben und damit zurück ins Dezimalsystem umwandeln

$$101101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 32 + 8 + 4 + 1 = 45_{10}$$



Wenn man eine Zahl vom Dezimalsystem ins Binärsystem umwandeln möchte, kann man sich entweder überlegen, welche Zweierpotenzen in der Zahl stecken, oder man verwendet einen *Algorithmus* zur Umwandlung (welcher hier aber nicht Thema sein soll).

### Exkurs

Neben dem Binär- und Dezimalsystem gibt es auch noch andere bekannte Systeme, z. B. das Hexadezimalsystem. Hier ist die Basis 16, da wir aber nur 10 Ziffern (0 bis 9) unterscheiden, werden für die restlichen benötigten Ziffern Großbuchstaben (von *A* bis *F*) verwendet. Zum Einsatz kommt das Hexadezimalsystem z. B. bei Hardwareadressen (MAC-Adressen) oder den IPv6 Adressen. Ein Vorteil beim Hexadezimalsystem ist, dass 4 Bits genau eine Ziffer im Hexadezimalsystem ergeben. Das liegt daran, dass  $2^4 = 16$  ist.

Um dem *Calliope mini* im Modul multimediales Büchlein zu zeigen, auf welcher Buchseite wir gerade sind, verwenden wir die Binärdarstellung. Dadurch, dass der *Calliope mini* 4 Anschlüsse (P0 bis P3) besitzt, sind  $2^4 = 16$  Seiten möglich. Für ein kürzeres Büchlein mit 8 Seiten werden nur 3 Bits, also die Anschlüsse P0 bis P2 benötigt. Die Kodierung kann Tabelle der folgenden Tabelle entnommen werden:

### Kodierungstabelle Binärsystem

P0	P1	P2	Zahl
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Hierbei fällt einem ein System auf. In der letzten Spalte wechseln sich 0 und 1 immer ab, in der Spalte davor folgen auf zwei Nullen immer zwei Einsen, während es in der vordersten Spalte schon 4 und 4 sind. Damit lässt sich die Tabelle einfach fortsetzen. Um ein Buch mit 16 Seiten zu basteln schreibt man vor die bisherige Tabelle eine Spalte mit acht Nullen, darunter die gleiche Tabelle nochmal mit acht führenden Einsen.



## 4 Materialien für den Unterricht

In diesem Kapitel finden Sie die konkrete Planung für die unterrichtliche Umsetzung.

Jede Unterrichtseinheit beginnt mit dem Ausweis der **Ziele und Kompetenzen** und einer Darstellung des **zeitlichen Umfangs**. In einigen Fällen sind **Lernvoraussetzung auf der methodischen Ebene** angegeben.

Darauf folgt die **empfohlene Vorgehensweise**. Eine Abweichung von der vorgeschlagenen Umsetzung und Anpassung an die Bedürfnisse der Lernsituation ist jederzeit möglich.

Zu guter Letzt gibt es einen **Überblick** über die **Kopiervorlagen** (KV) zur Unterrichtseinheit. In einzelnen Fällen muss oder kann **zusätzliches Material** miteinbezogen werden.

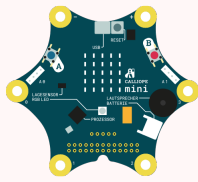
Im Anschluss daran finden Sie alle nötigen Kopiervorlagen. Die Farbgebung gibt Aufschluss über die Art des Dokuments.

- Hinweise sind **rot**.
- Arbeitsblätter und Materialien zur Auslage sind **dunkelblau**.
- Aufgaben mit methodischem Schwerpunkt sind **grün**.

## Übersicht über die Unterrichtseinheiten

Thema der Einheit	Kurzbeschreibung	Lernziele	Zeit	Seite
<b>Bewässerung</b>	Die Schülerinnen und Schüler bauen eine Bewässerungsanlage für Pflanzen.	Kennenlernen und Formulieren von Algorithmen.	1-2 Doppelstunden	S. 22
<b>Keksdose</b>	Die Schülerinnen und Schüler entwickeln eine Möglichkeit zu überprüfen, wie oft Kekse aus einer Keksdose entnommen werden.	Informatische Modellierung	2 Doppelstunden	S. 32
<b>Multimediales Büchlein</b>	Die Schülerinnen und Schüler basteln ein Büchlein, das sich mithilfe des <i>Calliope mini</i> multimedial erweitern lässt.	Kennenlernen und Verwenden von Codierung	2-3 Doppelstunden	S. 40

## Kurzanleitung *Calliope mini*



Der *Calliope mini* ist ein programmierbarer Mikrocontroller, mit dem verschiedene kreative Ideen umgesetzt werden können. Über einen entsprechenden Editor können die LED-Anzeige, die beiden Knöpfe »A« und »B«, die Anschlüsse P0, P1, P2 und P3, Lagesensoren, ein kleiner Lautsprecher, ein Helligkeitssensor, ein Funkmodul sowie weitere angeschlossene Peripherie angesprochen werden.

### Programmierung mit einem Computer:

In dieser Anleitung wird kurz erklärt, wie man (eigene) Programme auf dem *Calliope mini* zum Ablauf bringt:

1. Am Computer mithilfe eines Internetbrowsers einen *Calliope*-Editor unter <http://calliope.cc/editor> öffnen. Hierzu stehen zur Zeit drei zur Auswahl, der Editor mit dem aktuell größten Funktionsumfang ist der PXT-Editor.  
*Hinweis:* Einzelne Bausteine wie z. B. **spiele Note ...** können niemals alleine stehen, sondern müssen immer in Blöcke wie z. B. **on start**, **dauerhaft** oder **wenn Knopf ... gedrückt** eingebettet sein.
2. Das dort entwickelte Programm kann über den **Herunterladen**-Button heruntergeladen werden.
3. Abhängig vom verwendeten Betriebssystem/Browser erscheint nun entweder das bekannte Download-Kontextmenü, oder das Programm mit der Endung `.hex` wird im Download-Verzeichnis gespeichert.
4. Nun verbindet man den *Calliope mini* mittels Micro-USB Kabel mit dem Computer. (*Achtung! Man benötigt ein Micro-USB Kabel, welches auch Daten übertragen kann. Unter Umständen können die Kabel nur Strom übertragen.*)
5. Der *Calliope mini* sollte nun wie ein gewöhnliches USB-Laufwerk angezeigt werden, so dass die `.hex`-Datei auf den *Calliope mini* kopiert werden kann. Während des Kopiervorgangs blinkt die LED über dem A-Taster. (Abhängig vom Betriebssystem verschwindet der *Calliope mini* kurz und wird dann ohne die Datei erneut angezeigt. Dies ist aber kein Fehler.)
6. Durch drücken des **RESET**-Knopfes wird das Programm nun zur Ausführung gebracht.

### Hinweis:

Die USB-Verbindung wird nach der Übertragung des Programms nicht mehr benötigt. Man kann die Verbindung also trennen und den *Calliope mini* stattdessen über den Batterieanschluss mit Strom versorgen.

### Programmierung mit der *micro:bit*-App

Mithilfe der *micro:bit*-App kann der *Calliope mini* auch mit einem Smartphone programmiert werden. Dazu müssen die beiden Geräte mit den folgenden Schritten über Bluetooth gekoppelt werden:

1. *micro:bit*-App auf das Smartphone laden und Bluetooth aktivieren.
2. Nun wählt man in der *micro:bit*-App »Connections« und anschließend »Pair a new *micro:bit*«.
3. Die dann folgende Anleitung analog auf dem *Calliope mini* durchführen.



# Technische Informatik

## Schaltungen

### Unterrichtseinheit 1





## Schaltungen mit Kupferklebeband




### Ziele und Kompetenzen


Die Schülerinnen und Schüler ...

-  verstehen die Notwendigkeit von logischen Schaltungen.
-  wissen wie verschiedene logische Schaltungen aufgebaut sind und sind in der Lage, diese selbstständig anzufertigen.

### Zeit

 1 Doppelstunde


### Voraussetzungen

 Stromkreis





### Empfohlene Vorgehensweise

- ▶ Für die Informatik sind Schaltungen unerlässlich. In dieser Unterrichtseinheit geht es um den generellen Aufbau von logischen Schaltungen.
- ▶ Einfache Schaltungen, lassen sich ohne Probleme mit Kupferklebeband realisieren. Zeigen Sie den Schülerinnen und Schülern beispielhaft eine Schaltung aus einem Schalter und einer LED
- ▶ Nun sollen die Schülerinnen und Schülern UND- bzw. ODER-Schaltungen basteln. Eine Möglichkeit besteht darin, die Hälfte der Klasse UND- und die andere Hälfte ODER-Schaltungen basteln zu lassen.
- ▶ Abhängig von der Leistungsstärke können den Schülerinnen und Schülern Hilfen, wie Tabelle oder Schaltplan an die Hand gegeben werden.

### Material-überblick

 Arbeitsblatt

### Zusätzliches Material

-  Kupferklebeband (beidseitig leitend)
-  Knopfzelle (3 Volt)
-  LEDs
-  Pappe

Basierend auf einer Idee von Noller: *Drawing Circuits: Bare-conductive vs. Circuit Scribe vs. Chibitronics.*

## Schaltungen mit Kupferklebeband



### Benötigtes Material:

- Kupferklebeband,
- LEDs,
- Knopfzellen (3 Volt),
- Pappe.

### Einfacher Stromkreis:

- Auf dem Foto rechts ist eine Schaltung mit einem Schalter abgebildet. Ist dieser geschlossen, leuchtet die LED, ist er geöffnet ist die LED aus.
- Wichtig ist es die Einbaurichtung der LED (langes Füßchen +, kurzes Füßchen -) zu beachten.
- Ebenso solltest du Kurzschlüsse vermeiden.



### Aufgabe:

- Bastele eine UND- und eine ODER-Schaltung. Weiter unten sind die entsprechenden Tabellen abgebildet.

#### UND-Schaltung

$S_1$	$S_2$	LED
0	0	0
0	1	0
1	0	0
1	1	1

#### ODER-Schaltung

$S_1$	$S_2$	LED
0	0	0
0	1	1
1	0	1
1	1	1

# Algorithmen

## Pflanzenbewässerung

### Unterrichtseinheit 2







## Pflanzenbewässerung




### Ziele und Kompetenzen




Die Schülerinnen und Schüler ...

-  kennen die informatischen Kontrollstrukturen *solange*-Schleife und *wenn-dann-sonst*-Verzweigung.
-  identifizieren informatische Kontrollstrukturen in alltäglichen Situationen.
-  formulieren einen Algorithmus zur Bewässerung einer Pflanze.
-  realisieren mithilfe des *Calliope mini* das Programm, das den Algorithmus zur Bewässerung implementiert.

### Zeit

 1-2 Doppelstunden

### Lernvoraussetzungen

-  einfacher Stromkreis
-  Leitfähigkeit verschiedener Materialien
-  Think-Pair-Share

### Empfohlene Vorgehensweise

- ▶ Wiederholen Sie zu Beginn das Thema Leitfähigkeit, doch anstatt des einfachen Stromkreises wird dieses Mal der *Calliope mini* zur Quantifizierung der Leitfähigkeit benutzt. Wenn das Material leitet, erscheint ein lachender Smiley, wenn nicht ein trauriger. Die Schülerinnen und Schüler können dies in Gruppen ausprobieren.
- ▶ Stellen Sie den Schülerinnen und Schülern folgende Situation vor:
 

Im Klassenraum gibt es viele Pflanzen. Nun stehen die Ferien an und viele der schönen Pflanzen würden vertrocknen. Eine Lösung bestünde darin, dass jemand regelmäßig zum Gießen vorbeikommt. Wie kann denjenigen technisch unterstützen, so dass er eventuell auch mal einen Tag zuhause bleiben kann?
- ▶ Lassen Sie Schülerinnen und Schüler in einem Think-Pair-Share überlegen, woran man erkennen kann, dass eine Pflanze genügend Wasser hat, bzw. wie man das mit dem *Calliope mini* feststellen kann.
- ▶ Lassen Sie die Schülerinnen und Schüler dies in Gruppen mithilfe des bereits verwendeten Programms zum Test auf Leitfähigkeit und mit Hilfe von Topfpflanzen und Gießkannen ausprobieren.
- ▶ Besprechen Sie informatische Kontrollstrukturen wie Schleife und Verzweigung anhand von Alltagsbeispielen.
- ▶ Lassen Sie die Schülerinnen und Schüler einen Algorithmus modellieren, der sich um die Bewässerung der Pflanze kümmert, d. h. in regelmäßigen Abständen die Feuchtigkeit prüft und dann entweder durch ein akustisches Signal darauf aufmerksam macht, dass die Pflanze bewässert werden sollte, oder dies durch die Ansteuerung eines Ventils übernimmt.



## Pflanzenbewässerung










- ▶ Stellen Sie den Schülerinnen und Schülern das auf dem *Calliope mini* verwendete Programm vor.







*optional:*

- ▶ Wenn genügend Computer oder Smartphones zur Verfügung stehen, können die Schülerinnen und Schüler das Programm mithilfe des Editors (*siehe Kurzanleitung* – <http://uni-w.de/a6>) umsetzen.



### Material- überblick

-  KV Leitfähigkeit mit dem *Calliope mini*
-  KV Think-Pair-Share
-  KV Informatische Kontrollstrukturen
-  KV Bewässerung von Pflanzen – Programm
-  Mögliche Lösung »Bewässerungsanlage«
-  Programm für den *Calliope mini* – Leitfähigkeit
-  Programm für den *Calliope mini* – Beispiellösung Pflanzenbewässerung

### Zusätzliches Material

-  Je einen *Calliope mini* mit Batterie, auf dem bereits das Programm läuft,
-  je 2 Krokodilklemmen,
-  um nicht die Krokodilklemmen in die Erde stecken zu müssen z. B. je 2 Nägel/Schrauben,
-  verschiedene leitfähige bzw. nicht leitfähige Materialien (analog zur Unterrichtseinheit einfacher Stromkreis),
-  Blumentöpfe mit trockener Erde und Pflanze und
-  Gießkannen.

*optional:*

-  Je ein steuerbares Ventil mit Wasseranschluss oder Reservoir und
-  Computer und Micro-USB Kabel, oder Smartphones.

### Sonstige Hinweise

- ▶ Möglich ist auch statt einem lachenden Smiley für leitet und traurigem Smiley für leitet nicht, den PIN analog statt digital auszulesen. Dann kann man sich auf dem Display ausgeben lassen *wie gut* bestimmte Materialien den Strom leiten.
- ▶ Diese Variante ist auch für eine Feinjustierung der Bewässerungsanlage interessant, denn wann sollte eine Pflanze bewässert werden? Wenn die Erde schon staubtrocken oder noch ein bisschen feucht ist?



## Leitfähigkeit mit dem Calliope mini

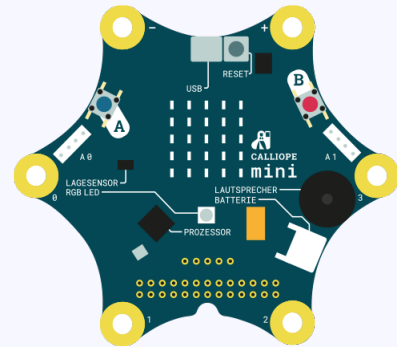


### Benötigtes Material:

- Calliope mini mit Batterien,
- 2 Kabel mit Krokodilklemmen,
- verschiedene Materialien, bei denen die Leitfähigkeit geprüft werden soll.

### Versuchsaufbau:

- Schließe den Calliope mini an die Batterie an,
- Verbinde eine Krokodilklemme mit dem »-«-Anschluss und die andere Klemme mit dem »0«-Anschluss.
- Wenn du nun den Calliope mini einschaltest, kannst du die verschiedenen Materialien auf ihre Leitfähigkeit prüfen, indem du sie mit den beiden anderen Enden der Krokodilklemmen verbindest.



### Aufgabe:

Halte die Leitfähigkeit der verschiedenen Materialien in der Tabelle fest.

Material:	leitet?		Material:	leitet?	
	ja	nein		ja	nein
_____			_____		
_____			_____		
_____			_____		
_____			_____		
_____			_____		

Wenn du fertig bist, kannst du Gruppen aus den verschiedenen Materialien bilden.



# Ich – Du – Wir – Gespräch

Methode Think–Pair–Share

## Aufgabe



Woran kannst du erkennen, dass eine Pflanze genügend Wasser hat?  
Kannst du das auch mit dem *Calliope mini* testen?

### 1. Schritt

Arbeite alleine.

Schreibe deine Ideen auf.

### 2. Schritt

Tausche dich mit deinem Sitznachbarn aus. Einigt euch auf **eine** gute Idee.

Schreibt eure gemeinsame Idee auf.

### 3. Schritt

Stellt eure Idee der Klasse vor.

Welche anderen guten Ideen hatten deine Klassenkameraden?

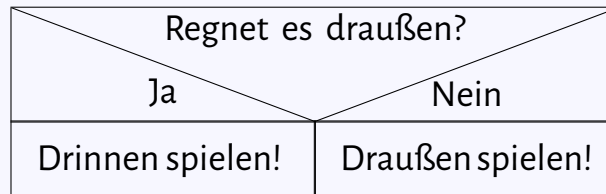
Schreibe sie auch auf.

## Informatische Kontrollstrukturen I



### Kontrollstruktur Verzweigung:

Damit der *Calliope mini* einen traurigen oder einen fröhlichen Smiley zeigen kann, hat er die Möglichkeit, Entscheidungen zu treffen. So etwas kennen wir auch aus unserem Alltag:



### Aufgabe:

Denke dir ein eigenes Beispiel aus:

Ja	Nein

### Aufgabe:

Tausche dich mit deinem Sitznachbarn aus und denkt euch gemeinsam ein weiteres Beispiel aus:

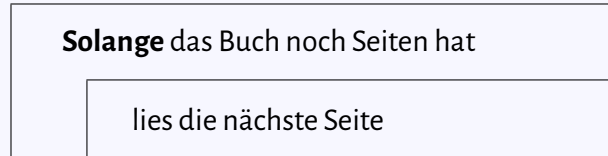
Ja	Nein

## Informatische Kontrollstrukturen II



### Kontrollstruktur Schleife:

In manchen Fällen möchte man eine Aktion immer wieder durchführen, solange eine vorgegebene Bedingung erfüllt ist.



### Aufgabe:

Denke dir ein eigenes Beispiel aus:

Solange

### Aufgabe:

Tausche dich mit deinem Sitznachbarn aus und denkt euch gemeinsam ein weiteres Beispiel aus:


Solange

## Pflanzenbewässerung



Bislang zeigt der *Calliope mini* einen traurigen Smiley ☹, wenn das Material nicht leitet und einen fröhlichen Smiley ☺, wenn das Material den Strom leitet. Unser Ziel ist es aber, dass sich der *Calliope mini* um die Bewässerung der Pflanze kümmert.

### Aufgabe:

 Versuche – mithilfe der kennengelernten Kontrollstrukturen – einen Algorithmus zu formulieren, der sich um die Bewässerung einer Topfpflanze kümmert:

## Bewässerung von Pflanzen – Programm



**Was passiert überhaupt?** Der Calliope stellt fest, ob die Erde feucht oder trocken ist und zeigt den entsprechenden Smiley.

```
wenn Erde feucht dann
  | zeige fröhlichen Smiley
sonst
  | zeige traurigen Smiley
```

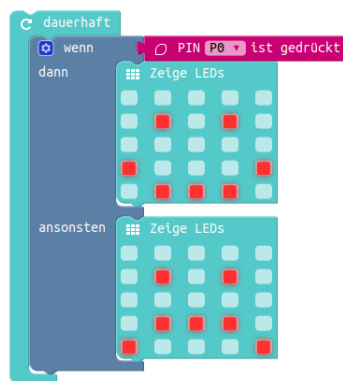
Dies erkennt der *Calliope mini* daran, ob der Stromkreis zwischen dem »-«-Anschluss und dem »0«-Anschluss geschlossen ist oder nicht.

```
wenn Eingangssignal an PIN-0 dann
  | zeige fröhlichen Smiley
sonst
  | zeige traurigen Smiley
```

Allerdings wird dies nicht nur einmal geprüft, sondern immer wieder. Deshalb läuft das Ganze in einer Schleife.

```
solange der Calliope mini eingeschaltet ist tue
  | wenn Erde feucht dann
  |   | zeige fröhlichen Smiley
  | sonst
  |   | zeige traurigen Smiley
```

Das Programm für den *Calliope mini* sieht dann folgendermaßen aus:



## Mögliche Lösung »Bewässerungsanlage«



Eine Variante ist es, den bereits bestehenden Algorithmus zu erweitern:

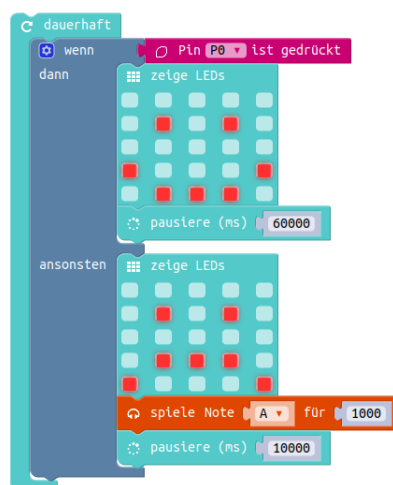
```
solange der Calliope mini eingeschaltet ist tue
  wenn Erde feucht dann
    zeige fröhlichen Smiley
  sonst
    zeige traurigen Smiley
```

Wir erweitern diesen Algorithmus einfach um eine Tonausgabe für den Fall, dass die Erde trocken ist. Alternativ könnte man hier auch das Signal geben, das Ventil der automatischen Wasserzufuhr z. B. für 2 Sekunden zu öffnen oder soundso viel Wasser hinzuzufügen.

Da eine permanente Überprüfung nicht notwendig ist, können wir darüber hinaus den *Calliope mini* für eine gewisse Zeit schlafen legen. Das hat den Vorteil, dass das Wasser auch die Chance hat, sich zu verteilen.

```
solange der Calliope mini eingeschaltet ist tue
  wenn Erde feucht dann
    zeige fröhlichen Smiley
    pausiere für eine Minute
  sonst
    zeige traurigen Smiley
    spiele einen Ton ab
    pausiere für 10 Sekunden
```

Wenn wir diesen Ansatz jetzt auf den *Calliope mini* übertragen, würde unser Programm folgendermaßen aussehen:





# Algorithmen

## Keksdose

### Unterrichtseinheit 3






## Keksdose




### Ziele und Kompetenzen




Die Schülerinnen und Schüler ...

-  erkennen informatische Phänomene in ihrer Alltagswelt.
-  können die informatische Modellierung auf Problemstellungen aus ihrer Lebenswelt anwenden.
-  kooperieren bei der Bearbeitung informatischer Probleme und tauschen sich in Fachsprache über informatische Inhalte aus

### Zeit

 2 Doppelstunden

### Lernvoraussetzungen

-  Informatischer Modellierungskreislauf
-  Informatische Kontrollstrukturen
-  Erste Erfahrung mit der Programmierung des *Calliope mini*



### Empfohlene Vorgehensweise

- ▶ Stellen Sie folgende Problemsituation vor:  
Im Klassenraum gibt es eine Dose mit Keksen, die sich Schülerinnen und Schüler zur Belohnung nehmen dürfen. Jedoch werden auch sonst Kekse entnommen. Wir wollen herausfinden, wie oft die Dose heimlich geöffnet wurde.
- ▶ Überlegen Sie gemeinsam wie die Keksdose mithilfe des *Calliope mini* überwacht werden kann.
- ▶ Lassen Sie die Schülerinnen und Schüler in Gruppen den informatischen Modellierungskreislauf durchführen
- ▶ Lassen Sie zwischendurch die Gruppen ihren jeweiligen Projektstand im Plenum referieren.

### Materialüberblick

 KV Arbeitsblatt

### Zusätzliches Material

-  Keksdosen (ggf. mit Scharnier, siehe auch *Sonstige Hinweise*).
-  weitere Materialien entsprechend den Modellierungen der Schülerinnen und Schüler.

### Sonstige Hinweise

- ▶ Mögliche Lösungen können über den Lagesensor eines auf dem Deckel der Dose montierten *Calliope mini* erfolgen.
- ▶ Hierfür benötigt man Dosen mit Scharnier.
- ▶ Ein alternativer Ansatz wäre z. B. durch das Unterbrechen eines Stromkreises beim öffnen der Dose möglich.
- ▶ ggf. müssen für verschiedene Lösungsmöglichkeiten Materialien beschafft werden.

Basierend auf einer Idee von Greve und Steingans: »Elemente der Technischen Informatik – ab der Grundschule«.





## Keksdose

Gruppenarbeit

### Aufgabe



Überlegt euch, wie ihr mithilfe des *Calliope mini* die Keksdose überwachen könnt.

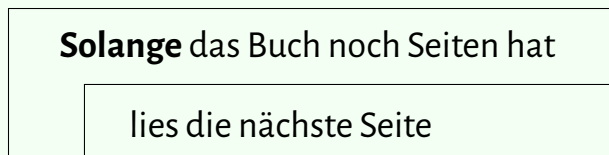
### Checkliste

- Sammelt zunächst Ideen, wie ihr die Keksdose überwachen wollt.
- Einigt euch auf eine Idee.
- Überlegt euch einen Algorithmus zu eurer Idee
- Welche Materialien benötigt ihr für die Umsetzung?
- Setzt den Algorithmus mit dem *Calliope mini* um.
- Testet eure Keksdosensicherung. Gibt es vielleicht noch Optimierungsmöglichkeiten?

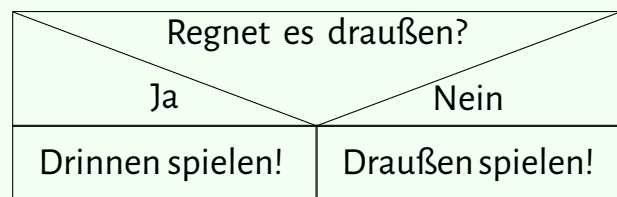
### Hinweise:

Erinnert euch an die Kontrollstrukturen:

#### Schleife



#### Verzweigung



# Algorithmen

## Quizbuzzer

# Modul 4






## Quizbuzzer




### Ziele und Kompetenzen



Die Schülerinnen und Schüler ...

-  verwenden die Anschlüsse des *Calliope mini*, um mehrere Eingangssignale zu lesen.
-  wenden ihr Wissen zur technischen Informatik an, um Peripherie in Form von Quizbuzzern zu bauen und an den *Calliope mini* anzuschließen.
-  verwenden informatische Kontrollstrukturen, um verschiedene Eingaben ihrer Modellierung entsprechend zu verarbeiten.

### Zeit

 1–2 Stunden

### Lernvoraussetzungen

-  Basiswissen zur Technischen Informatik
-  Grundlegende Fertigkeiten im Umgang mit der Programmieroberfläche des *Calliope mini*




### Empfohlene Vorgehensweise

- ▶ Problematisieren Sie die Situation:
  - Wie lässt sich bei einem Quiz eindeutig feststellen, wer die Antwort am schnellsten weiß?
  - Es gibt verschiedene analoge Möglichkeiten, wie den eigenen Namen rein zu rufen, auf eine Klingel zu hauen etc.
  - Mit einem Informatiksystem lässt sich diese Frage jedoch viel einfacher beantworten.
- ▶ Lassen Sie die Schülerinnen und Schüler zunächst überlegen wie ein Algorithmus unabhängig vom *Calliope mini* aussehen muss, der die Problematik löst.
- ▶ Versuchen Sie nun gemeinsam mit den Schülerinnen und Schülern die Ergebnisse auf den *Calliope mini* zu übertragen.
- ▶ Wie muss der *Calliope mini* verkabelt werden? Wie sieht ein mögliches Programm aus.
- ▶ Abhängig von den Fähigkeiten stehen sowohl eine Bastelanleitung, als auch ein fertiges Programm zur Verfügung.





## Quizbuzzer



### Material- überblick

-  Arbeitsblatt,
-  Hilfe: Bastelanleitung,
-  Hilfe: Programm für den *Calliope mini* – Quizbuzzer

### Zusätzliches Material

-  Einen *Calliope mini* mit Batterie
-  8 Kabel mit Krokodilklemmen,
-  4 Drucktaster,
-  Computer und Micro-USB Kabel, oder Smartphones zur Programmierung

### Sonstige Hinweise

- ▶ Um die Quizbuzzer praktisch einsetzen zu können, kann es sinnvoll sein, die Taster in kleine Holzboxen o. Ä. zu verbauen.

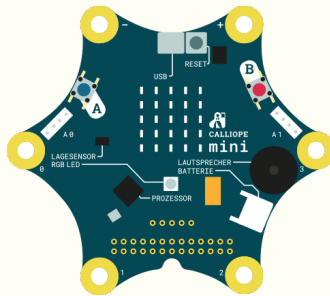
## Quizbuzzer



Wie lässt sich bei einem Quiz eindeutig feststellen, wer die Antwort am schnellsten weiß?

### Aufgabe:

- Überlege dir einen Aufbau, wie Quizbuzzer und *Calliope mini* verkabelt sein müssen, damit der *Calliope mini* die verschiedenen Buzzer unterscheiden kann. Zeichne deinen Aufbau hier ein:



**Hinweis:** Eine mögliche Lösung bietet die Bastelanleitung an.

### Aufgabe:

- Überlege dir einen möglichen Algorithmus, der die Signale der Anschlüsse liest, anzeigt welcher Buzzer zuerst gedrückt wurde und erst nach dem Reset z. B. nach Drücken der A-Taste, die Buzzer wieder freigibt.

## Bastelanleitung – Quizbuzzer

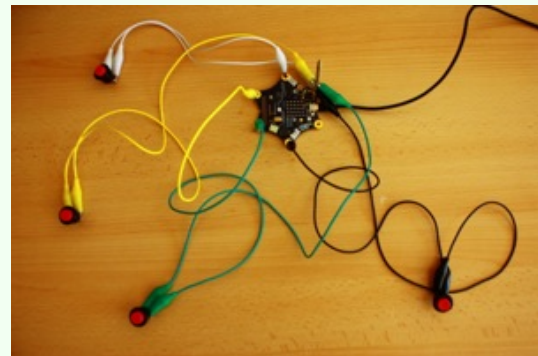


### Benötigtes Material:

- *Calliope mini* mit Batterien,
- 8 Kabel mit Krokodilklemmen,
- 4 Drucktaster,
- Schraube mit Mutter o. ä. um 4 Krokodilklemmen am »–«-Pol anschließen zu können.

### Quizbuzzer:

- ▶ Verbinde den ersten Anschluss der Drucktaster mithilfe der Krokodilklemmen mit den Anschlüssen *P0* bis *P3* des *Calliope mini*.
- ▶ Befestige am Minuspol des *Calliope mini* eine Schraube mit Mutter, damit es möglich wird alle Drucktaster mit dem Minuspol zu verbinden.
- ▶ Verbinde nun den zweiten Anschluss aller Drucktaster mit dem Minuspol des *Calliope mini* bzw. der dort angebrachten Schraube.
- ▶ Nun kann der *Calliope mini* entsprechend programmiert werden.



### Hinweis:

Für den praktischen Einsatz der Quizbuzzer kann es sinnvoll sein, die einzelnen Buzzer in kleinen Boxen unterzubringen oder von vornherein entsprechende Taster zu besorgen.



# Codierung

## Multimediales Büchlein

### Unterrichtseinheit 5






## Multimediales Büchlein




### Ziele und Kompetenzen



Die Schülerinnen und Schüler ...

-  kennen den Fachbegriff »Codierung« und können eine Codierung beispielhaft mithilfe des Binärsystems umsetzen.
-  verwenden die *UND*-Verküpfung und Verzweigungen, um verschiedene Eingabewerte zu unterscheiden.
-  setzen sich kreativ mit den Möglichkeiten des *Calliope mini* auseinander, um ihr Büchlein multimedial zu gestalten.

### Zeit

 2-3 Doppelstunden

### Lernvoraussetzungen

-  einfacher Stromkreis
-  erste Erfahrungen mit der grafischen Programmieroberfläche des *Calliope mini*





### Empfohlene Vorgehensweise

- ▶ Idee vom Multimedialen Büchlein vorstellen
- ▶ Lassen Sie die Schülerinnen und Schüler Geschichten erfinden, die sie in ihrem Büchlein erzählen wollen.
- ▶ Lassen Sie die Schülerinnen und Schüler in Partnerarbeit überlegen, wie sie die einzelnen Seiten ihrer Geschichte auf dem *Calliope mini* umsetzen wollen.
- ▶ Problematisieren Sie mit den Schülerinnen und Schülern: »Woher weiß *Calliope* auf welcher Seite wir uns befinden?«
- ▶ Thematisieren Sie mit den Schülerinnen und Schülern *Codierungen*. Hierfür eignen sich besonders gut die Materialien zum Thema *Codierung* aus dem Projekt »Informatik an Grundschulen« (siehe Fricke u. a., 2016).
- ▶ Thematisieren Sie mit den Schülerinnen und Schülern als eine mögliche Codierung für die Seitenzahlen das Binärsystem.
- ▶ Basteln Sie mit den Schülerinnen und Schülern das Büchlein und lassen Sie von den Schülerinnen und Schülern gestalten.





## Multimediales Büchlein



### Material- überblick

-  KV Arbeitsblatt Binärsystem
-  KV Bastelanleitung
-  Rahmenprogramm *Calliope mini* – Büchlein
-  Mögliches Programm für den *Calliope mini* – Büchlein

### Zusätzliches Material

-  4 Krokodilklemmen pro Büchlein
-  Pappkarton aus denen die Büchlein gebastelt werden
-  Kupferklebeband (schmal)
-  Tacker mit mind. 10 cm Reichweite, oder Nadel und Faden (um die Bücher zu »binden«)

Basierend auf einer Idee von Tech Will Save Us: *Animated micro:book*.

## Das Binärsystem



### Vom Binärsystem ins Dezimalsystem:

Möchtest du die Zahl 1101 aus dem Binärsystem in das Dezimalsystem umrechnen geht das z. B. mithilfe einer Stellenwerttafel. Mit Hilfe der Stellenwerttafel kannst du die Zahlen aus dem Binärsystem einfach ins Dezimalsystem umrechnen:

16	8	4	2	1
0	1	1	0	1

Jetzt brauchst du nur noch die Werte, bei denen in der unteren Spalte eine 1 steht zu addieren und erhältst  $8 + 4 + 1 = 13$ . Damit hast du 1101 ins Dezimalsystem umgerechnet.

### Aufgabe:

 Führe das obige Verfahren mit diesen Zahlen durch:

	16	8	4	2	1	Lösung
(a)	0	0	1	1	0	
(b)	1	0	1	1	0	
(c)	0	1	0	1	1	
(d)	1	1	1	1	1	

## Das Binärsystem



### Vom Dezimalsystem in Binärsystem:

Um eine Zahl vom Dezimalsystem ins Binärsystem umzurechnen gibt es verschiedene Verfahren. Eine Möglichkeit besteht darin, das auf dem letzten Arbeitsblatt verwendete Verfahren umzukehren und zu überprüfen, welche Zweierpotenzen in die gewählte Dezimalzahl passen.

Soll zum Beispiel die 21 in das Binärsystem umgewandelt werden, kann man folgende Tabelle zur Hand nehmen.

16	8	4	2	1

Nun wird überprüft, welches die größte Zweierpotenz ist, die in die 21 passt. In diesem Fall passt die 16 in die 21, somit wird an die erste Stelle der Tabelle (unter der 16) eine 1 notiert.

Danach wird die Differenz von 21 und 16 gebildet, um den Rest zu ermitteln, also  $21 - 16 = 5$ .

Es wird wiederum geschaut, welches die größte Zweierpotenz ist, die in die 5 passt, Die 8 entfällt, so dass an dieser Stelle in der Tabelle eine 0 notiert wird. Die nächste Zweierpotenz die passt, ist die 4. Demnach wird an der Stelle der 4 eine 1 notiert und wieder die Differenz von 5 und 4 gebildet:  $5 - 4 = 1$ . Da in die verbliebene 1 die 2 nicht hineinpasst, ergibt sich folgende Tabelle:

16	8	4	2	1
1	0	1	0	1

Somit wird die 21 im Binärsystem als  $10101_2$  dargestellt.

### Aufgabe:

 Führe das obige Verfahren mit den Zahlen 7, 14, 17 und 15 durch.

1. Rechnung:

16	8	4	2	1

2. Rechnung:

16	8	4	2	1

3. Rechnung:

16	8	4	2	1

4. Rechnung:

16	8	4	2	1

## Bastelanleitung

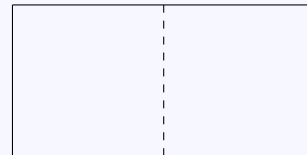


### Benötigtes Material:

- Calliope mini mit Batterien,
- 4 Kabel mit Krokodilklemmen,
- Pappkarton,
- Kupferklebeband.

### Das Büchlein:

- ▶ Schneide aus dem Pappkarton die Seiten aus. Für ein Büchlein benötigst du 5 Blätter im Format 10 cm × 20 cm.
- ▶ Falte diese nun in der Mitte (siehe Skizze) und stecke sie wie im Bild zusehen ineinander.
- ▶ Nun gibt es zwei Möglichkeiten
  - Entweder du tackerst das Büchlein in der Faltkante zusammen,
  - oder du benutzt Nadel und Faden um das Büchlein zu »binden«.
- ▶ Nun solltet ihr ein leeres Büchlein haben.



### Das Innenleben:

- ▶ Als erstes benötigen wir eine Karte mit der wir die Seitennummern auslesen können (siehe Bild).
- ▶ Dazu bekleben wir einen 10 cm × 10 cm Karton mit 4 Streifen Kupferklebeband
- ▶ Anhand dieser Vorlage können wir nun die einzelnen Seiten mit dem Kupferklebeband versehen.

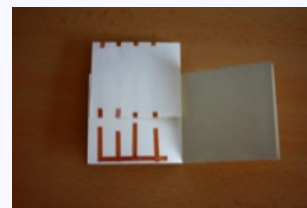
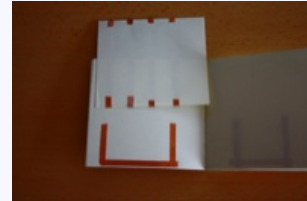


## Bastelanleitung



### Fortsetzung Innenleben:

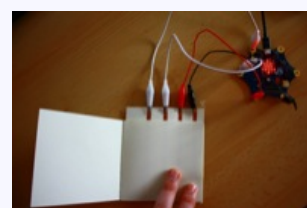
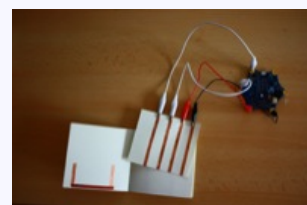
- ▶ Wir bekleben immer die Rückseiten der Seiten und zwar so, dass vom Minuspol auf unsere Karte, alle Kombinationen von P0 bis P3 erreicht werden.
- ▶ Der äußere Papierbogen bleibt dabei als *Umschlag* frei, ebenso wie die letzte Seite.
- ▶ Auf den beiden Bildern siehst du, wie z. B. die erste und letzte Seite des Büchleins beklebt werden, wenn du die folgende Kodierung verwendest:



P0	P1	P2	Zahl
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

### Fast fertig:

- ▶ Was jetzt noch fehlt ist die Verkabelung.
- ▶ Dazu werden der Minuspol, sowie die Anschlüsse P0 bis P2 von links nach rechts mit den Kupferbahnen auf der *Lesekarte* verbunden.
- ▶ Jetzt könnt ihr das Büchlein mit Leben füllen.
- ▶ Und natürlich den *Calliope mini*!



# Technische Informatik

## Ampel

### Unterrichtseinheit 6







## Ampel




### Ziele und Kompetenzen



Die Schülerinnen und Schüler ...

-  modellieren für einfache Problemstellungen, wie eine Ampelschaltung, Algorithmen, indem sie Abläufe formulieren und darstellen.
-  verwenden die Anschlüsse des *Calliope mini* um externe Peripherie wie LEDs anzusteuern.

### Zeit

 1–2 Doppelstunden




### Lernvoraussetzungen

-  Stromkreis
-  Kontrollstrukturen






### Empfohlene Vorgehensweise

- ▶ Ampeln kennen wir alle aus unserem alltäglichen Leben, doch kann man mit einfachen Mitteln eine Ampel bauen?
- ▶ Hierzu wird der informatische Modellierungskreislauf verwendet.
- ▶ Lassen Sie die Schülerinnen und Schüler zunächst überlegen, welche Phasen eine Ampel hat, wie man eine Ampel auslösen kann und wie lange die einzelnen Phasen dauern sollen.
- ▶ Für das entstandene Modell sollen die Schülerinnen und Schüler einen Ablaufplan entwickeln.
- ▶ Lassen sie die Schülerinnen und Schüler nun in Gruppen eine Ampel bauen und den Algorithmus auf dem *Calliope mini* umsetzen.

### Materialüberblick

-  Arbeitsblatt
-  Hilfe: Schaltplan Ampel
-  Mögliche Lösung – Ampel

### Zusätzliches Material

-  *Calliope mini* + Micro-USB-Kabel
-  Kupferklebeband
-  LEDs in rot, gelb und grün
-  Kabel mit Krokodilklemmen
-  Pappkarton

### Sonstige Hinweise

- ▶ Neben einer Ampel lassen sich so auch noch viele weitere Projekte realisieren, z. B. ein interaktiver Stadtplan, bei dem man verschiedene *Points of Interest* aufleuchten lassen kann.
- ▶ Der Kreativität sind keine Grenzen gesetzt.

## Ampel



### Aufgabe



Überlege dir die Reihenfolge der Ampelphasen und welche Farben jeweils aufleuchten.

### Aufgabe



Entwerfe einen Algorithmus, der eine Ampel simuliert. Überlegen dir, wie lange die Übergänge sein sollen und wie lange z. B. die Rotphase nach »drücken« der Ampel sein soll.

### Aufgabe:



Bastele nun mithilfe des Materials eine Ampel und programmiere den *Calliope mini* entsprechend deines Schaltplans.



*Hinweis:* Wenn du beim basteln oder beim Ansteuern der LEDs Schwierigkeiten hast findest du im Material einen Schaltplan bzw. Codeschnipsel zur Unterstützung.



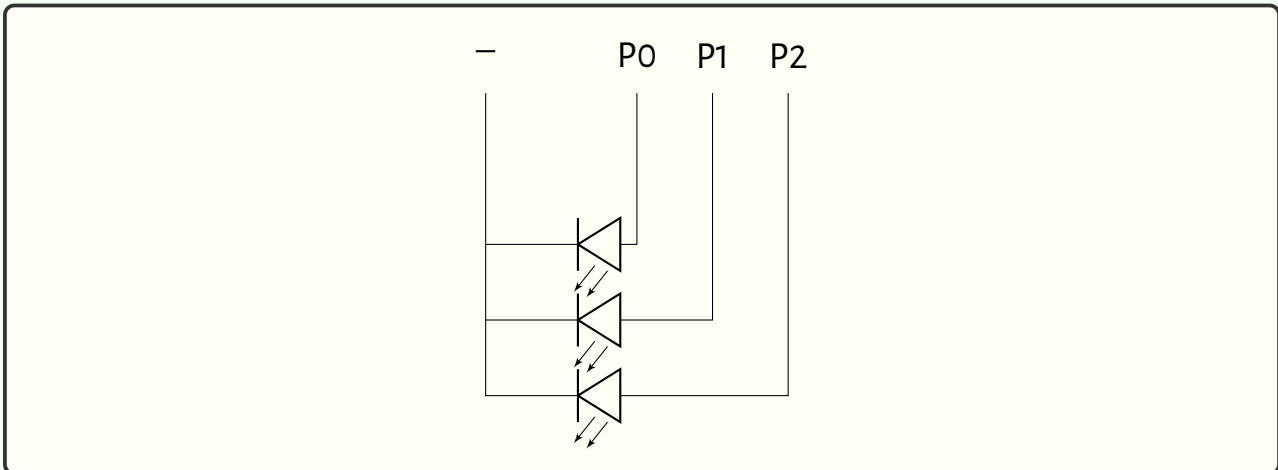
Teste deine Ampel, wenn du nicht das gewünschte Ergebnis erhältst, durchlaufe den Modellierungskreislauf noch einmal.


## Hilfe: Ampel



### Schaltplan

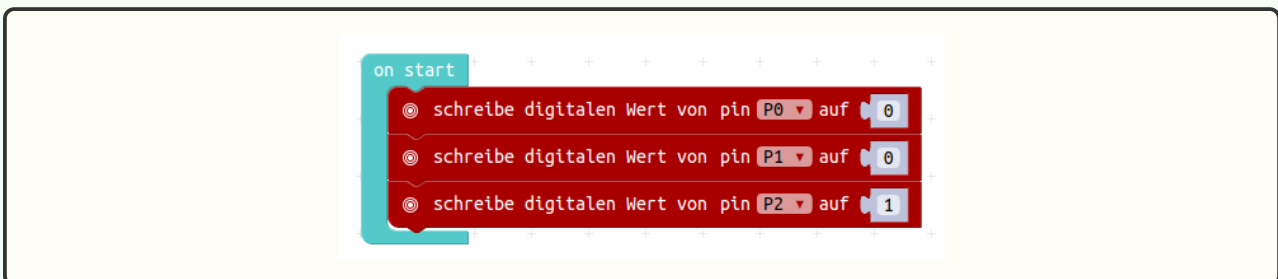
Eine mögliche Lösung für die Ampel sieht wie folgt aus:



Hierbei sind -, P0, P1 und P2 die Anschlüsse des *Calliope mini* und  steht für die LEDs. Da es sich bei LEDs um Halbleiter handelt, ist anders als bei herkömmlichen Glühbirnen auf die Einbaurichtung zu achten. Hierbei steht der Strich für das kurze Beinchen.

### Ansprechen der LEDs

Hast du Schwierigkeiten beim Ansprechen der LEDs? Hier finden Sie eine mögliche Startkonfiguration:



Unter der Annahme das die Zuordnung PIN-0 = Rot, PIN-1 = Gelb und PIN-2 = Grün ist, zeigt die Ampel in der Ausgangslage Grün.

## 5 Verwendungsnachweis

### Abbildungsverzeichnis

1	Information, Abbildung Pieper und Müller, <a href="#">2013</a> (vgl. Fuhr, <a href="#">2000</a> ). . . . .	5
2	Die Fachgebiete der Informatik, Anpassung durch Haase (vgl. Humbert, <a href="#">2016</a> ). . . . .	7
3	Modellierungskreis der Informatik (Humbert, <a href="#">2006</a> , S. 14). . . . .	8

## Literatur

- Claus, Volker und Andreas Schwill. *Duden Informatik A–Z. Fachlexikon für Studium und Praxis*. Hrsg. von Meyers Lexikonredaktion. 4., überarb. u. aktualis. Aufl. Mannheim, Leipzig, Wien, Zürich: Bibliographisches Institut, Feb. 2006. ISBN: 3-411-05234-1.
- Cormen, Thomas H. u. a. *Introduction to algorithms*. 2. 1<sup>st</sup> editon 1993. Cambridge, London: McGraw-Hill Book Company, 2001.
- Ernst, Hartmut. *Grundkurs Informatik*. 5., vollst. überarb. Aufl. Wiesbaden: Springer Fachmedien, 2015. ISBN: 978-3-658-01628-9.
- Ferrari, Anusca. *DIGCOMP: A Framework for Developing and Understanding Digital Competence in Europe*. JRC technical reports. Luxembourg: European Commission – Joint Research Centre – Institute for Prospective Technological Studies (JRC-IPTS), 21. Aug. 2013. URL: <https://is.gd/F96pvN> (besucht am 09. 06. 2016).
- Fricke, Martin u. a. »Informatik an Grundschulen – Modul Kryptologie Lehrerhandreichung (n. n. E.)« Didaktik der Informatik – Bergische Universität Wuppertal. Wuppertal, 2016.
- Fuhr, Norbert. *Informationssysteme – Stammvorlesung im WS 99/00 (IR-Teil)*. Jan. 2000. URL: <https://is.gd/MuWUOA> (besucht am 17. 05. 2016).
- Greve, Martin und Frederik Steingans. »Elemente der Technischen Informatik – ab der Grundschule«. In: *Seminarskriptum zur Veranstaltung »Ausgewählte Fragen zur Didaktik der Informatik«*. Hrsg. von Ludger Humbert. Wuppertal, 28. Nov. 2016. URL: <http://uni-w.de/bn> (besucht am 02. 01. 2017).
- Humbert, Ludger. *Didaktik der Informatik – mit praxiserprobtem Unterrichtsmaterial*. 2., überarbeitete und erweiterte Aufl. Leitfäden der Informatik. Wiesbaden: B.G. Teubner Verlag, Aug. 2006. ISBN: 3-8351-0112-9.
- *Veranstaltungskarte »Informatik im Alltag – Durchblicken statt Rummklicken« – Wintersemester 2016/2017*. 26. Juli 2016. URL: <http://uni-w.de/99> (besucht am 20. 10. 2016).
- Humbert, Ludger und Hermann Puhlmann. »Essential Ingredients of Literacy in Informatics«. In: *Informatics and Student Assessment. Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics*. Hrsg. von Johannes Magenheimer und Sigrid Schubert. Bd. 1. GI-Edition – Lecture Notes in Informatics (LNI) – Seminars S-1. Dagstuhl-Seminar of the German Informatics Society (GI) 19.–24. September 2004. Bonn: Köllen Druck+Verlag GmbH, 2004, S. 65–76. ISBN: 3-88579-435-7. URL: <http://is.gd/B6S18k> (besucht am 19. 02. 2016).
- Noller, Stephan. *Drawing Circuits: Bare-conductive vs. Circuit Scribe vs. Chibitronics*. 9. Dez. 2014. URL: <http://metager.to/9byza> (besucht am 24. 01. 2017).
- Pieper, Johannes und Dorothee Müller, Hrsg. *Material für den Informatikunterricht*. Arnsberg, Dortmund, Hamm, Wuppertal, Solingen, Juni 2013. URL: <http://uni-w.de/1t> (besucht am 29. 04. 2016).
- Sellin, Lukas. »Ausgewählte Elemente der theoretischen Informatik als Element der informatischen Bildung im Primarbereich«. Bachelorarbeit. Wuppertal: Fachgebiet Didaktik der Informatik – Bergische Universität, Apr. 2016. URL: <http://uni-w.de/4q> (besucht am 21. 08. 2016).
- Strecker, Kerstin. *Kryptografie*. Virtuelle Lehrerweiterbildung Informatik in Niedersachsen, 2009. URL: <http://vlin.de/vlin2/material/KryptologieVLIN.pdf> (besucht am 01. 01. 2016).
- Tech Will Save Us. *Animated micro:book*. URL: <http://metager.to/-icrf> (besucht am 24. 12. 2016).