

smile

IT

SMART · FUTURE · ME

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



Beschreibung des Workshops: **Smartes Stimmungslicht**

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 – Smartes Stimmungslicht im Smart Home	9
1.5 Variante 2 – Smartes Stimmungslicht, IoT Version	9
1.6 Variante 3 – Smartes Stimmungslicht offline	10
Materialliste	11
1.7 Benötigte Materialien	11
1.8 Lern-Materialien	12
Verlaufspläne	13
1.9 Variante 1	13
1.10 Variante 2	17
1.11 Variante 3	18
Hinweise zur Blockprogrammierungsumgebung BEESM	20
„Lessons learnt“	20
Anknüpfungspunkte mit anderen Workshops	20

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 diesem Workshop entwickeln die Teilnehmerinnen (TN) ein kleines reaktives System in Form eines Licht-Objekts, das aus einzelnen RGB-LED-Lämpchen, oder auch Strips, Ringen, Matrizen, Lichterketten (etc.) und Bastelmaterialien kreativ gestaltet wird.

Um Grundlagen der Sensorik und Aktorik zu vermitteln, werden die TN ermutigt, verschiedene Szenarien zu entwickeln, die zeigen, wie die Licht-Objekte auf die Umwelt bzw. die Smart Home Umgebung mit verschiedenen Licht-Farben und Animationen reagieren können. Als Grundlage dazu wird das Thema Smart (Home) erörtert und die entsprechende Umgebung explorativ erarbeitet. Als Alternative zum Smart Home hat sich eine IoT-Variante 2 bewährt, die z.B. Online-Wetterdaten integriert. In Variante 3 reagieren die Licht-Objekte auf Sensoren, die mit dem Arduino verbunden sind (z.B. PIR, Temperatur-, Ultrasonic-, Lichtsensor etc.). Eine Kombination aus den drei Varianten ist für erfahrene TN möglich.

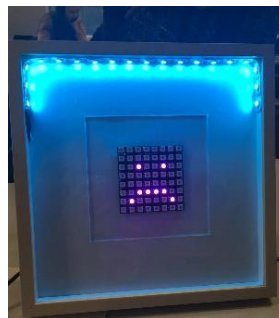
Je nach technischen Voraussetzungen sind verschiedene Szenarien möglich, die von den TN möglichst frei zu gestalten sind. Beispiele: Lichtobjekte zeigen das Wetter an (Temperatur, Regen-Wahrscheinlichkeit etc.). Die Lichtfarbe einer Lampe passt sich der Tageszeit an oder ein LED Wecker ermöglicht das Aufwachen durch langsam hochdimmendes Licht. Eine Lampe passt sich der Umgebungs-Helligkeit und/oder den anderen Lampen im Smart Home an. Eine Emotionslampe reagiert auf die vom Gesichtserkennungssystem interpretierten Emotionen. Eine Lampe dient als Warnsystem, wenn z.B. Bewegung festgestellt wird oder der Kühlschrank nicht die gewünschte Temperatur hat. Denkbar ist es auch, eine Lichtstimmung zusammen mit den Lampen im Smart Home für verschiedene Szenen wie z.B. „Party“, „Abendessen“, „Kino“ etc. zu programmieren.



1 Aus Käseschachtel, 24er LED Ring bzw. Strip & Architektenpapier oder Effektfolie kann eine Laterne gebastelt werden.



2 60er LED Ring als Uhr programmiert.



3 Neopixel Matrix und Strip sind in einem Bilderrahmen eingebaut.



4 3D-Bilderrahmen, Separatoren aus Pappe, weißes Papier, einzelne LEDs (in Anlehnung an Weather-Display¹).

¹ <https://www.instructables.com/id/WiFi-Weather-Display-With-ESP8266/>

Verwendete Technologien:	<p>Variante 1 + 2: Wifi-fähiger Mikrocontroller (ESP 8266, z.B. Wemos D1 (mini), Adafruit Feather Huzzah, etc.), Arduino IDE, BEESM um den ESP8266 Controller zu programmieren und wie in Variante 1 das Smart Home anzusteuern.</p> <p>Variante 3: Arduino (Uno) und Sensoren. Da in dieser Variante kein Wifi benötigt wird, kann alternativ zu BEESM auch mit der frei verfügbaren Blockprogrammierungsumgebung mBlock (o.ä.) gearbeitet werden².</p>	
Geeignet für Labortyp:	x	Unspezifisch (Variante 2 + 3)
		FabLab
	x	Smart Home Lab (Variante 1)
		Robotik Lab
Zielgruppe/Klassenstufe:		5. bis 7. Klasse
	X	8. bis 9. Klasse
	X	10. bis 11. Klasse
		12. bis 13. Klasse
mögliche Zahl an Teilnehmenden:	12	
Workshopleitende:	2-3 (bei 12 Teilnehmerinnen)	
Geschätzter Zeitaufwand:	20 Std. -> 4 - 5 Tage	
Lernziele:	<ul style="list-style-type: none"> ■ Grundlegende Programmierkonzepte; ■ Sensorik und Aktorik; ■ Additives RGB Farbmodell & Farbmischung; ■ Informatik als kreatives Fach. 	
Vorkenntnisse der Schülerinnen:	<ul style="list-style-type: none"> ■ Keine 	
Vorkenntnisse der/des Workshopleitenden:	<ul style="list-style-type: none"> ■ Grundlagen Arduino Programmierung: Neopixel LEDs, ggf. Sensoren; ■ Blockprogrammierung: BEESM für Variante 1 und 2, alternativ mBlock für Variante 3 möglich. 	
Voraussetzungen an die Infrastruktur:	<ul style="list-style-type: none"> ■ Laptops mit WLAN/Internet; ■ (An)Steuerung des Smart Homes über einen Wifi-fähigen Mikrocontroller (mit BEESM) über OpenHab für Variante 1. 	
Sonstige Voraussetzungen:	<ul style="list-style-type: none"> ■ Keine 	

² <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 3 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, die der Rechner der Reihe nach ausführt um bestimmte Aufgaben zu bearbeiten oder zu lösen.
- Die Teilnehmenden können zielgerichtet grundlegende Programmierkonzepte mit Hilfe einer Blockprogrammiersprache anwenden, das sind bedingte Anweisungen, Schleifen, Variablen.
- Die Teilnehmenden verstehen das Prinzip der additiven Farbmischung (RGB) und können zielgerichtet Farben aus den drei RGB Werten mischen.
- 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 Licht-Objekt herzustellen und zu programmieren, das dynamisch auf die Umgebung reagiert; wenn möglich wird dieses wie in Variante 1 in ein Smart Home eingebunden. Die TN können in dem Workshop eigene Ideen und Anwendungsszenarien dazu entwickeln, die somit möglichst an die eigene Lebenswelt anknüpfen.

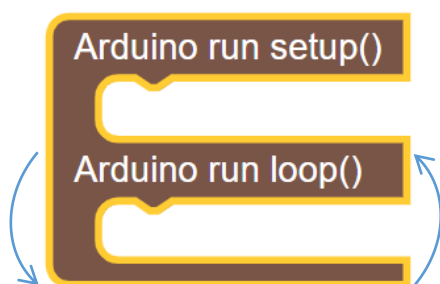
Das Programmieren der einzeln adressierbaren LEDs (RGB Neopixel, mit dem Chip WS2811/12(B)) ist gut für den Programmier-Einstieg geeignet: Mit einfachen Programmen können schnell ansprechende und attraktive Resultate erzielt werden. Die LEDs veranschaulichen Programmiergrundlagen, wie Ablauf und Schleifen unmittelbar visuell. Anhand des additiven Farbmodells kann ferner in das Bit-System eingeführt werden.

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 (bzw. einem anderen kompatiblen Mikrocontroller), einem passenden Display, einer oder mehrerer LEDs und einer Block-Programmierungsumgebung arbeiten können. Nach einer grundlegenden Einführung werden weitere Programmierkonzepte, wie Schleifen, Variablen und bedingte Anweisungen vermittelt und mit einer oder mehreren RGB-LEDs erprobt.

Diese Workshopbeschreibung bezieht sich auf die Block-Programmierungsumgebung BEESM, die vom Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)³ im Rahmen des smile Projekts entwickelt wurde. BEESM bietet neben der Arduino-Programmierung auch eine Block-Programmierungsumgebung für das Living Lab BAALL⁴ und den Roboter Turtlebot⁵, die in weiteren smile Workshops genutzt wird. Am Ende dieser Dokumentation befinden sich weitere Hinweise zu BEESM.

Im Folgenden werden die Blöcke gezeigt, so wie sie in 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-

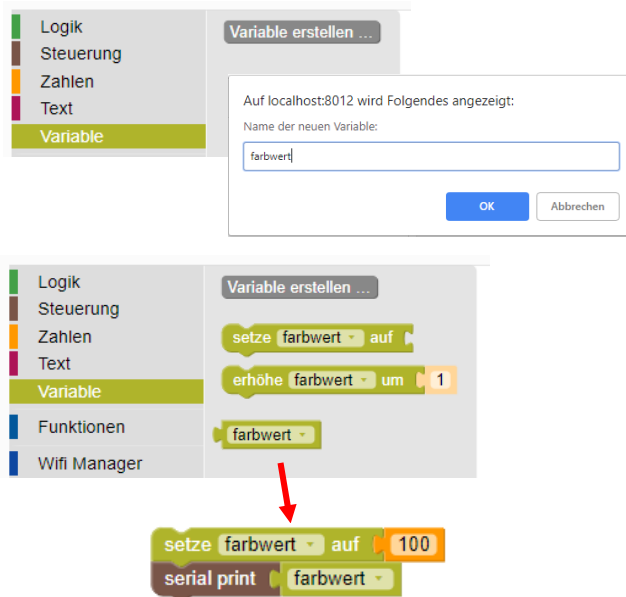
³ Forschungsbereich Cyber Physical Systems (CPS) Bremen

⁴ <https://www-cps.hb.dfki.de/research/baall>

⁵ <https://www.turtlebot.com/>

Sketch (oder ein „Hello World“ Sketch) im Setup, in der Loop sowie mit und ohne der `warte()` Funktion programmiert werden.

Eine **Variable** ist ein Platz, um Werte zu speichern, ähnlich wie ein Karton, in dem

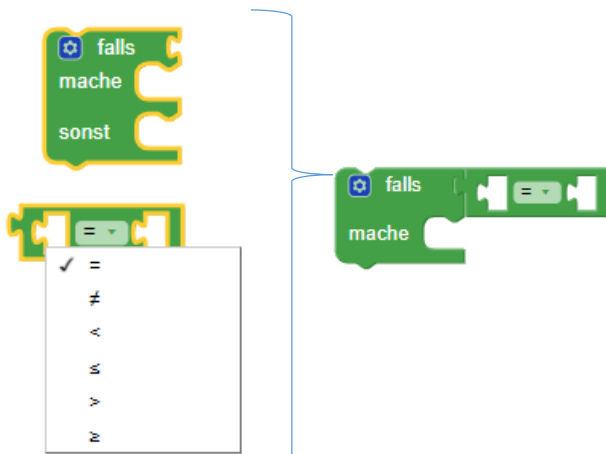


man etwas hineintun kann. Die Variable muss eindeutig mit einem Namen bezeichnet werden, damit der Rechner im Laufe der Programmausführung auf den Wert in der Variablen zugreifen kann. Der Wert einer Variable kann im Laufe der Programmausführung verändert werden.

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**

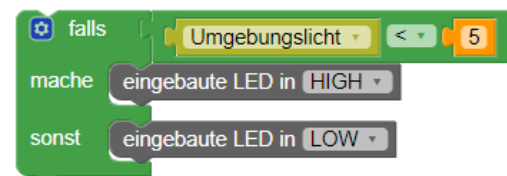


dies zutrifft, **dann** mache dies, **sonst** das“.

Bedingungen werden also benutzt, um eine Aktion von einer bestimmten Bedingung abhängig zu machen.

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) das Licht an (HIGH), **sonst** schalte das Licht aus (LOW). Der Licht-Wert (z.B. in Lux) ist in der Variable „*Umgebungslicht*“ gespeichert. Ist dieser Wert kleiner als 5, dann ist es dunkel.



In der Regel wird eine **Schleife** benutzt, wenn ein Programm-Teil wiederholt abgearbeitet werden soll. Der **For-Loop** kann ggf. mit Neopixel-Streifen (oder Ringen, Matrizen etc), bei denen mehrere LEDs hintereinandergeschaltet sind, vermittelt werden. Siehe Arbeitsblätter Material14/15.

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. Für die Angabe der Farbe stehen 24 Bits zur Verfügung. Daraus folgt, dass jeder dieser 3 Farbkanäle (r, g, b) in 8 Bit codiert ist und somit die Farbintensität jeweils zwischen 0 und 255 liegt. 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“. Auf diese Weise können über 16 Mio. Farben gemischt werden, um genau zu sein: $2^8 * 2^8 * 2^8 \rightarrow 256*256*256$ verschiedene Farben.

Durch das experimentelle Programmieren verschiedener Farben bekommen die TN ein Gefühl für das Mischen der Farben.

Sensorik/Aktorik

Smarte (intelligente) Anwendungen „handeln“ selbstständig und vorausschauend.

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 physikalische oder chemische Größen (Helligkeit, Druck, Temperatur etc.) in seiner Umgebung und wandelt diese Größe in ein elektrisches Signal um. Der Aktor (als „Gegenstück“ zum Sensor, lat. „agere“, → agieren, handeln, tun, wirken) verarbeitet das elektrische Signal weiter, indem er dieses in eine physikalische Größe umwandeln kann. Das kann zum Beispiel Bewegung durch einen Motor sein oder eine LED, aber auch Helligkeit und Temperatur sind physikalische Größen.

Welche Beispiele kennen die TN für Sensoren/Aktoren und smarte Systeme aus Ihrer Erfahrung?

Variationen

1.4 Variante 1 – Smartes Stimmungslicht im Smart Home

Phase	Kurze Zusammenfassung	Dauer: ca. 4 Tage (20 Std)
Einstieg	Motivation des Themas Einführung in Smart Home Technologien. Was kann smartes Licht bedeuten?	
Einstieg: Hallo Welt!	Einführung, und praktische Beispiele für die eingesetzten Technologien: <ul style="list-style-type: none">■ Arduino■ Block-Programmierungsumgebung■ Ein Programm■ Ein Display für den Arduino. Die TN schreiben ein erstes Programm (z.B. Hallo Welt) für den Arduino	
Vertiefung I: RGB-LED	Die TN programmieren eine RGB-LED und lernen das additive Farbmodell kennen.	
Vertiefung II: Sensorik/Aktorik & das BAALL	Das BAALL (Living Lab): Laborführung Die TN lernen das Prinzip Sensorik/Aktorik kennen Die TN lernen die Programmierkonzepte Variablen und Bedingungen anhand der Programmierung des BAALL.	
Vertiefung III: Schleifen	Die TN programmieren LEDs, die in Reihe geschaltet sind und/oder programmieren das ‚Faden‘ einer LED als Einstieg in das Thema Schleifen.	
Projektarbeit	Die TN setzen eigene Licht-Objekte praktisch um. Das beinhaltet das Konstruieren und Programmieren der Objekte.	

1.5 Variante 2 – Smartes Stimmungslicht, IoT Version

Um die Licht-Objekte unabhängig vom Smart Home zu erstellen, können die Anwendungen auf andere Daten zugreifen, wie im folgenden Beispiel auf die von der OpenWeatherMap⁶ bereitgestellten Wetterdaten.

Phase	Kurze Zusammenfassung	Dauer: ca. 4 Tage (20 Std)
Einstieg	Motivation des Themas Einführung in Smart Home Technologien. Was kann smartes Licht bedeuten?	
Einstieg: Hallo Welt!	Einführung, und praktische Beispiele für die eingesetzten Technologien: <ul style="list-style-type: none">■ Arduino■ Block-Programmierungsumgebung■ Ein Programm■ Ein Display für den Arduino. Die TN schreiben ein erstes Programm (z.B. Hallo Welt) für den Arduino	
Vertiefung I: RGB-LED	Die TN programmieren eine RGB-LED und lernen das additive Farbmodell kennen.	
Vertiefung II: Sensorik/Aktorik & OpenWeatherMap	Das BAALL (Living Lab): Laborführung (o.ä.) Die TN lernen das Prinzip Sensorik/Aktorik kennen Die TN lernen die OpenWeatherMap API kennen und vertiefen anhand dessen die Programmierkonzepte Variablen und Bedingungen.	
Vertiefung III: Schleifen	Die TN programmieren LEDs, die in Reihe geschaltet sind und/oder programmieren das ‚Faden‘ einer LED als Einstieg in das Thema Schleifen.	

⁶ <https://openweathermap.org/>

Projektarbeit	Die TN setzen eigene Licht-Objekte praktisch um. Das beinhaltet das Konstruieren und Programmieren der Objekte.
---------------	---

1.6 Variante 3 – Smartes Stimmungslicht offline

Die Anwendung des folgenden Beispiels reagiert auf die Werte des Licht-, PIR-, Temperatur-Sensors oder Buttons, die mit dem Mikrocontroller verbunden sind. Diese Variante kann auch mit einem nicht Wifi-fähigen Mikrocontroller, wie der Arduino Uno, durchgeführt werden.

Phase	Kurze Zusammenfassung	Dauer: ca. 4 Tage (20 Std)
Einstieg	Motivation des Themas Einführung in Smarte Technologien. Was kann smartes Licht bedeuten?	
Einstieg: Hallo Welt!	Einführung, und praktische Beispiele für die eingesetzten Technologien: <ul style="list-style-type: none"> ■ Arduino ■ Block-Programmierungsumgebung ■ Ein Programm ■ Ein Display für den Arduino. Die TN schreiben ein erstes Programm (z.B. Hallo Welt) für den Arduino	
Vertiefung I: RGB-LED	Die TN programmieren eine RGB-LED und lernen das additive Farbmodell kennen.	
Vertiefung II: Sensorik/Aktorik	Das BAALL (Living Lab): Laborführung (o.ä.) Die TN lernen das Prinzip Sensorik/Aktorik kennen. Die TN befassen sich explorativ mit ausgewählten Sensoren und vertiefen dabei die Programmierkonzepte Variablen und Bedingungen.	
Vertiefung III: Schleifen	Die TN programmieren LEDs, die in Reihe geschaltet sind und/oder programmieren das ‚Faden‘ einer LED als Einstieg in das Thema Schleifen.	
Projekte	Die TN setzen eigene Licht-Objekte praktisch um. Das beinhaltet das Konstruieren und Programmieren der Objekte.	

Materialliste

1.7 Benötigte Materialien

12 TN, für Gruppen zu je 2 TN:

Alle Varianten:

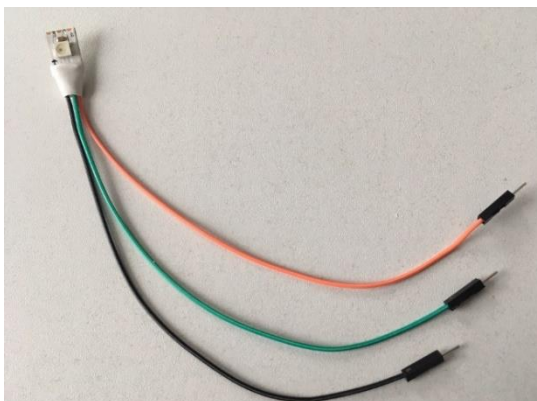
- Mindestens 6x Laptops/Computer
- Neopixel LEDs (WS2811/12(B)): 6 x einzelne LEDs (gelötet mit Kabel zum verbinden mit Mikrocontroller (siehe Abbildung 5, besser einige LEDs auf Vorrat, da diese schnell kaputt gehen). Diese können auch während des Workshops von den TN gelötet werden. Zusätzlich verschiedene Neopixel: Strips in verschiedener Länge, Matrizen, Ringe, Ketten etc.. für Projekte
- Mikrocontroller entsprechend der Variante mit USB-Kabel, Netzteil oder Powerbank
- Bastelmaterialien, evtl. vorgefertigte Arbeitsmaterialien zur Gestaltung der Licht-Objekte, z.b. Schmuck/"Lampenschirme", Käseschachteln, 3D-Bilderrahmen, Transparent-/Architektenpapier, Kästchen/5mm Acryglasstap (siehe smile Workshop "Der Blume geht ein Licht auf") Ping-Pong-Ball etc...

Zusätzlich Variante 1 & 2:

- 6 x Wifi-fähiger ESP-Mikrocontroller: Z.B. Wemos D1 mini + OLED Display Shield

Zusätzlich Variante 3:

- 6 x Arduino: Arduino/Seeeduino (mit Grove Shield/Pin zum Einstieg) + (Grove) RGB LC Display
- Verschiedene Sensoren (Licht, Farbe, Sound, Ultraschall, PIR, Temperatur, Button, Potentiometer...), passende Widerstände, Steckkabel etc.



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

1.8 Lern-Materialien

Nr.	Titel	Beschreibung	Benötigt für Variante(n)
_02	PowerpointStimmungslicht _esp8266, _arduino	PowerPoint Präsentation, Folien begleitend zum Workshop V1/V2 (_esp8266) oder V3 (_arduino)	1, 2, 3
_03	Handout NeopixelBlöcke	Übersicht über Blöcke in BEESM, mit denen Neopixel LEDs programmiert werden	1, 2, 3
_04	Übungen LED	LED programmieren und das additive Farbmodell: Erklärungen und kleine Aufgaben. Arbeitsblätter geeignet für individuelle Gruppenarbeit oder Bearbeitung in Gesamtrunde.	1, 2, 3
_05	Übungen BAALL	Sensorik/Aktorik und das Living Lab BAALL: Erklärungen und kleine Aufgaben mit Variablen und Bedingungen. Arbeitsblätter geeignet für individuelle Gruppenarbeit oder Bearbeitung in Gesamtrunde.	1
_06	Handout BAALL	Übersicht zum Austeilen an die TN. Liste, die die Objekte des Living Labs BAALL zeigt.	1
_07	Challenges BAALL	Verschiedene Projektideen: Kleine smarte Systeme mit LEDs und Sensoren aus dem Living Lab. Beispiele, geschrieben auf Mod. Kärtchen	1
_08	Übungen Wetterdaten	Sensorik/Aktorik, die OpenWeatherMap API: Erklärungen und kleine Aufgaben mit Variablen und Bedingungen. Arbeitsblätter geeignet für individuelle Gruppenarbeit oder Bearbeitung in Gesamtrunde.	2
_09	Handout WetterBlöcke	Übersicht zum Austeilen an die TN: Blöcke in BEESM, mit denen auf Wetterinformationen zugegriffen wird.	2
_10	Challenges Wetter	Verschiedene prakt Aufgaben mit Wetterdaten und LEDs. Beispiele, geschrieben auf Mod. Kärtchen	2
_11	Übungen Sensoren	Sensorik/Aktorik und wie mit Sensoren Werte ausgelesen werden. Variablen und Bedingungen. Zum Austeilen für individuelle Gruppenarbeit.	3
_12	Erklärkarten Sensoren	Sensoren, die auf handlichen Karten angebracht sind und anzeigen, wie diese an den Controller angeschlossen werden. Beispiele	3
_13	Challenges Sensor	Aufgaben, um verschiedene Sensoren explorativ zu erarbeiten mit kleinen Projekten. Beispiele	3
_14	Arbeitsblatt Schleife LED Fade	Einführung in die Programmierung von Schleifen anhand des Themas „Fading“ einer LED	1, 2, 3
_15	Arbeitsblatt Schleife LED Kette	Einführung in die Programmierung von Schleifen anhand des Themas LED-Ketten	1, 2, 3
_16	Handout Flussdiagramm	Symbole und Flussdiagramm Beispiel	1, 2, 3
_17	Anleitung Wifi Manager	Anleitung, wie das Licht-Objekt mit dem Wifi-Manager eine Netzwerk-Verbindung aufbauen kann. Auch für zu Hause für die TN.	2

Verlaufspläne

1.9 Variante 1

Einstieg; Programmierung Hallo Welt (~ 2,5 Stunden)

Zeit	Phase	Inhalt	Material:
60 Min.		<i>Es gibt 60 min. Zeit zum Kennenlernen, Pre-Fragebogen, Interviews sowie Formalitäten</i>	02_Stimmungslicht, Folien 1 - 17
15 Min.	Einstieg	<p>Die Teilnehmerinnen (TN) beschäftigen sich in 2er Gruppen mit einer oder mehreren Fragen (Flüsterrunde):</p> <ul style="list-style-type: none"> ■ Was bedeutet Smart Home für sie? ■ Welche nützlichen intelligenten Gegenstände für den Alltag gibt es, die sie gerne hätten? ■ Was kann ein Smartes Stimmungslicht sein? <p>Die TN machen sich Notizen. Anschließend werden Ideen zusammengetragen. Genannte Stichworte wie Vernetzung/Konnektivität, Komfort, Clever/intelligent, selbstständiges „handeln“, Automatisierung, Steuerbarkeit, etc. werden vom KL eingeordnet und erläutert.</p>	<p>Evtl. zum Vergleich: Was sagen Elektronik Azubis dazu? https://www.youtube.com/watch?v=cta6B2U6Pu0</p>
10 Min.	Hinführung	Technikeinführung: Mikrocontroller und Display verteilen, anschließen und zusammenbauen	<p>Wemos d1 mini + OLED-Shield Alternativ: Seeeduino, (Arduino Uno) + Grove RGB LCD & passendes USB Kabel</p>
60 Min.	Erarbeitung	<p>Einführung in die Programmierung mit Aufgaben:</p> <ul style="list-style-type: none"> ■ Die Programmierumgebung (BEESM) wird vorgestellt und wird von den TN ausprobiert ■ Einführung Arduino Programmstruktur: Setup und Loop ■ Blöcke (Programmieranweisungen) für Ausgabe auf Display ■ Aufgabe: Ein erstes Programm: Ein Text soll ausgegeben werden („Hallo Welt“) ■ Die TN lernen experimentell das Display kennen. 	
10 Min.	Sicherung	<p>Diskussion und Zusammenfassung:</p> <ul style="list-style-type: none"> ■ Display: Wieviel Text/Zeilen/Spalten passt darauf? ■ Ausgabe im Setup vs. Ausgabe in der Loop 	

Vertiefung I; Programmierung RGB LED (~ 2,5 Stunden)

Zeit	Phase	Inhalt	Material
			02_Stimmungslicht, Folien 18-26
10 Min.	Einstieg	Neopixel LEDs werden vorgestellt, verschiedene „Formen“ (Strips, Ringe, Matrizen, ...) werden ggf. herumgegeben. LED Animationen und oder Lampen können zur Motivation gezeigt werden, aber ohne zu viel vorwegzunehmen.	LED Pixel und LED Strips/Ringe Beispiellampen als Anschauungsobjekte
10 Min.	Hinführung	Technikeinführung: LED, Mikrocontroller verteilen und anschließen. Die KL helfen und kontrollieren.	
10 Min.	Hinführung	Blöcke (Programmieranweisungen) um LEDs zu schalten werden vom KL erläutert.	03_Handout NeopixelBlöcke
90 Min.	Erarbeitung	<p>Ein erstes LED Programm soll auf der Basis des Handouts von den TN selbstständig geschrieben werden: Eine LED soll in einer beliebigen Farbe leuchten.</p> <p>Achtung: Die LED sollte zunächst mit einem weißen (Transparent-) Papier abgedeckt werden, da diese ansonsten sehr hell in die Augen scheint.</p> <p>Die TN erarbeiten das additive Farbmodell anhand des Übungsblattes:</p> <ul style="list-style-type: none"> Arbeitsauftrag: Trage die Zahlentripel der Haupt- und grundlegenden Mischfarben auf dem Arbeitsblatt ein, bestimme experimentell die Zahlentripel der Farben auf den Farbkarten <p>Einführung in die Neopixel-Programmierung mit Übungsaufgaben:</p> <ul style="list-style-type: none"> einzelnes Pixel einfärben Farbintensität/Helligkeit variieren Einzelnes Pixel soll blinken Einzelnes Pixel soll faden <p>Raum für eigene Ideen/Experimente soll gegeben werden</p> <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>	04_Übungen LED
10 Min.	Sicherung	Ausgewählte Lösungen aus den Aufgaben werden gemeinsam besprochen	
20 Min.	Sicherung	Die Ergebnisse der kreativen Herausforderung werden von den TN in der Gesamtrunde vorgestellt	

Vertiefung II; Programmierung Das BAALL (~ 4,5 Stunden)

Zeit	Phase	Inhalt	Material
			05_Übungen BAALL
10 Min.	Einstieg	Einführung in das Thema Aktoren und Sensoren.	
50 Min.	Laborführung	Führung im Living Lab (BAALL): Das Smart Home wird vorgestellt und möglichst praktisch erprobt	
20 Min.	Hinführung	Die TN erfassen die items des Smart Homes wie Lichter, Türen, Bett, Außensensoren etc. und ordnen diese ggf. der Sensorik und/oder Aktorik zu. <ul style="list-style-type: none"> ■ Welche Sensoren / Aktoren gibt es im BAALL? ■ Wie wirken diese zusammen? ■ Welche Werte können diese Annehmen? ■ Welchen Kategorien können diese zugeordnet werden? 	06_Handout BAALL
60 Min.	Erarbeitung	Weiterführende Programmierkonzepte: Variablen. Der KL erklärt das Konzept der Variablen und verschiedene Übungen dazu werden bearbeitet. Ein erstes „BAALL Programm“ soll auf der Basis des Übungsblattes (S. 16/17) von den TN selbstständig geschrieben werden: Hole den aktuellen Status und gebe diesen auf dem Display aus.	
10 Min.	Sicherung	Ausgewählte Aufgaben des Übungsblattes werden gemeinsam besprochen.	
70 Min.	Erarbeitung	Weiterführende Programmierkonzepte: Bedingungen. Zur Anwendung der Konzepte arbeiten die TN (in Gruppen) selbstständig und bekommen jeweils eine Herausforderung, die individuell umgesetzt werden soll. Beispiele: <ul style="list-style-type: none"> ■ Temperaturlampe (System reagiert auf Temperatur Sensoren, oder auf Wind- / Helligkeit) ■ Emotionslampe (System reagiert auf Gesichts- Emotionserkennung) ■ Nachtschlampe (System reagiert auf andere Lampen im Smart Home) 	7_Challenges BAALL (ein paar Beispiele...)
40 Min.	Sicherung	Ergebnisse der Challenge werden von den Gruppen präsentiert.	
10 Min.	Sicherung	Gibt es erste Ideen zur Umsetzung der smarten Lampen? Ideen werden besprochen und gesammelt.	

Vertiefung III; Programmierung Schleifen (~ 2,5 Stunden)

Zeit	Phase	Inhalt	Material
			02_Stimmungslicht, Folien 26-33
10 Min.	Einstieg	Fading Effekte oder lange animierte LED Streifen werden zur Motivation gezeigt.	Bsp. animierte LED Strips/Ringe
25 Min.	Hinführung	Wiederholung: Programmiert einen Fading-Effekt (die LED soll langsam angehen und evtl. danach wieder ausgehen) Das Konzept der Schleife soll zunächst noch nicht vorweggenommen werden...	
20 Min.	Hinführung	Lösungsansätze werden besprochen. Das Konzept der Schleifen in der Programmierung wird vom KL eingeführt.	
75 Min.	Erarbeitung	Die TN (in Gruppen) bearbeiten selbstständig verschiedene Programmierübungen zu dem Thema For-Schleife, wie auf den Arbeitsblättern erläutert.	14_Arbeitsblatt Schleife LED Fade 15_Arbeitsblatt Schleife LED Kette
20 Min.	Sicherung	Ergebnisse werden in Gesamtrunde besprochen.	

Umsetzung der Projekte (~ 8 Stunden)

Zeit	Phase	Inhalt	Material
30 Min.	Einstieg	In der Gesamtrunde werden Ideen der TN besprochen und die Umsetzung in den Schritten Problem, Planung, Entwicklung, Präsentation geplant.	
30 Min.	Erarbeitung	Die TN verschriftlichen Ihren Plan zur Umsetzung mit Hilfe von Flussdiagrammen, Struktogramm o.ä. (abhängig vom Alter und Erfahrung der TN)	Schablonen für Diagramme, z.B. 16_Handout Flussdiagramm
4 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 in der Gesamtrunde präsentiert.	
60 Min.		<i>Anmerkung: Es gibt 60 min. Zeit für Abschluss, Post-Fragebogen sowie Interviews, ggf. Role Models</i>	

1.10 Variante 2

Die Verlaufspläne der Einheiten Einstieg, Vertiefung I, Vertiefung III, und Projekte sind identisch mit Variante 1. Die Varianten unterscheiden sich in den genutzten Sensoren, die in Vertiefung II eingeführt werden.

Vertiefung II; Programmierung Aktoren/Sensoren, Beispiel Wetterdaten (~ 4,5 Std)

Zeit	Phase	Inhalt	Material
			08_Übungen_ Wetterdaten
10 Min.	Einstieg	Einführung in das Thema Aktoren und Sensoren.	
50 Min.	Laborführung	Führung im Living Lab (BAALL): Das Smart Home wird vorgestellt (Oder ein anderer praktischer Bezug zum Thema sollte hergestellt werden)	
10 Min.	Hinführung	<p>Optional: Die TN ordnen die items des Smart Homes wie Lichter, Türen, Bett, Außensensoren etc. der Sensorik und/oder Aktorik zu.</p> <ul style="list-style-type: none">Welche Sensoren / Aktoren gibt es im BAALL?Wie wirken diese zusammen? <p>Der KL verengt auf Wetter-Sensoren (z.B. über das Thema Kleiderschrank (im BAALL), der Wetterdaten abrufen, anzeigt und auf Basis dieser Daten ein Outfit vorschlägt...)</p> <ul style="list-style-type: none">Beispiele für Sensoren, die Wetterdaten liefern?Welche Werte können diese annehmen?	
10 Min.	Hinführung	<p>Die Blöcke (Programmieranweisungen) um auf die Wetterdaten zuzugreifen und diese auszulesen stellt der KL in der Gesamtrunde vor.</p> <p>Die OpenWeatherMap API wird vorgestellt. Der KL erklärt beispielhaft wie das Abrufen und Verarbeiten der Wetterdaten funktioniert, also das, was sich hinter den Blöcken verbirgt. Beispiel Wetterdaten als JSON String.</p> <p>https://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22</p>	09_Handout WetterBlöcke https://openweathermap.org
60 Min.	Erarbeitung	Weiterführende Programmierkonzepte: Variablen. Der KL erklärt das Konzept der Variablen und verschiedene Übungen dazu werden in der Gesamtgruppe bearbeitet.	

		Ein erstes „Wetter Programm“ soll auf der Basis des Handouts von den TN selbstständig geschrieben werden: Die Temperatur soll ausgelesen und angezeigt werden.	
10 Min.	Sicherung	Ausgewählte Aufgaben des Übungsblattes werden besprochen.	
70 Min.	Erarbeitung	Weiterführende Programmierkonzepte: Bedingungen Zur Anwendung der Konzepte arbeiten die TN (in Gruppen) selbstständig und bekommen jeweils eine unterschiedliche Herausforderung, die individuell umgesetzt werden soll. Beispiele: <ul style="list-style-type: none"> ■ Regenlampe (System zeigt an, ob es regnet.) ■ UV-Index (System zeigt, ob UV-Schutz nötig ist.) Optional: Die TN dokumentieren die Ergebnisse und bereiten diese auf (z.B. als Poster)	10_ Challenges Wetter (ein paar Beispiele...)
40 Min.	Sicherung	Die TN präsentieren die Ergebnisse der Challenge.	
10 Min.	Sicherung	Gibt es erste Ideen zur Umsetzung der smarten Lampen? Ideen werden besprochen und gesammelt.	

1.11 Variante 3

Die Verlaufspläne der Einheiten Einstieg, Vertiefung I, Vertiefung III und Projekte sind identisch zu Variante 1.

Vertiefung II; Programmierung Aktoren/Sensoren (~ 4,5 Std)

Zeit	Phase	Inhalt	Material
			11_Übungen_Sensoren
10 Min.	Einstieg	Einführung in das Thema Aktoren und Sensoren.	
50 Min.	Laborführung	Führung im Living Lab (BAALL): Das Smart Home wird vorgestellt. (Oder ein anderer praktischer Bezug zum Thema sollte hergestellt werden)	
10 Min.	Hinführung	Optional: Die TN ordnen die items des Smart Homes wie Lichter, Türen, Bett, Außensensoren etc. der Sensorik und/oder Aktorik zu. <ul style="list-style-type: none"> ■ Welche Sensoren / Aktoren gibt es im BAALL? ■ Wie wirken diese zusammen? ■ Welche Sensoren kennen die TN noch aus ihrem Alltag? 	

40 Min.	Erarbeitung	<p>Weiterführende Programmierkonzepte: Variablen. Der KL erklärt das Konzept der Variablen und verschiedene Übungen dazu werden in der Gesamtgruppe bearbeitet.</p> <p>Ein erstes „Sensor Programm“ soll von den TN selbstständig geschrieben werden: Die Werte an einem Pin sollen ausgelesen und angezeigt werden (Ergibt ohne Sensor zunächst 0).</p>	
10 Min.	Sicherung	Aufgaben/Lösungen des Übungsblattes werden besprochen.	
100 Min.	Erarbeitung	<p>Weiterführende Programmierkonzepte: Bedingungen</p> <p>Die TN in Gruppen bearbeiten jeweils 2 Sensoren (z.B. Ultraschall, Licht, Temperatur, PIR Potentiometer, Schalter, ...), um diese experimentell zu bestimmen.</p> <p>Dazu lösen die Gruppen selbstständig durch Recherche verschiedene Aufgaben zu ihren Sensoren. Die KL unterstützen die einzelnen Gruppen.</p> <ul style="list-style-type: none"> ■ Wie funktioniert der Sensor? ■ Wie wird er angeschlossen? ■ Wie wird er programmiert? ■ Anwendung bekannt aus dem Alltag? ■ Was könnte eine sinnvolle Anwendung des Sensors/der beiden Sensoren sein? ■ Challenge: Schreibe ein Programm, das beide Sensoren verbindet. 	<p>12_Erklärkarten Sensoren (ein paar Beispiele...)</p> <p>13_Challenges Sensoren (ein paar Beispiele...)</p>
40 Min.	Sicherung	Die Teilnehmerinnen präsentieren die Ergebnisse der Challenge.	
10 Min.	Sicherung	Gibt es erste Ideen zur Umsetzung der smarten Lampen? Ideen werden besprochen und gesammelt.	

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. Zur Entwicklung von BEESM nutzen wir die Google-Blockly-Bibliothek⁷. 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“

Für die Varianten 1 und 2 benötigen wir ab Vertiefung II die Wifi Fähigkeiten des ESP8266 Moduls um auf den Smart Home Server bzw. das Internet (z.B. openweathermap.org) zugreifen zu können. In den Workshops haben wir den kompakten und günstigen Wemos D1 mini genutzt.

Es hat sich herausgestellt, dass der Arduino Uno (wir verwenden den Seeeduno mit Grove Steckplatz), den wir in Variante 3 benutzen, zuverlässiger und schneller arbeitet als der Wemos Controller, was die TN ermutigt, befreiter Ideen auszuprobieren und insgesamt im Workshop sehr viel mehr Spaß bringt. Auch führt das zum Arduino zugehörige RGB LC Display zu attraktiveren Resultaten als das OLED Display, das wir als Shield mit dem Wemos D1 mini verwenden.

Wir empfehlen daher auch in den ersten beiden Modulen (Einführung und RGB LEDs) der Varianten 1 und 2 mit einem Arduino zu arbeiten und erst später, wenn Wifi benötigt wird, auf den Wemos D1 mini umzusteigen. Die begleitenden Powerpoint Folien (Material_02) können beliebig ausgetauscht werden.

Es hat sich bestätigt, dass für die TN die Umsetzung eigener Ideen besonders motivierend ist. Das sind die Phasen freier Projektarbeiten in diesem Workshop, wie die hier als „Challenges“ bezeichneten Übungsprojekte zwischendurch und natürlich die „großen“ eigenen Projekte am Ende des Workshops. Allerdings ist der Betreuungsaufwand der freien Projekte sehr hoch, da sowohl die Anwendung/eigene Programmierung als auch die Lampen individuell gestaltet werden können. Wir empfehlen 3 Kursleiter für eine Gruppe von 12 TN, damit keine allzu langen und frustrierenden Leerlaufzeiten für die TN entstehen, die individuell betreut werden müssen. Kann diese Betreuung nicht gewährleistet werden, sollten die möglichen Projekt-Themen eingeschränkt oder vorgegeben werden.

Anknüpfungspunkte mit anderen Workshops

⁷ <https://developers.google.com/blockly/>