

smile

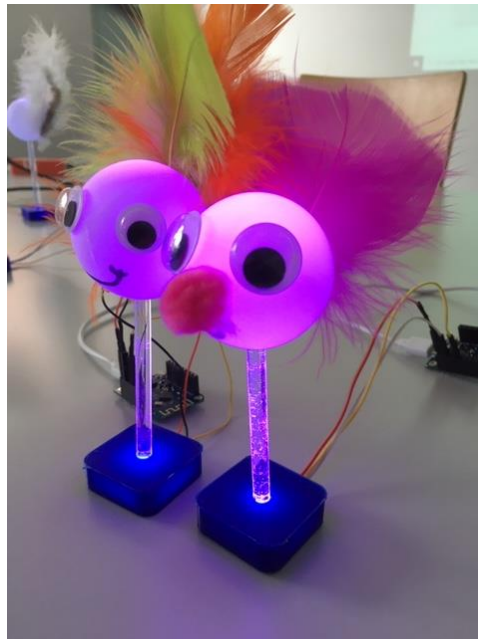


SMART · FUTURE · ME

GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung



## Beschreibung des Workshops: **Buntes Jahreszeiten Licht**

# Inhalt

Kurzzusammenfassung .....	3
Ziele .....	5
1.1    Affektive Lernziele .....	5
1.2    Kognitive Lernziele .....	5
1.3    Psychomotorische Lernziele .....	5
Lerninhalte .....	6
Variationen .....	9
1.4    Variante 1 – Stimmungsvolles Jahreszeiten Licht .....	9
Materialliste .....	10
1.5    Benötigte Materialien .....	10
1.6    Lern-Materialien .....	11
Verlaufspläne .....	12
Hinweise zur Blockprogrammierungsumgebung BEESM .....	16
„Lessons learnt“ .....	16
Anknüpfungspunkte mit anderen Workshops .....	17

Gestaltung und Konzeption dieses Workshops:



## Kurzzusammenfassung

Licht beeinflusst unser Wohlbefinden und unsere Stimmung erheblich und wird daher in Wohnsituationen zunehmend bewusst eingesetzt. Smart Home Technologien bieten darüber hinaus viele Möglichkeiten funktionale und dynamische Beleuchtung einzusetzen, die vor allem im Zusammenspiel mit anderen Geräten und Sensoren noch viel mehr kann als nur den Raum zu erhellen. In Anlehnung daran, entwickeln die Teilnehmerinnen (TN) in diesem Workshop ein kleines reaktives System in Form eines Licht-Objekts, das aus einzelnen RGB-LED-Lämpchen und Bastelmaterialien kreativ gestaltet wird.

Sensorik und Aktorik sowie Programmiergrundlagen sollen in diesem Workshop spielerisch vermittelt werden. Daher werden die TN ermutigt, kreativ zu sein und vielfältige Szenarien zu entwickeln, die zeigen, wie die Licht-Objekte auf die Umwelt mit verschiedenen Licht-Farben und Animationen reagieren können. Schließlich werden die Licht-Objekte auf Arduino Sensoren (z.B. PIR, Temperatur, Abstand-, Sound-, Lichtsensor etc.) reagieren, die zuvor explorativ erarbeitet werden. Beliebte Szenarien sind Licht-Objekte, die automatisch angehen, wenn der PIR Sensor Bewegung feststellt (Bewegungsmelder) oder wenn die Umgebung dunkel ist (automatisch dimmendes Licht). Dabei soll der Bezug zum Alltag/Smart Home immer wieder erörtert werden.

Je nach Zeit im Jahr sind verschiedene Workshop Themen möglich, die von den TN kreativ interpretiert werden können. Im Rahmen von smile wurden die Workshops „Osterlichter“, „Bunte Weihnachten“ und „Smarte Grusellichter“ durchgeführt. Die den Workshop begleitenden PowerPoint-Folien (Material\_02) beziehen sich hier beispielhaft auf den Workshop „Smarte Grusellichter“.



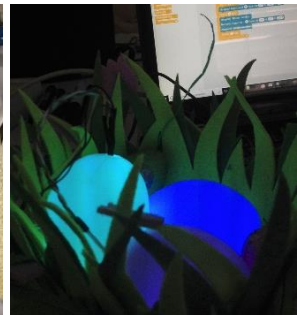
1 Ein Ping-Pong Ball dient als Lampenschirm. Mit Mullbinden werden daraus kleine Geister zu Halloween. Aus dem Workshop Smarte Grusellichter



2 Mini-Lampe: in Anlehnung an den Workshop „Der Blume geht ein Licht auf“. Eine LED, 3D-gedrucktes Kästchen, Acryglasstab, Ping-Pong Ball; hier als Halloweenmotiv im Grusellichter Workshop



3 1 LED von einem LED Streifen in einem 3D-Rahmen mit Sternfolie und Wintermotiv als Scherenschnitt.



4 Kurze LED-Kette im Osterkörbchen. Weiße Plastik Ostereier dienen als Lampenschirme.

Dieser Workshop ist eine kurze Variante des Workshops „Smarte Stimmungslichter“ und ist für jüngere TN konzipiert.

<b>Verwendete Technologien:</b>	Arduino (Uno), Grove-Shield und Grove-Sensoren. Blockprogrammierungsumgebung BEESM oder eine andere verfügbare Blockprogrammierungsumgebung wie mBlock <sup>1</sup> .	
<b>Geeignet für Labortyp:</b>	<b>x</b>	Unspezifisch
		FabLab
		Smart Home Lab
		Robotik Lab
<b>Zielgruppe/Klassenstufe:</b>	<b>X</b>	5. bis 7. Klasse
		8. bis 9. Klasse
		10. bis 11. Klasse
		12. bis 13. Klasse
<b>mögliche Zahl an Teilnehmenden:</b>	12	
<b>Workshopleitende:</b>	2-3 (bei 12 Teilnehmerinnen)	
<b>Geschätzter Zeitaufwand:</b>	15 Std. -> 3 - 4 Tage	
<b>Lernziele:</b>	<ul style="list-style-type: none"> <li>➤ Grundlegende Programmierkonzepte;</li> <li>➤ Sensorik und Aktorik;</li> <li>➤ Additives RGB Farbmodell &amp; Farbmischung;</li> <li>➤ Informatik als kreatives Fach.</li> </ul>	
<b>Vorkenntnisse der Schülerinnen:</b>	<ul style="list-style-type: none"> <li>➤ Keine</li> </ul>	
<b>Vorkenntnisse der/des Workshopleitenden:</b>	<ul style="list-style-type: none"> <li>➤ Grundlagen Arduino Programmierung: Neopixel LEDs, Sensoren;</li> <li>➤ Blockprogrammierung: BEESM, Ardublockly, alternativ mBlock etc.</li> </ul>	
<b>Voraussetzungen an die Infrastruktur:</b>	<ul style="list-style-type: none"> <li>➤ -</li> </ul>	
<b>Sonstige Voraussetzungen:</b>	<ul style="list-style-type: none"> <li>➤ Keine</li> </ul>	

<sup>1</sup> <https://www.mblock.cc/en-us/>

# Ziele

## 1.1 Affektive Lernziele

- Die Teilnehmenden stellen eine positive emotionale Verbindung mit der Informatik her.
- Die Teilnehmenden haben ein erhöhtes Selbstbewusstsein in Bezug auf IT.
- Die Teilnehmenden verstehen Informatik als wichtige Disziplin, die es erlaubt, die zunehmend digitalisierte Umwelt zu verstehen und mitzugestalten.

## 1.2 Kognitive Lernziele

- Die Teilnehmenden können je 2 Beispiele für Sensoren und Aktoren beschreiben und diese in Form von praktischen Beispielen der Eingabe, Verarbeitung und Ausgabe zuordnen.
- Die Teilnehmenden verstehen ein (Computer-) Programm als Folge von Anweisungen/Befehlen, die der Rechner der Reihe nach ausführt um bestimmte Aufgaben zu bearbeiten oder zu lösen.
- Die Teilnehmenden verstehen grundlegende Programmierkonzepte mit Hilfe einer Blockprogrammiersprache, das sind bedingte Anweisungen und Variablen.
- Die Teilnehmenden verstehen das Prinzip der additiven Farbmischung (RGB) und mischen Farben aus den drei RGB Werten.
- Die Teilnehmenden können eigene Ideen und Vorgehensweisen darstellen.

## 1.3 Psychomotorische Lernziele

- Die Teilnehmenden lernen die Eingabe mit Maus und Tastatur.

## Lerninhalte

Ziel des Workshops ist es, ein smartes, automatisch auf Sensoren reagierendes, Licht-Objekt programmieren und zu konstruieren. Die TN können in dem Workshop eigene Ideen und Anwendungsszenarien dazu entwickeln, die somit möglichst an die eigene Lebenswelt anknüpfen. Die Licht-Objekte können frei gestaltet werden.

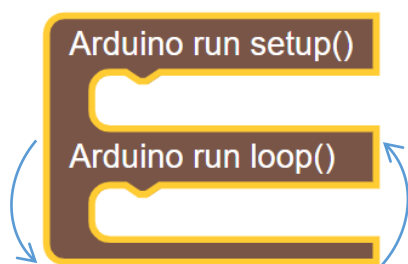
Das Programmieren der einzeln adressierbaren LEDs (RGB Neopixel, mit dem Chip WS2811/12(B)) ist gut für den Programmier-Einstieg geeignet: Mit einfachen Programmen können schnell ansprechende und attraktive Resultate erzielt werden. Die LEDs veranschaulichen den Programmablauf bei der Ausführung unmittelbar visuell.

### **Arduino Programmierung - Grundlegende Programmierkonzepte**

Um die Szenen umzusetzen, ist es notwendig grundlegende Programmierkonzepte anzuwenden, die mit Hilfe einer Blockprogrammiersprache vermittelt werden. Dabei werden die TN an die Grundlagen der Arduino-Programmierung und des Physical Computing herangeführt, indem gezeigt wird, wie sie mit dem Arduino, einem passenden Display, einer Neopixel-LED und mit einer Block-Programmierungsumgebung arbeiten können. Nach einer grundlegenden Einführung werden weitere Programmierkonzepte vermittelt und mit einer RGB-LED erprobt.

Diese Workshopbeschreibung bezieht sich auf die Programmierungsumgebung BEESM, die auf Google's Blockly basiert<sup>2</sup> und vom Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)<sup>3</sup> im Rahmen des smile Projekts entwickelt wurde. Am Ende dieser Dokumentation befinden sich weitere Hinweise und Literatur dazu.

Im Folgenden werden die Blöcke gezeigt, so wie sie in Blockly's BEESM gestaltet sind:



Die Arduino **Programmstruktur** besteht aus der **Setup**-Funktion und der **Loop**-Funktion.

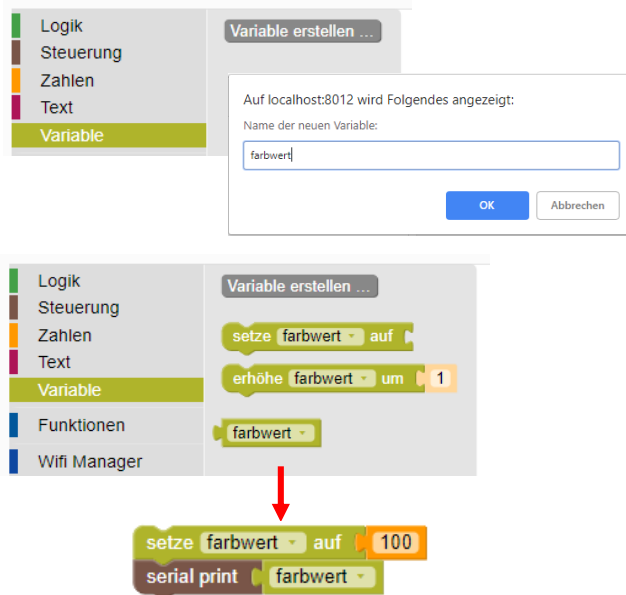
Im Setup stehen Anweisungen, wie z.B. grundlegende Definitionen, die einmal zu Beginn des Programms (z.B. nach dem Hochladen des Programms) ausgeführt werden. Die Anweisungen in der Loop werden nach dem Setup in einer Endlosschleife wiederholt.

Um ein Gefühl zu vermitteln wie und wie schnell ein Programm ausgeführt bzw. wie die Anweisungen beim Programmablauf abgearbeitet werden, kann zu Beginn ein Blink-Sketch (oder ein „Hello World“ Sketch) im Setup, in der Loop sowie mit und ohne der `warte()` Funktion programmiert werden.

<sup>2</sup> <https://developers.google.com/blockly/>

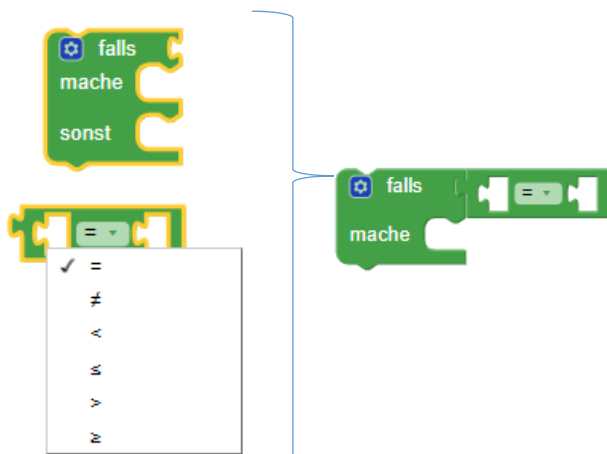
<sup>3</sup> Forschungsbereich Cyber Physical Systems (CPS) Bremen

Eine **Variable** ist ein Platz, um Werte zu speichern, ähnlich wie ein Karton, in dem man etwas hineintun kann. Die Variable muss eindeutig mit einem Namen bezeichnet werden, damit der Rechner im Laufe der Programmausführung auf den Wert in der Variablen zugreifen kann. Der Wert einer Variable kann im Laufe der Programmausführung verändert werden.



Um einen Wert in einer Variable zu speichern, wird der Block „setze *variablenname* auf x“ verwendet. Zum Lesen wird die Variable in eine andere Anweisung reingezogen. Um das Konzept der Variablen zu üben wird z.B. ein Zähler programmiert.

**Bedingungen** dienen zum Steuern des Programmflusses und umschreiben: „**Falls** dies zutrifft **dann** mache dies, **sonst** das“.

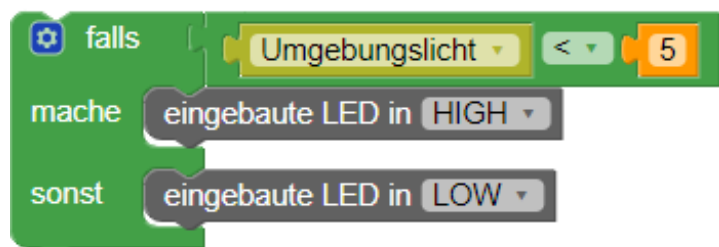


Aus dem Alltag kennen wir Sätze wie z.B.: „Wenn ich müde bin, dann gehe ich schlafen.“ Damit drücken wir aus, dass bestimmte Aktionen nur ausgeführt werden, wenn eine Bedingung erfüllt ist (z.B. müde sein).

Um diese Verzweigungen im Programmablauf zu realisieren, benötigen wir einen

Bedingungsblock und einen Vergleichsblock, mit dem überprüft wird, ob der Vergleich zweier Werte wahr ist oder falsch.

Beispiel: **Falls** es dunkel ist, **mache** (schalte) die LED Lampe an (HIGH), **sonst** schalte die LED Lampe aus (LOW). Der Licht-Wert ist in der Variable „Umgebungslicht“ gespeichert. Ist dieser Wert kleiner als 5, dann ist es dunkel.



Die TN sollten über ein Beispiel-Programm, das Bedingungen anwendet, an das Konzept herangeführt werden und dieses im weiteren Projektverlauf als Referenz heranziehen können.



## Additives Farbmodell

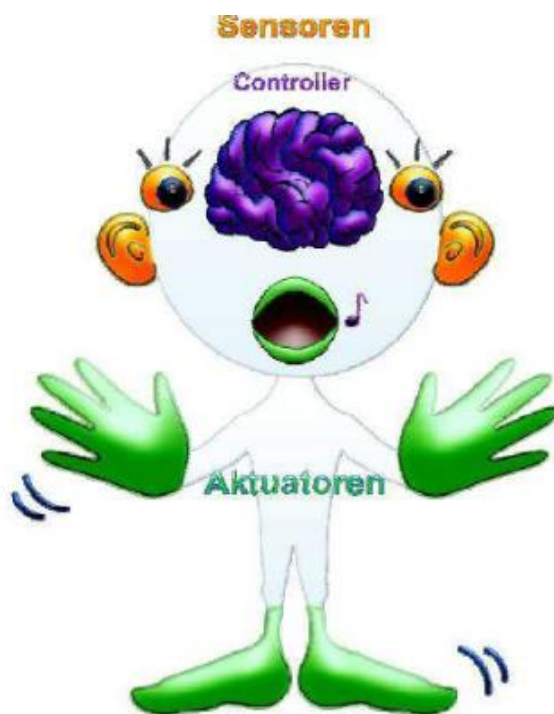
Das Programmieren der RGB-LEDs erfordert Grundkenntnisse des additiven Farb-Modells. Dieses ist von dem subtraktiven Farbmodell zu unterscheiden, das die TN evtl. aus dem Mischen von Farben mit dem Tuschkasten kennen (im allg. Druckfarben). Das additive Farbmodell bezieht sich allerdings auf die Lichtfarben rot, grün und blau, die zueinander addiert werden und mit denen z.B. moderne Monitore/Displays, Projektoren, Fernseher oder LEDs arbeiten, da diese Licht ausstrahlen.

Im RGB-Farbraum steht R für rot, G für grün und B für blau. Die Farbintensität jeder Farbe kann zwischen 0 und 255 liegen. Wenn alle drei Farben mit je 255 zusammengemischt werden (255, 255, 255) entsteht weiß in maximaler Intensität. Bei (0,0,0) entsteht schwarz und ist gleichbedeutend mit „kein Licht“.

Durch Experimentieren Farben lernen die TN schrittweise den Bezug der Zahlen zu den Lichtfarben.

## Sensorik/Aktorik

Bei smarten Objekten sind zumeist verschiedene Geräte vernetzt, die mit **Sensoren** zur Erfassung der Umgebung ausgestattet sind und auch mit **Aktoren**, die aktiv in die Umgebung eingreifen: Der Sensor (lat. „sentire“, -> fühlen, empfinden) misst Größen wie z.B. Helligkeit, Temperatur etc. in seiner Umgebung. Der Aktor (lat. „agere“, -> agieren, handeln, tun, wirken) als „Gegenstück“ zum Sensor wirkt auf etwas ein, das kann zum Beispiel Bewegung durch einen Motor oder eine LED sein.



5 Analogie Männchen

Jüngeren TN kann dieser Zusammenhang in Analogie zum Menschen oder allgemein eines Lebewesens, (z.B. dem Analogie-Männchen<sup>4</sup>, siehe Abb. 5 links) erklärt werden. Die Sinnesorgane (Augen, Ohren etc.) nehmen Reize auf und leiten diese ans Gehirn weiter. Dieses ist vergleichbar mit Sensoren in der technischen Welt. Mit dem Bewegungsapparat und der Stimme können wir agieren, ähnlich wie die Aktoren (Motoren, Lautsprecher, Lampen etc.). Genauer wird das Analogie-Männchen im Kontext Informatik-Unterricht beschrieben in Dittert, Wajda, Schelhowe (2016).

Welche Beispiele kennen die TN für Sensoren/Aktoren und smarte Systeme aus Ihrem Alltag?

<sup>4</sup> Quelle: Dittert, N., Wajda, K. & Schelhowe, H., 2016. *Kreative Zugänge zur Informatik: Praxis und Evaluation von Technologie-Workshops für junge Menschen*. Staats- und Universitätsbibliothek Bremen (Open Access) [online] <http://elib.suub.uni-bremen.de/edocs/00105551-1.pdf>; S. 100-101



# Variationen

## 1.4 Variante 1 – Stimmungsvolles Jahreszeiten Licht

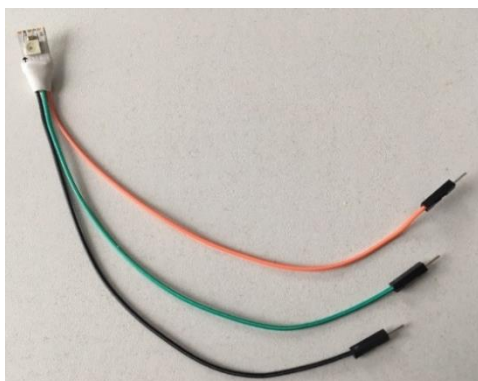
Phase	Kurze Zusammenfassung	Dauer: ca. 4 Tage (20 Std)
Einstieg	Motivation des Themas Einführung in smarte Technologien. Was kann smartes Licht bedeuten?	
Programmierung I: Hallo Welt!	Einführung, und praktische Beispiele für die eingesetzten Technologien: <ul style="list-style-type: none"><li>■ Arduino</li><li>■ Block-Programmierungsumgebung</li><li>■ Ein Programm</li><li>■ Ein Display für den Arduino.</li></ul> Die TN schreiben ein erstes Programm (z.B. Hallo Welt) für den Arduino	
Programmierung II: RGB-LED	Die TN programmieren eine RGB-LED und lernen das additive Farbmodell kennen.	
Programmierung III: Sensorik/Aktorik	Die TN lernen das Prinzip Sensorik/Aktorik kennen. Die TN befassen sich explorativ mit ausgewählten Sensoren und vertiefen dabei die Programmierkonzepte Variablen und Bedingungen.	
Projektarbeit	Die TN setzen eigene Licht-Objekte prototypisch um. Die Licht-Objekte werden gebastelt und programmiert.	

# Materialliste

## 1.5 Benötigte Materialien

12 TN, für Gruppen zu je 2 TN:

- Mindestens 6x Laptops/Computer
- Neopixel LEDs (WS2811/12(B)), 6 x einzelne LEDs (gelötet mit Kabel zum Verbinden mit Mikrocontroller (siehe Abbildung 6, besser einige LEDs auf Vorrat, da diese schnell kaputt gehen).
- 6 x Arduino: Arduino/Seeeduino mit Seeed Grove Shield zum Einstieg<sup>5</sup> mit USB-Kabel, Netzteil oder Powerbank (siehe Abbildung 7)
- 6 x Seeed Grove RGB LC Display<sup>6</sup> (siehe Abbildung 7)
- Verschiedene Grove Sensoren (z.B. Licht, Sound, Ultraschall, PIR, Temperatur, Druckknopf ...), und passende Grove Kabel
- Bastelmaterialien, evtl. vorgefertigte Arbeitsmaterialien zur Gestaltung der Licht-Objekte, z.b. Schmuck/"Lampenschirme" passend zu Ostern (Plastik-Ostereier, Material/Anleitung für Origami Tulpe), Transparent-/Architektenpapier, Effektfolie, Kästchen/5mm & Acryglasstap (siehe smile Workshop "Der Blume geht ein Licht auf") Ping-Pong-Ball, Kleine 3D Rahmen etc. (siehe Abbildungen 1 – 4)
- Um den TN die Objekte mit nach Hause geben zu können, werden noch 12 x günstigere Controller, wie der Wemos D1 mini, einfache Sensoren und für jede TN eine LED benötigt. Sie Abschnitt "Lessons learnt"



6 Neopixel LED, einzelne LED abgeschnitten von einem Led Strip, Jumper-Kabel an LED gelötet.



7 Arduino Uno mit Grove Shield und Grove RGB LC Display

<sup>5</sup> <https://www.seeedstudio.com/Base-Shield-V2.html>

<sup>6</sup> <https://www.seeedstudio.com/Grove-LCD-RGB-Backlight.html>

## 1.6 Lern-Materialien

Nr.	Titel	Beschreibung	Benötigt für Variante(n)
_01	Workshopbeschreibung		
_02	PowerpointJahreszeitenLicht	PowerPoint Präsentation, begleitend zum Workshop Als Beispiel hier: Workshop „Smarte Grusellichter“	1
_03	Handout NeopixelBlöcke	Übersicht über Blöcke in BEESM, mit denen Neopixel LEDs programmiert werden	1
_04	Übungen LED	LED programmieren und das additive Farbmodell: Erklärungen und kleine Aufgaben. Arbeitsblätter geeignet für individuelle Gruppenarbeit oder Bearbeitung in Gesamtrunde.	1
_05	Challenge RGB Farbe	Farbkarten zum Ausdrucken und Ausschneiden, LED-Farben sollen entsprechend programmiert werden	1
_06	Challenge Grove Sensoren	Aufgaben mit Hilfestellung, um verschiedene Sensoren explorativ zu erarbeiten	1
_07	Challenge Grove Sensoren Lsg	Lösungen zum Material_06/Challenge Grove Sensoren. Können zum Gestalten der Dokumentationen verwendet werden.	1
_08	Vorlage Sensor Dokumentation	Vorlage kann von TN benutzt, um Challenge Grove Sensoren (Material_06) zu lösen, d.h. die Sensoren übersichtlich zu dokumentieren	1

# Verlaufspläne

## Einstieg; Programmierung Hallo Welt (~ 3 Stunden)

Zeit	Phase	Inhalt	Material:
45 Min.		<i>Es gibt 45 min. Zeit zum Kennenlernen, Pre-Fragebogen sowie Interviews</i>	<b>02_PowerpointJahreszeitenLicht, Folien 2 - 17</b>
25 Min.	Einstieg	<p>Die Teilnehmerinnen sollen sich in 2er Gruppen mit einer oder mehreren Fragen beschäftigen (Flüsterrunde). Auch sehr fantasievolle Themen der TN sollen dabei aufgenommen werden.</p> <ul style="list-style-type: none"> <li>■ Welche nützlichen intelligenten Gegenstände für den Alltag gibt es, die sie gerne hätten?</li> <li>■ Was kann ein Smartes Stimmungslicht sein?</li> </ul> <p>Die TN machen sich Notizen. Anschließend werden Ideen zusammengetragen.</p>	
10 Min.	Hinführung	<p>Technikeinführung: Arduino und Seeed Grove System</p> <p>Verteilen und Zusammenbau der Hardware. Das RGB Display wird an den I2C Pin gesteckt</p>	Arduino, Seeed Grove Shield, usb-Kabel, + Grove RGB LC-Display
90 Min.	Erarbeitung	<p>Einführung in die Programmierung mit Aufgaben:</p> <ul style="list-style-type: none"> <li>■ Die Programmierumgebung (BEESM) wird vorgestellt und wird von den TN ausprobiert</li> <li>■ Einführung Arduino Programmstruktur: Setup und Loop</li> <li>■ Blöcke (Programmieranweisungen) für Ausgabe auf Display</li> <li>■ <b>Aufgabe:</b> Ein erstes Programm: Ein Text soll ausgegeben werden (z.B. „Hallo Welt“)</li> <li>■ Die TN lernen experimentell das Display kennen.</li> </ul>	
10 Min.	Sicherung	<p>Diskussion und Zusammenfassung:</p> <ul style="list-style-type: none"> <li>■ Display: Wieviel Text passt darauf?</li> <li>■ Ausgabe im Setup vs. Ausgabe in der Loop</li> </ul>	

## Vertiefung I; Programmierung RGB LED (~ 3 Stunden)

Zeit	Phase	Inhalt	Material
			<b>02_PowerpointJahreszeitenLicht, Folien 18-26</b>
10 Min.	Einstieg	Neopixel LEDs werden vorgestellt. Lampen können zur Motivation gezeigt werden, aber ohne zu viel vorwegzunehmen.	LED Lämpchen, Beispiellampen als Anschauungsobjekte
10 Min.	Hinführung	Technikeinführung: LED austeilen. Die TN verbinden Ihre LEDs mit dem Controller. Die KL helfen und kontrollieren.	
10 Min.	Hinführung	Blöcke (Programmieranweisungen) um LEDs zu schalten werden vom KL erläutert.	03_Handout NeopixelBlöcke
120 Min.	Erarbeitung	<p><b>Ein erstes LED Programm</b> soll auf der Basis des Handouts von den TN selbstständig geschrieben werden: Eine LED soll in einer beliebigen Farbe leuchten.</p> <p>Die TN erarbeiten das additive Farbmodell anhand des Übungsblattes:</p> <ul style="list-style-type: none"> <li>Arbeitsauftrag/Challenge: Notieren die Zahlentripel der Haupt- und grundlegenden Mischfarben, bestimme experimentell die Zahlentripel der Farben auf den Farbkarten</li> </ul> <p>Einführung in die Neopixel-Programmierung mit Übungsaufgaben:</p> <ul style="list-style-type: none"> <li>einzelnes Pixel einfärben</li> <li>Farbintensität/Helligkeit variieren</li> <li>Einzelnes Pixel soll blinken</li> <li>Pixel soll faden (ggf. mit geeigneten Block)</li> <li>Raum für eigene Ideen/Experimente soll gegeben werden</li> </ul> <p>Je nach Lerngeschwindigkeit und Interessen der TN kann eine kreative Herausforderung zum Abschluss gegeben werden: Modellieren der LED Farben/Animationen, die z.B. einen Wetterzustand (sonnig, wolkig, Gewitter, Regen, ...) oder eine Emotion (glücklich, traurig, überrascht, böse, ...) ausdrücken.</p>	<p>04_Übungen LED</p> <p>05_Challenge RGB Farbe</p>
10 Min.	Sicherung 1	Ausgewählte Ergebnisse aus den Aufgaben werden gemeinsam besprochen. Anschließend Diskussion: Was ist ein Programm?	
20 Min.	Sicherung 2	Die Ergebnisse der kreativen Challenge werden von den TN in der Gesamtrunde vorgestellt	

## Vertiefung II; Programmierung Aktoren/Sensoren (~ 2,5 Std)

Zeit	Phase	Inhalt	Material
			<b>02_PowerpointJahreszeitenLicht, Folien 27 - 40</b>
10 Min.	Einstieg	Einführung in das Thema Aktoren und Sensoren.	
30 Min.	Erarbeitung	Der KL erklärt das Konzept <b>Variablen</b> und verschiedene Übungen dazu werden Schritt für Schritt in der Gesamtgruppe bearbeitet. Variablen werden benötigt, um Werte der Sensoren speichern und verarbeiten zu können.	
10 Min.	Sicherung	Ausgewählte Aufgaben werden besprochen.	
80 Min.	Bearbeitung	<p>Jede TN-Gruppe erhält einen anderen Sensor (z.B. Abstand, Licht, Temperatur, Schalter, PIR...), um diesen experimentell zu bestimmen: Dazu bearbeiten die Gruppen selbstständig „Challenges“ zu ihren Sensoren. Die KL helfen in den Gruppen, z.B. um die Sensoren unter unterschiedlichen Bedingungen zu testen.</p> <ul style="list-style-type: none"> <li>■ Was misst der Sensor</li> <li>■ Wie/Wo wird er angeschlossen</li> <li>■ Welche Werte werden ausgegeben?</li> <li>■ Welche Programmieranweisungen/Blöcke werden benötigt?</li> <li>■ Anwendungen des Sensors aus dem Alltag bekannt?</li> </ul> <p>Jede Gruppe bereitet eine Dokumentation mit den Lösungen aus der Challenge übersichtlich für die anderen TN auf. Um diese zu gestalten, können die Lösungen (Material_07) und die Vorlage (Material_08) verwendet werden.</p> <p>Danach werden <b>Bedingungen</b> vom Kursleiter in der Gesamtgruppe eingeführt. Die TN sollten über ein Beispiel-Programm, das Bedingungen anwendet, an das Konzept herangeführt werden. Auf dieser Basis schreiben die TN ein Programm zu „ihrem“ Sensor.</p>	<p>06_Challenge Grove Sensoren</p> <p>07_Challenge Grove Sensoren Lsg 08_Vorlage SensorDokumentation</p>
20 Min.	Sicherung	Jede Gruppe präsentiert das, was sie über ihren Sensor herausgefunden und programmiert hat.	

## Umsetzung der Projekte (~ 6,5 Stunden)

Zeit	Phase	Inhalt	Material
30 Min.	Einstieg	In der Gesamtrunde werden einzelne Ideen der TN besprochen. Es ist wichtig, dass die Umsetzung mit dem KL geplant wird.	
3 Std.	Erarbeitung	Die Licht-Objekte werden umgesetzt. Das beinhaltet das Programmieren und gestalten der Lampen.	
60 Min	Sicherung	Die TN erstellen eine Präsentation (Poster) um das Licht-Objekt vorzustellen.	
60 Min.	Sicherung	Ergebnisse werden von den TN präsentiert.	
60 Min.		<i>Anmerkung: Es gibt 60 min. Zeit für Abschluss, Post-Fragebogen sowie Interviews, ggf. Role Modells/Laborführung etc...</i>	



# Hinweise zur Blockprogrammierungsumgebung BEESM

BEESM ist eine visuelle blockbasierte Programmierungsumgebung, die in einführenden Programmierkursen und Workshops eingesetzt werden kann. BEESM ermöglicht es, intelligente Umgebungen zu programmieren. Das heißt konkret, dass unser Smart Home BAALL, Arduino-Code und Robot Operating System (ROS) für den Turtlebot3 einzeln und in Kombination miteinander programmiert werden können. BEESM wurde in mehreren einführenden Programmierkursen und Workshops im Rahmen von SMILE eingesetzt und evaluiert, um herauszufinden, wie Programmierkenntnisse und Einstellung der Schüler/innen zur Programmierung im Laufe der Zeit beeinflusst werden.

Konferenzpaper dazu: <https://dl.acm.org/doi/abs/10.1145/3240167.3240239>

Die Software auf GitHub: <https://github.com/projekt-smile/BEESM>

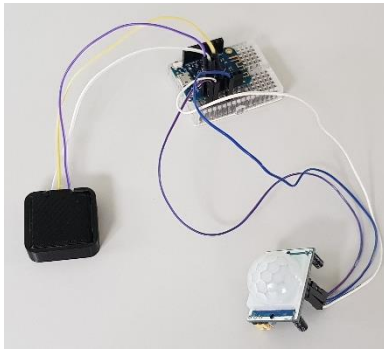
## „Lessons learnt“

Besonders motivierend ist die Umsetzung eigener Ideen, also die Phasen freier Projektarbeiten in diesem Workshop, wie die hier als „Challenges“ bezeichneten Übungsprojekte zwischendurch und natürlich die eigenen Projekte am Ende des Workshops. Allerdings ist der Betreuungsaufwand der freien Projekte sehr hoch, da Anfängerinnen viel Unterstützung bei der Programmierung benötigen und sowohl die Anwendung/Programmierung als auch die Lampen individuell gestaltet werden. Aus unserer Erfahrung heraus empfehlen wir 3 Kursleiter für eine Gruppe von 12 TN, damit keine allzu langen und frustrierenden Leerlaufzeiten für die TN entstehen. Ist der Betreuungsschlüssel kleiner, sollte die Vielfalt der möglichen Projekt-Themen eingeschränkt werden.

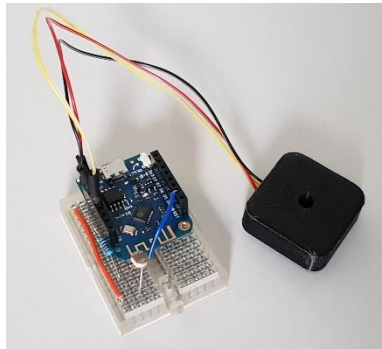
Ebenso gehört es zur Motivation, dass die TN ihre individuell gestalteten Licht-Objekte mit nach Hause nehmen können, was bei dem Arduino Grove System, das als Prototyping Plattform gedacht ist, aus Kostengründen nicht immer leistbar ist. Außerdem sind die Grove Bauteile sperrig und schwer in kleine Licht-Objekte zu verbauen. Realistischer ist es den TN ihre Objekte mit dem günstigen Controller Wemos D1 mini und einfachen Sensoren mitzugeben. Nachdem die Prototypen der Licht-Objekte mit den entsprechenden Grove System fertig programmiert sind, laden die TN (oder KL) über BEESM ihren Block-Code auf den Wemos hoch, oder es wird die Arduino IDE genutzt, um den generierten Code auf den Controller zu laden. Hierbei kann auch sehr gut auf die Vorgehensweise des Prototyping bei der Software-Entwicklung eingegangen werden.

Gegebenenfalls muss der Code zur Verarbeitung der Sensorwerte angepasst werden bzw. der entsprechende Block in BEESM getauscht werden. Dazu müssen die Dokumentationen der entsprechenden Sensoren herangezogen werden.

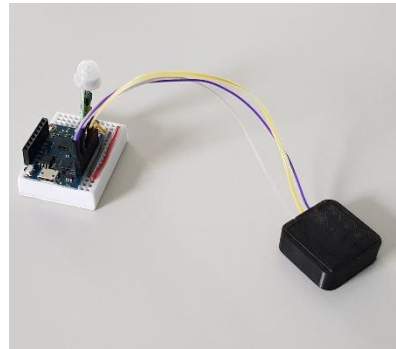
Die KL stecken die Systeme (Controller, gewünschter Sensor, LED) auf kleinen Steckbrettern für die TN zusammen (vgl. folgende Abbildungen), da eine entsprechende Lerneinheit hierzu nicht vorgesehen ist.



8 PIR, kleines Steckbrett, LED in einem gedruckten Kästchen, in dem die Lampen gesteckt werden



9 Lichtsensitiver Widerstand, kleines Steckbrett, LED in einem gedruckten Kästchen.



10 Mini PIR, kleines Steckbrett, LED in einem gedruckten Kästchen.

## Anknüpfungspunkte mit anderen Workshops

Dieser Workshop ist eine kurze Version des Workshops „**Smartes Stimmungslicht**“, der für jüngere Schülerinnen bis einschließlich Klassenstufe 7 geeignet ist. Die Lerninhalte und Ziele sind sehr ähnlich, diese werden aber weniger detailliert und mit mehr praktischen Beispielen und Vorgaben vermittelt.

Für diese jüngere Zielgruppe hat sich bewährt mit dem handlichen Arduino Grove System zu arbeiten, um Ideen schnell und prototypisch umsetzen zu können. Dieses erlaubt es, auch Elektronik-Anfängern/innen durch Ausprobieren bzw. Experimentieren herauszufinden, was machbar ist und eigene Ideen umzusetzen. Des Weiteren wird nur mit einer Neopixel LED gearbeitet, womit das Thema Schleifen entfallen kann. Als motivierender Faktor wird ausreichend Raum zum Basteln gegeben um die Licht-Objekte individuell zu gestalten. Als Beispiele seien hier die Workshops unter dem Motto „Halloween Grusellichter“, „Bunte Ostern“ oder „Smarte Weihnachtslichter“ genannt.