

smile

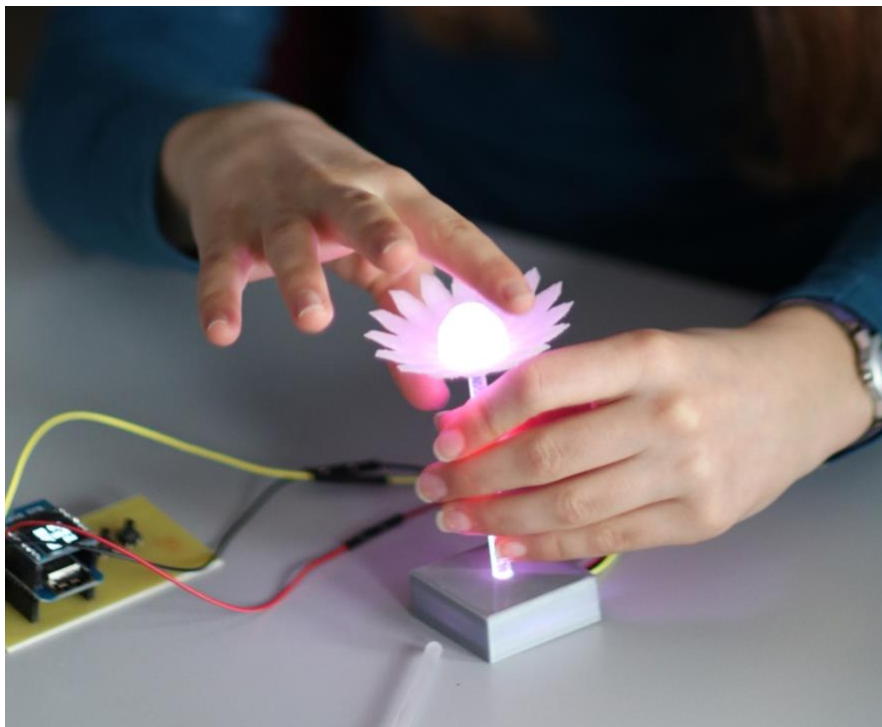
IT

SMART · FUTURE · ME

GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung



## Beschreibung des Workshops: **Der Blume geht ein Licht auf!**

# Inhalt

1	Kurzzusammenfassung .....	3
2	Ziele.....	5
2.1	Affektive Lernziele .....	5
2.2	Kognitive Lernziele .....	5
2.3	Psychomotorische Lernziele .....	6
3	Lerninhalte .....	6
3.1	Parametrisches Design .....	6
3.2	Zusammensetzen der Lampe .....	9
3.3	Physical Computing.....	9
3.4	Openweathermap-API.....	12
4	Variationen .....	13
4.1	Variante 1 – Kompletter Workshop vom Entwurf bis zur Präsentation.....	13
4.2	Variante 2 – Erstellen einer leuchtenden Blüte .....	13
4.3	Variante 3 – Erstellen einer eigenen Blüte mittels <i>BlocksCAD</i> .....	13
5	Materialliste .....	14
5.1	Benötigte Materialien.....	14
5.2	Lern-Materialien.....	14
6	Verlaufspläne .....	15
6.1	Einstieg in parametrisches Design .....	15
6.2	Vertiefung in <i>BlocksCAD</i> .....	16
6.3	Einarbeitung in Physical Computing .....	17
6.4	Sicherung der Ergebnisse.....	18
7	„Lessons learnt“ .....	19

Gestaltung und Konzeption dieses Workshops:



Carl von Ossietzky Universität Oldenburg  
Abteilung Didaktik der Informatik

# 1 Kurzzusammenfassung

In diesem Workshop wird eine smarte Wetter-Blume entwickelt. Dazu entwerfen die Teilnehmenden das Modell einer Blume, das anschließend mit einem 3D-Drucker gedruckt wird. Dann programmieren die Teilnehmenden Lichteffekte, die mithilfe einer API wetterabhängig auf einer LED in der Blume angezeigt werden. Der Workshop kann vollständig an drei Tagen durchgeführt werden. Am Ende haben die Teilnehmenden dann alle Bestandteile der Blume selbst entwickelt und können diese mit nach Hause nehmen. Es besteht auch die Möglichkeit, den Workshop auf einen oder zwei Tage zu beschränken und somit beispielsweise nur das Konzept des parametrischen Designs zu behandeln (siehe hierfür die verschiedenen Varianten). Im kompletten Workshop werden die Teilnehmenden am ersten Tag anhand eines Tutorials mit dem Programm *BlocksCAD* vertraut gemacht und sind danach in der Lage, eine Blüte nach ihren Vorstellungen mittels parametrischem Design zu erstellen. Am zweiten Workshop-Tag bekommen die Teilnehmenden einen Einstieg in Physical Computing, um einen Mikrocontroller zu programmieren, damit die LED verschiedene Farben anzeigen kann. Am dritten Tag werden die erstellten Elemente zusammengesetzt, sodass eine Blüte entsteht, die abhängig vom Wetter den jeweils programmierten Farbeffekt anzeigt. Diese Blume wird anschließend von den Teilnehmerinnen durch das Erstellen eines Videos oder eines Posters präsentiert.

<b>Verwendete Technologien:</b>	Parametric Design ( <i>BlocksCAD</i> ), evtl. 3D-Druck	
<b>Geeignet für Labortyp:</b>	<input checked="" type="checkbox"/>	unspezifisch
	<input type="checkbox"/>	FabLab
	<input type="checkbox"/>	Smart Home Lab
	<input type="checkbox"/>	Robotik Lab
<b>Zielgruppe/Klassenstufe:</b>	<input checked="" type="checkbox"/>	5. bis 7. Klasse
	<input checked="" type="checkbox"/>	8. bis 9. Klasse
	<input checked="" type="checkbox"/>	10. bis 11. Klasse
	<input type="checkbox"/>	12. bis 13. Klasse
<b>mögliche Zahl an Teilnehmenden:</b>	10 - 15	
<b>Workshopleitende:</b>	2 - 3	
<b>Geschätzter Zeitaufwand:</b>	6 – 18 Stunden (je nach Variante)	
<b>Lernziele:</b>	<ul style="list-style-type: none"><li>Die Teilnehmenden können nach diesem Workshop mit <i>BlocksCAD</i> Formen im Sinne des parametrischen Designs nach ihren Vorstellungen erstellen.</li><li>Die Teilnehmenden kennen grundlegende Konzepte des Physical Computing und entwickeln einen blockbasierten Programmcode für einen Mikrocontroller, um auf einer LED Farbeffekte anzeigen zu lassen.</li></ul>	
<b>Vorkenntnisse der Teilnehmenden:</b>	<ul style="list-style-type: none"><li>Keine</li></ul>	

<b>Vorkenntnisse der Workshopleitenden:</b>	<ul style="list-style-type: none"> <li>• Keine</li> </ul>
<b>Voraussetzungen an die Infrastruktur:</b>	<ul style="list-style-type: none"> <li>• Einen Computer/Laptop mit Internetzugang pro Teilnehmenden</li> <li>• Evtl. Drucker</li> <li>• 3D-Drucker bzw. die Möglichkeit, 3D-Drucker zu nutzen</li> </ul>
<b>Sonstige Voraussetzungen:</b>	<ul style="list-style-type: none"> <li>• Keine</li> </ul>

## 2 Ziele

### 2.1 Affektive Lernziele

- Die Teilnehmenden entdecken Informatik als kreativen, gestalterischen und kommunikativen Prozess.
- Die Teilnehmenden stellen eine positive emotionale Verbindung mit der Informatik her.
- Die Teilnehmenden haben ein erhöhtes Selbstbewusstsein in Bezug auf IT.

### 2.2 Kognitive Lernziele

*Zu parametrischem Design:*

- Die Teilnehmenden kennen parametrisches Design und wissen, wofür man es einsetzen kann.
- Die Teilnehmenden können mit BlocksCAD eigene Formen mit parametrischem Design erstellen.
- Die Teilnehmenden kennen Schleifen und Variablen und können diese beim Programmieren anwenden
- Die Teilnehmenden kennen das Grundprinzip von Mengenoperationen, wie sie bei parametrischem Design angewendet werden.
- Die Teilnehmenden kennen die Grundlagen des Lötens.

*Zu Physical Computing:*

- Die Teilnehmenden vertiefen Programmierkonzepte, indem sie mit Schleifen, Variablen und Mengenoperationen arbeiten.
- Die Teilnehmenden lernen Mikrocontroller als Informatiksysteme kennen, die Eingabeimpulse in Form von Daten verarbeiten und in veränderter Form ausgeben.
- Die Teilnehmenden kennen das additive Farbsystem und können damit eigene Farben mischen.

## 2.3 Psychomotorische Lernziele

- Die Teilnehmenden entwickeln notwendige feinmotorische Fähigkeiten im Umgang mit kleinen Lötarbeiten.
- Die Teilnehmenden sind in der Lage, kleine Lötverbindungen mit dem LötKolben zu erstellen.

## 3 Lerninhalte

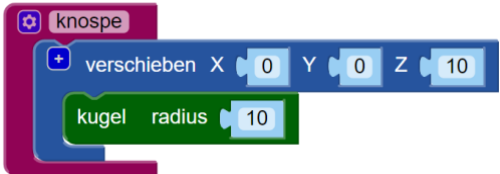
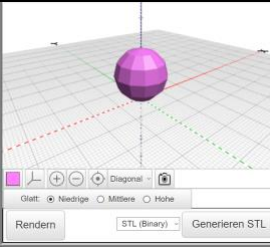
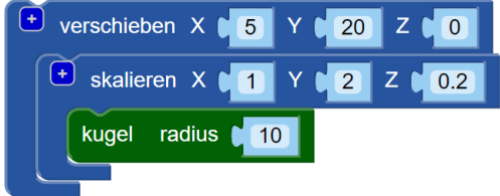
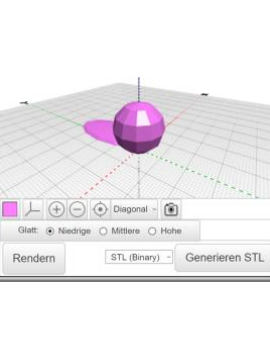
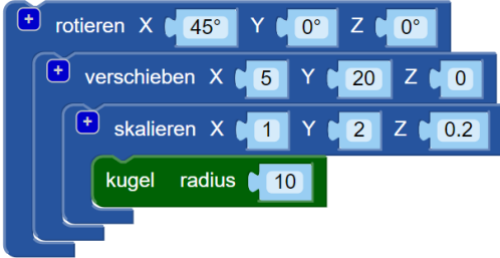
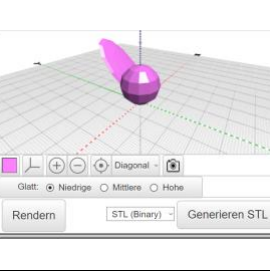
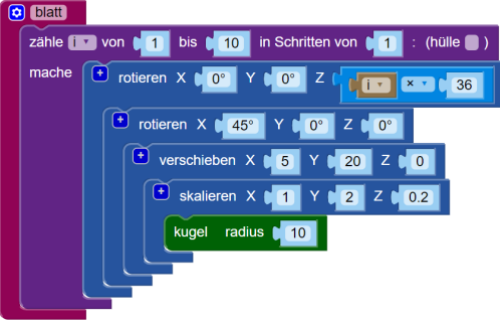
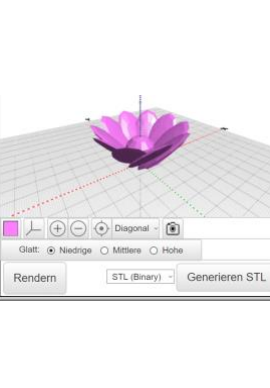
### 3.1 Parametrisches Design

Regelmäßige Formen und Muster treten in der Natur häufig auf, beispielsweise in Eiskristallen, Blumen und Bienenwaben. Aber auch in moderner Architektur und in der Kunst werden oft regelmäßige Formen verwendet. Diese regelmäßigen Formen lassen sich mit parametrischem Design herstellen. Das Merkmal von parametrischem Design ist, dass seine Form durch eine Kombination von Parametern, also variablen Größen, erzeugt wird. So können Abhängigkeiten, Wiederholungen und Beziehungen zwischen ihnen hergestellt werden. Durch eine Veränderung dieser Abhängigkeiten entstehen dann verschiedene Formen.

Die Erstellung von parametrischen Formen kann mit CAD-Systemen (aus dem Englischen: computer-aided design) vorgenommen werden. Die damit erstellten Modelle können anschließend mit einem 3D-Drucker ausgedruckt werden. In dem im Workshop verwendeten CAD-System *BlocksCAD* können diese Formen durch verschiedene Blöcke erstellt werden.

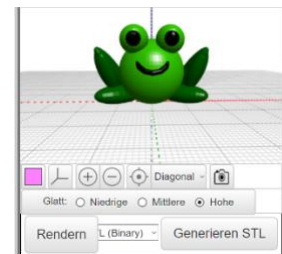
In dem Workshop wird das frei online verfügbare CAD-System [BlocksCAD](#) verwendet. Ein Beispielcode für eine entworfene Blume ist im GIT-Repository hinterlegt. Die xml-Datei kann auf dem PC gespeichert werden und dann in BlocksCAD über Projekte von dem PC in die Programmieroberfläche hochgeladen werden. Exemplarisch wird hier der Code für eine Blume erklärt. Eine ähnliche Blume erhalten die Teilnehmenden nach Durcharbeitung des Arbeitsblattes (Material\_04) und die dazugehörigen Hilfekarten (Material\_05). Die Teilnehmenden können auf Grundlage der Kompetenzen, die sie mit diesem Arbeitsblatt erworben haben, anschließend auch andere Formen für ihre Wetterlampe entwerfen.


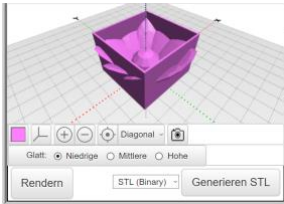

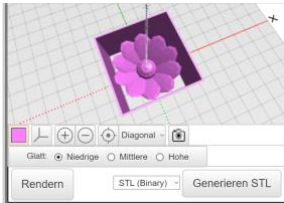

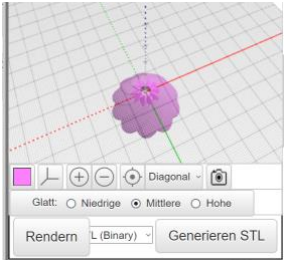
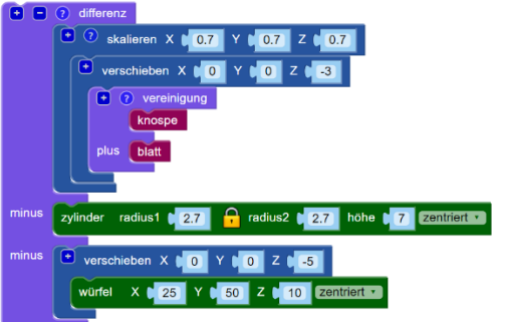
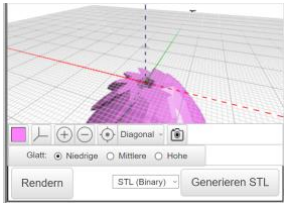
Die Blüte besteht aus zwei Hauptkomponenten: Knospe und Blatt, wobei nur ein Blatt modelliert wird und dieses dann mithilfe einer Schleifenoperation vervielfältigt und rotiert wird. Der Code wird von innen nach außen gelesen. Die Längeneinheiten, die in die Blöcke eingetragen werden, sind in Millimetern. Eine Längeneinheit entspricht also einem Millimeter.

	<p>Die Knospe besteht aus einer Kugel, die auf der Z-Achse verschoben wurde.</p>	
	<p>Für das Blatt wird zunächst eine Kugel erstellt, die anschließend mit den Blöcken <i>skalieren</i> und <i>verschieben</i> zu einem Blütenblatt gestreckt wird.</p>	
	<p>Mit dem <i>rotieren</i>-Block wird das Blatt schließlich in den richtigen Winkel zur X-Achse gestellt.</p>	
	<p>Mit einer Zählschleife wird das modellierte Blatt dann vervielfältigt. Es entstehen insgesamt 10 Blütenblätter, die um die Knospe bzw. den Ursprung des Koordinatensystems rotiert werden.</p>	

Denkbar sind anstelle einer Blume auch andere Formen wie beispielsweise ein Wetterfrosch oder ein Regenschirm.

Bevor das entworfene Modell nun in den 3D-Druck gegeben werden kann, müssen noch wenige Änderungen vorgenommen werden. Die folgenden Schritte werden auch durchgeführt, wenn die Teilnehmenden eine andere Form als eine Blume entworfen haben.



	<p>Das Modell sollte eine Größe von 4x4x4 nicht überschreiten. Zur Kontrolle kann in BlocksCAD ein Kasten erzeugt werden. Alternativ können die Längeneinheiten auf dem Koordinatensystem abgelesen werden.</p>	
	<p>Die Komponenten des Modells wurden vereint. Dann kann das Modell mit dem <i>skalieren</i>-Block angepasst werden. Um die Skalierung auszugleichen, wird auf der Z-Achse mit einem verschieben-Block gearbeitet.</p>	
	<p>Um den Acrylglasstab einzufügen, wird eine Aussparung im Modell benötigt. Dies geschieht mit dem <i>zylinder</i>-Block (Radius 2.6mm, Höhe min. 7mm) in einem differenz-Block.</p>	
	<p>Es ist möglich, am Modell Korrekturen vorzunehmen, indem bspw. mit einem <i>würfel</i>-Block in einem <i>differenz</i>-Block der untere Teil des Modells abgeschnitten wird.</p>	
<p>Um das Modell anschließend auszudrucken, sollte ein Rendern auf hoher Qualität durchgeführt werden. Anschließend kann die STL-Datei generiert werden. Um den BlocksCAD-Code zu speichern, kann zusätzlich das Projekt auf den PC heruntergeladen werden.</p>		



### 3.2 Zusammensetzen der Lampe

Das Zusammensetzen der benötigten Materialien wird in der Kurzanleitung (Material\_06) für die Teilnehmenden erläutert. Benötigt werden dafür für die jeweiligen Teilnehmenden

- ein Mikrocontroller ESP8266
- ein Display
- ein Steckbrett
- drei Kabel
- ein Acrylglasstab
- eine LED

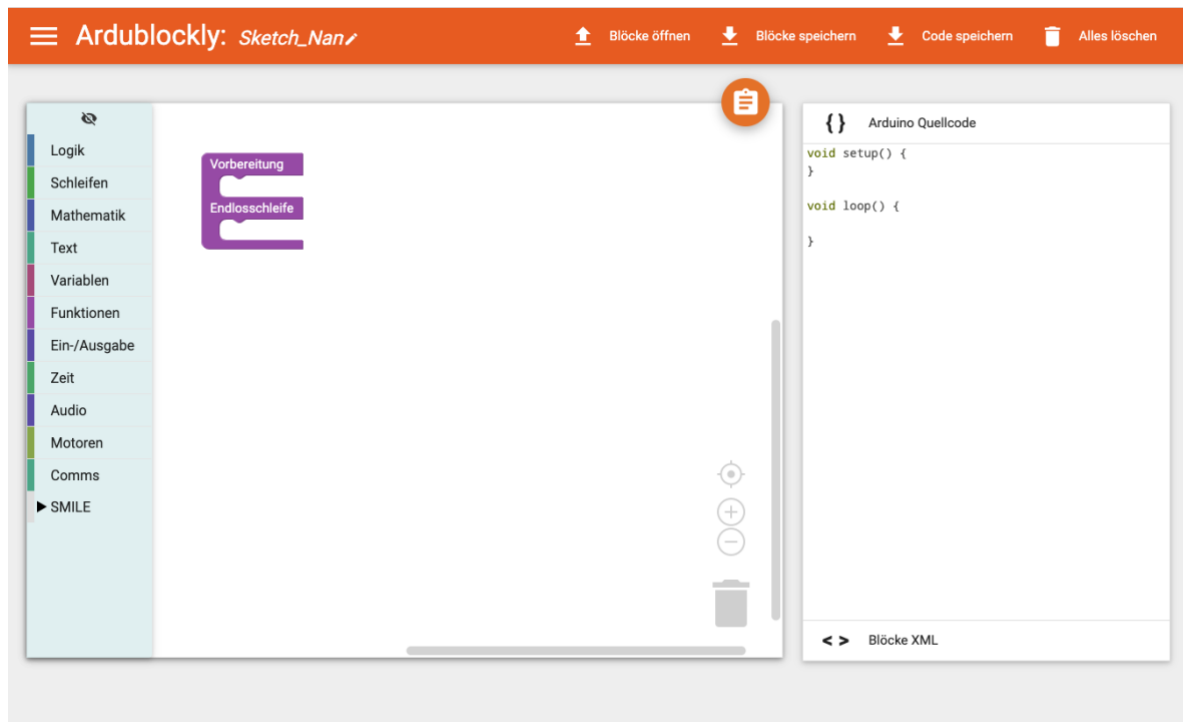
Diese Materialien müssen vor Beginn des Workshops bestellt werden (vgl. Benötigte Materialien). Weiterhin wird pro Teilnehmer eine Box benötigt, in die die LED geklebt werden kann. Diese Box muss vor Beginn des Workshops 3D-gedruckt werden. Die STL-Dateien dazu finden sich im [GIT-Repository](#) (Material\_10, Material\_11). Während des Workshops werden dann die Stechkabel an die LED gelötet und die LED mithilfe der Stechkabel mit dem Mikrocontroller verbunden. Die LED wird außerdem in die Box geklebt. In dieser befindet sich eine Aussparung, in die der Acrylstab mit der Blume oder sonstiger von den Teilnehmenden erzeugten Figur gesteckt werden kann. Die Programmierung des Mikrocontrollers wird im folgenden Abschnitt erläutert.

### 3.3 Physical Computing

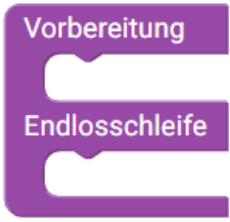

Beim Physical Computing werden informatische Systeme betrachtet, die Software und Hardware nutzen um mit der Umwelt zu interagieren, indem sie mithilfe von Sensoren die Umwelt erfassen und mittels Aktoren entsprechend darauf reagieren. Ein entsprechend programmierter Mikrocontroller kann zum Beispiel die Temperatur oder die Luftfeuchtigkeit mithilfe von Sensoren erfassen und über Aktoren beispielsweise ein Geräusch oder einen Lichteffekt erzeugen wenn diese einen bestimmten Grenzwert überschreiten.

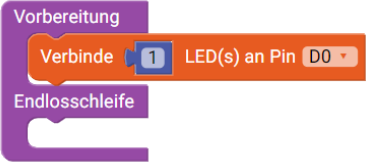






In diesem Workshop soll eine LED entsprechend der Wetterlage, die über die Openweathermap-API (siehe folgendes Kapitel) aus dem Internet abgerufen wird, in verschiedenen Effekten leuchten. Es wird hierfür der Mikrocontroller ESP8266 entweder mithilfe der *ArduinoIDE* in C und C++ oder in der blockbasierten Programmieroberfläche Ardublockly programmiert. In beiden Fällen wird die *ArduinoIDE* benötigt die durch die Installation von Bibliotheken (im Englischen *libraries*) erweitert werden kann. Die für diesen Workshop benötigten Bibliotheken sollten im Vorfeld des Workshops auf den verwendeten Geräten installiert werden (Eine Handreichung ist durch Material\_03 gegeben).

Für die textbasierte Programmierung des Mikrocontrollers mithilfe der *ArduinoIDE* findet sich ein entsprechend dokumentierter Sketch im [GIT-Repository](#) (Material\_12). Je nach Alter und Vorkenntnissen der Teilnehmenden ist jedoch die blockbasierte Programmierung mithilfe der online Programmieroberfläche [Ardublockly](#) zu empfehlen.



Die Oberfläche von Ardublockly unterteilt sich in drei Bereiche. Im Menü auf der linken Seite befindet sich eine Vielzahl von Blöcken. Diese reichen von logischen Verknüpfungen (Wenn-Dann Bedingung) über Schleifen, Mathematik, Variablen und Zeit bis zu eigens für den Workshop entwickelten SMILE-Blöcken. Im mittleren Arbeitsbereich können die Blöcke per Drag & Drop gezogen und angeordnet werden. Ardublockly stellt dabei parallel den durch die Blöcke erzeugten Arduino Code in der rechten Spalte dar, sodass die Teilnehmenden auch die textuelle Programmierung kennenlernen. Änderungen in den Blöcken werden zudem im Arduino Code gelb hinterlegt und können so besser nachvollzogen werden.

	<p>Das Grundgerüst der Programmierung des Mikrocontrollers besteht aus den beiden Teilen <i>setup</i> und <i>loop</i>. Diese werden in Ardublockly als Vorbereitung und Endlosschleife dargestellt und sind beim Starten der Oberfläche bereits geladen.</p>	<pre> {}  Arduino Quellcode  void setup() { }  void loop() { } </pre>
	<p>Alle Blöcke die innerhalb der <i>setup</i>-Funktion stehen werden beim Einschalten bzw. Zurücksetzen des Mikrocontrollers einmal ausgeführt.</p>	<pre> {}  Arduino Quellcode  void setup() { }  void loop() { } </pre>

	<p>In der <i>setup</i>-Funktion werden die grundlegenden Definitionen vorgenommen. Mit dem „Verbinde LED“-Block wird bspw. die Anzahl der LEDs sowie der Pin, an denen diese angeschlossen sind, festgelegt.</p>	<pre>#include &lt;FastLED.h&gt;  #define NUM_LEDS 1 #define DATA_PIN D0 CRGB leds[NUM_LEDS];  void setup() {   FastLED.addLeds&lt;NEOPIXEL, DATA_PIN&gt;(leds, NUM_LEDS); }  void loop() { }</pre>
	<p>Alle Blöcke innerhalb der <i>loop</i>-Funktion werden vom Mikrocontroller immer wieder durchlaufen.</p> <p>In der <i>loop</i>-Funktion werden Sensoren ausgelesen oder Aktoren wie beispielsweise LEDs angesteuert.</p>	<p><b>{} Arduino Quellcode</b></p> <pre>void setup() { }  void loop() { }</pre> <pre>#include &lt;FastLED.h&gt;  #define NUM_LEDS 1 #define DATA_PIN D0 CRGB leds[NUM_LEDS];  void setup() {   FastLED.addLeds&lt;NEOPIXEL, DATA_PIN&gt;(leds, NUM_LEDS); }  void loop() {   leds[int(0)] = CRGB(int(231), int(91), int(53));   FastLED.show(); }</pre>
<p>Um das mit Ardublockly erstellte Programm auf den Mikrocontroller zu übertragen, muss der Arduino Code in die zuvor installierte ArduinoIDE kopiert werden.</p> <p>Nach dem Einfügen kann der Programmcode kompiliert und anschliend hochgeladen werden.</p>		<p>In Zwischenablage kopieren </p> <p>    Hochladen mit Programmer</p>

In Ardublockly lässt sich der komplette Programmablauf der Wetterlampe, vom Verbinden des Mikrocontrollers mit dem Internet, über das Abrufen der Wetterinformationen, bis hin zum Darstellen entsprechender Wettereffekte, nachbauen. Eine entsprechende XML-Datei zum Importieren der Blöcke in Ardublockly findet sich zusammen mit einer Beschreibung der verwendeten Blöcke im [GIT-Repository](#) (Material\_15, Material\_16).

### 3.4 Openweathermap-API

Die aktuellen Wetterinformationen können über [OpenWeatherMap.org](https://openweathermap.org) abgerufen werden. Die zur Verfügung gestellte Programmierschnittstelle (API) benötigt für den Zugriff auf die Wetterzustände [API-Keys](#) welche nach einer Registrierung für alle können für die Teilnehmenden kostenlos generiert werden können. Diese Keys werden in den entsprechenden Block in Ardublockly oder direkt in den Arduino Code eingefügt.

Der Mikrocontroller ermittelt über HTTP-Requests und Parsen der übermittelten JSON-Strings die aktuelle Temperatur sowie Wetterlage (sonnig, bewölkt, gewittrig, ...) und spielt abhängig von diesen unterschiedliche Wettereffekte ab, welche von den Teilnehmenden selbst in Kleingruppen mit Ardublockly oder der ArduinoIDE gestaltet wurden.

## 4 Variationen

### 4.1 Variante 1 – Kompletter Workshop vom Entwurf bis zur Präsentation

Phase (laut Verlaufsplan)	Kurze Zusammenfassung	Dauer: 3 Tage á ca. 6 Stunden
Einstieg in parametrisches Design	Einführung und erste Erfahrung mit der Umgebung BlocksCAD anhand eines Arbeitsblattes in Form eines Tutorials.	
Vertiefung in <i>BlocksCAD</i>	Die Teilnehmenden erstellen mit BlocksCAD ein eigenes Modell, das danach mit einem 3D-Drucker gedruckt wird.	
Einarbeitung in Physical Computing	Die Teilnehmenden programmieren einen Mikrocontroller. Ziel ist die Anzeige eines Farbeffekts auf der LED passend zu einem von sieben Wetterzuständen. Dieser Mikrocontroller wird anschließend mit der selbstgestalteten Blume verbunden.	
Sicherung der Ergebnisse	Die Teilnehmenden präsentieren ihre erstellte smarte Blume in einem selbstgedrehten Tutorial oder Poster.	

### 4.2 Variante 2 – Erstellen einer leuchtenden Blüte

Phase (laut Verlaufsplan)	Kurze Zusammenfassung	Dauer: 2 Tage á ca. 6 Stunden
Einstieg in parametrisches Design	Einführung und erste Erfahrung mit der Umgebung BlocksCAD anhand eines Arbeitsblattes in Form eines Tutorials.	
Vertiefung in <i>BlocksCAD</i>	Die Teilnehmenden erstellen mit BlocksCAD eine eigene Blüte, die danach mit einem 3D-Drucker gedruckt wird.	
Einarbeitung in Physical Computing	Die Teilnehmenden programmieren einen Mikrocontroller. Ziel ist die Anzeige eines bestimmten Farbeffekts entsprechend eines Wetterzustands. Dieser Mikrocontroller wird anschließend mit der selbstgestalteten Blume verbunden.	

### 4.3 Variante 3 – Erstellen einer eigenen Blüte mittels *BlocksCAD*

Phase (laut Verlaufsplan)	Kurze Zusammenfassung	Dauer: ca. 6 Stunden
Einstieg in parametrisches Design	Einführung und erste Erfahrung mit der Umgebung BlocksCAD anhand eines Arbeitsblattes in Form eines Tutorials.	
Vertiefung in <i>BlocksCAD</i>	Die Teilnehmenden erstellen mit BlocksCAD eine eigene Blüte oder andere parametrisch gestaltete Form, die danach mit einem 3D-Drucker gedruckt werden kann.	

## 5 Materialliste

### 5.1 Benötigte Materialien

#### **Für Variante 3:**

- Computer/Laptop mit Internetzugang für alle SuS
- Beamer zur Demonstration von Zwischenschritten
- 3D-Drucker mit ausreichend Filament oder Zugang zu 3D-Druckern (optional)

#### **Zusätzlich für Variante 2:**

- 3D-Drucker mit ausreichend Filament oder Zugang zu 3D-Druckern
- Mikrocontroller ESP8266 mit LED und Display für jede Teilnehmerin
- Ausreichend USB-Kabel, um die Mikrocontroller an die Computer/Laptops anzuschließen
- Ggf. Powerbanks, um die Mikrocontroller mit Strom zu versorgen (optional)
- Boden der Blume zum Halten des Stängels und „Verstecken“ des Mikrocontrollers (3D-Druckdateien siehe Anhang)
- Plexiglas-Stangen mit 5 mm Durchmesser als Stängel der Blume
- Säge zum Anpassen der Länge des Stängels
- Heißkleber zum Befestigen der Blüte an dem Stängel und des Stängels an dem Boden

#### **Zusätzlich für Variante 1:**

- Ggf. Kameras/Smartphones zum Filmen, falls die Teilnehmenden kein eigenes Smartphone haben
- Optional: Flipchartbögen, Moderationskoffer mit Stiften, Moderationskarten etc.

### 5.2 Lern-Materialien

Nr.	Titel	Beschreibung	Benötigt für Variante(n)
M_04	Arbeitsblatt1	Enthält die angeleiteten Arbeitsaufträge zum Erstellen einer beispielhaften Blume.	1, 2, 3
M_05	Hilfekarten zu AB 1	Enthält Hilfestellungen bzw. Lösungen für einzelne Teilaufgaben aus Arbeitsblatt 1	1, 2, 3
M_07	Arbeitsblatt2	Enthält Arbeitsaufträge zum eigenständigen Erschließen des RGB-Farbschemas	1, 2
M_06	Kurzanleitung	Enthält eine Anleitung zum Zusammenbauen und Verbinden mit dem WLAN der smarten Lampe	1, 2

## 6 Verlaufspläne

### 6.1 Einstieg in parametrisches Design

Zeit	Phase	Inhalt	Material
10 Min.	Einführung	Vorstellung der Umgebung BlocksCAD anhand eines Beispiels und zeigen von Beispielen, die mit parametrischem Design erstellt wurden. Hierfür eignen sich Vasen oder auch Blüten, die mit dem Suchbegriff „parametric design“ im Internet schnell gefunden werden. An diesen Beispielen kann die Regelmäßigkeit gezeigt werden und somit dargestellt werden, was das Merkmal von parametrischem Design ist.	
10 Min.	Erarbeitung1	Teilnehmende erhalten ein Arbeitsblatt und bearbeiten Abschnitt 1, Kursleitung steht unterstützend zur Seite.	Arbeitsblatt 1 „BlocksCAD Tutorial“, Hilfekarten zu AB1
5 Min.	Sicherung1	Kurze gemeinsame Besprechung der Aufgabe samt Präsentation am Beamer.	
10 Min.	Erarbeitung2	Die Teilnehmende bearbeiten selbstständig Abschnitt 2 auf dem AB	Arbeitsblatt 1 „BlocksCAD Tutorial“, Hilfekarten zu AB1
5 Min.	Sicherung2	Kurze gemeinsame Besprechung der Aufgabe samt Präsentation am Beamer.	
50 Min.	Erarbeitung3	Kurze Erarbeitung bzw. Erklärung der Kursleitung, dass Schleifen dafür da sind, Befehle, die öfter ausgeführt werden, in einer Schleife zusammengefasst werden können, die dann immer wieder hintereinander aufgerufen wird. Danach wird der Vorteil von Variablen erläutert, in denen ein Wert gespeichert ist, der zum Beispiel innerhalb einer Schleife bei jeder Iteration entsprechend immer weiter verändert werden kann. Die Teilnehmenden bearbeiten zuerst Abschnitt 3 auf dem AB, in denen sich Schleifen und Variablen zunutze gemacht werden, um damit nach und nach eine beispielhafte Blüte zu erstellen. Die Kursleitung steht unterstützend zur Seite. Danach wird Abschnitt 4 bearbeitet, der den Teilnehmenden verschiedene Mengenoperationen in <i>BlocksCAD</i> zeigt, welche für das Erstellen des Loches unter der Blüte benötigt werden, damit der Stängel dort angebracht werden kann. Hierfür wird der Differenzblock verwendet, indem eine Form von einer anderen Form „abgezogen“ wird.	Arbeitsblatt 1 „BlocksCAD Tutorial“, Hilfekarten zu AB1
10 Min.	Sicherung3	Es wird gemeinsam Abschnitt 3 und 4 besprochen, um sicherzustellen, dass die Teilnehmenden sowohl Schleifen und Variablen als auch Mengenoperationen und ihre Funktion verstanden haben, damit sie diese bei der Erstellung einer eigenen Blüte selbst anwenden können.	

## 6.2 Vertiefung in *BlocksCAD*

Zeit	Phase	Inhalt	Material
10-15 Min.	Einführung	Verschiedene Blütenformen werden präsentiert und im Anschluss suchen die Teilnehmenden im Internet nach einer Blütenform für ihr eigenes Modell. Die Kursleitung sollte hierbei darauf achten, dass diese auch umsetzbar ist. (Bei Bedarf danach die ausgewählten Blüten ausdrucken, damit die Teilnehmenden sie während der Erarbeitung ständig präsent haben.)	
5 Min.	Hinführung	Ggf. eine kurze Zusammenfassung, in welchen Schritten man bei der Blütenerstellung vorgeht mit Verweis auf das Arbeitsblatt.	
45-90 Min. (je nach Kenntnisstand der TM)	Erarbeitung	Die Teilnehmenden erstellen ihre eigenen Blüten und die Kursleitung steht unterstützend zur Seite. Für den späteren Stängel aus Plexiglas muss ein entsprechendes Loch an die Unterseite der Blüte gelassen werden. Dieser Schritt wird vorne am Beamer vorgemacht und den Teilnehmenden die entsprechende Größe für das Loch vorgegeben. Um die Produktionsdauer und -kosten möglichst gering zu halten, wird für die Blüte eine Größe von nicht mehr als 4x4x4cm empfohlen.	
10 Min.	Sicherung	Die Teilnehmenden stellen den anderen Teilnehmenden ihr erstelltes Blütenmodell vor. Hierbei können einige Teilnehmende auch ihren Code erklären, falls die Zeit ausreicht.	



### 6.3 Einarbeitung in Physical Computing

Zeit	Phase	Inhalt	Material
15 Min.	Einführung	Um den Teilnehmenden zu zeigen, um was es gehen soll, wird ihnen der Mikrocontroller ESP8266 gezeigt mit dem fertigen Programm, das eine LED fortlaufend verschiedene Farben anzeigt. Zudem wird erklärt, was ein Mikrocontroller ist und dass dieser Mikrocontroller eine vereinfachte Version eines Computers ist. Anschließend wird mithilfe der Kurzanleitung der Mikrocontroller mit der LED verbunden. Es ist auch möglich, an dieser Stelle schon die LED in die Box mit dem Acrylgasstab zu legen, um die Farben, die programmiert werden schon anzeigen zu können.	
10 Min.	Hinführung	Den Teilnehmenden wird Ardublockly vorgestellt. Gemeinsam werden ein paar Blöcke auf der Arbeitsfläche platziert, Einstellungen verändert und anschließend das Hochladen auf den Mikrocontroller mithilfe der ArduinoIDE exemplarisch durchgeführt.	
15 Min.	Erarbeitung I	Die Teilnehmenden erhalten ein Arbeitsblatt und bearbeiten die Aufgabe 1 auf diesem selbstständig. Die Kursleitung steht unterstützend zur Seite	Arbeitsblatt „Ardublockly“ und dazugehörige Hilfekarten
5 Min.	Sicherung I	Kurze gemeinsame Besprechung der Aufgabe samt Präsentation am Beamer.	
15 Min.	Erarbeitung II	Die Teilnehmenden bearbeiten selbständig Aufgabe 2 und 5 vom Arbeitsblatt. In diesen sollen die Teilnehmenden Lieblingsfarben aussuchen und diese anzeigen lassen. An dieser Stelle sollte auf das additive Farbmodell eingegangen werden und die Äquivalenz zu der Schreibweise der Farbe als Wort aufgezeigt werden.	Arbeitsblatt „Ardublockly“ und dazugehörige Hilfekarten Arbeitsblatt 2 „Farben mischen“
60 Min.	Vertiefung	Die Teilnehmenden erarbeiten in Gruppen Effekte für die unterschiedlichen Wetterlagen, wobei jede Gruppe einen Effekt erarbeitet. Als Weiterführung oder didaktische Reserve kann bereits das Display auf den Mikrocontroller gesteckt werden, und die Teilnehmenden mithilfe entsprechender Blöcke Textnachrichten auf diesem anzeigen lassen.	
30 Min.	Sicherung	Der vollständige Programmcode ist sehr komplex und wird den Teilnehmenden daher zur Verfügung gestellt. Gemeinsam werden die wichtigsten Teile des Codes besprochen, die gestalteten Wettereffekte zusammengeführt und der Code auf die Mikrocontroller hochgeladen.	
60 Min.	Zusammensetzen der Blume	Die Teilnehmenden erhalten ihre gedruckten Formen, die sie mit dem Loch unterhalb auf dem Acrylgasstab befestigen. Die LED wird mit Heißkleber an den Boden der Box angeklebt und der vorgefertigte Kasten aufgesetzt. Die Teilnehmenden individualisieren ihre Blumen indem sie ihre Stängel aus Acrylglas mit Schleifpapier anrauen oder in der Länge variieren.	

## 6.4 Sicherung der Ergebnisse

Zeit	Phase	Inhalt	Material
10-15 Min.	Einführung	Es werden einige beispielhafte Werbe- und Erklärvideos angeschaut und Merkmale davon gesammelt, damit sich die Teilnehmenden Inspirationen für ihren eigenen Werbespot holen können.	
10 Min.	Hinführung	Die Teilnehmenden werden in Vierergruppen eingeteilt und besprechen gemeinsam Ideen für ihre Präsentation/ihr Video. Im Video sollen zum einen die Vorgehensweise und der Entstehungsprozess erklärt und gezeigt werden. Zudem soll aber auch Werbung für die Blume gemacht werden, warum man diese brauchen könnte.	
200 Min.	Erarbeitung	In den Vierergruppen erstellen die Teilnehmenden mithilfe von digitalen Medien (z.B. Powerpoint-Präsentation, Videokamera) und analogen Medien (z.B. Flipchartbögen, Moderationskarten) ihre eigene Präsentation der Blume.	
60 Min.	Sicherung	Die erstellten Präsentationen/Videos werden mit allen Teilnehmenden gemeinsam angeschaut und diskutiert.	

## 7 „Lessons learnt“

Nach der ersten Durchführung des Workshops fiel auf, dass die Teilnehmenden mit dem Verstehen eines textbasierten Programmcodes Schwierigkeiten hatten und ihnen dieser sehr abstrakt erschien. Die Programmierung mit den Blöcken aus *BlocksCAD* hingegen schien intuitiver, wodurch die Teilnehmenden mehr ausprobierten. In weiteren Durchführungen wurde daher mit der blockbasierten Programmieroberfläche *mBlock* gearbeitet, was die Programmierung der Wettereffekte erheblich erleichterte und den Teilnehmenden auch hier mehr experimentierten. Die Installation und Update der entsprechenden Software sowie der für den Workshop benötigten Erweiterungen auf jedem Computer führte jedoch zu einem erhöhten Administrationsaufwand im Vorfeld der Workshops. Die zuletzt eingesetzte Programmieroberfläche Ardublockly erlaubt ebenfalls die blockbasierte Programmierung und lässt sich als Webapplikation zentral administrieren. Durch die parallele Darstellung sowohl der blockbasierten als auch textbasierten Programmiersprache können zudem Änderungen unmittelbar nachvollzogen und damit enger Verknüpfung werden.

### **Bezüglich des Druckens der Blume:**

Die Erfahrung hat gezeigt, dass es am wenigsten zu Fehlern führt, wenn die Blumen nicht komplett hohl, sondern leicht gefüllt gedruckt werden. Außerdem sollte jeweils nur eine Blume auf einem Drucker produziert werden (Produktionszeit pro Blume ca. 2 Stunden).

Das Löten der LEDs kann bereits parallel zu der Phase des parametrischen Designs durchgeführt werden, um sicherzustellen, dass in der Phase des Physical Computing die Teilnehmenden, mindestens zu zweit eine LED mit Mikrocontroller haben, die sie ansteuern können. Auch ist es empfehlenswert, die LED schon für die Phase des Physical Computing in den vorgesehenen Kasten mit Acrylglasstab zu legen, da die LED zu hell leuchtet, um auf Dauer direkt hineinzuschauen.