

smile



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
4	Variationen.....	9
4.1	Variante 1 – Kompletter Workshop vom Entwurf bis zur Präsentation.....	9
4.2	Variante 2 – Erstellen einer leuchtenden Blüte.....	9
4.3	Variante 3 – Erstellen einer eigenen Blüte mittels <i>BlocksCAD</i>	9
5	Materialliste.....	10
5.1	Benötigte Materialien	10
5.2	Lern-Materialien.....	10
6	Verlaufspläne	11
7	„Lessons learnt“	15

Gestaltung und Konzeption dieses Workshops:



Carl von Ossietzky Universität Oldenburg
Abteilung Didaktik der Informatik

1 Kurzzusammenfassung

Dieser Workshop behandelt das Erstellen einer smarten Blume, die von den Teilnehmenden zuerst mittels parametrischem Design selbst entworfen wird und anschließend durch einen Sensor (bspw. Temperatur oder Helligkeit) verschiedene Farben annimmt. Der Workshop kann vollständig an drei Tagen durchgeführt werden, sodass am Ende die vollständig selbst produzierte Blume steht. 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 je nach Temperatur eine andere Farbe annimmt. Diese Blume wird anschließend von den Teilnehmerinnen durch das Erstellen eines Werbevideos präsentiert.

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

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

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.
- Die Teilnehmenden lernen Programmierung als Mittel zur Umsetzung eigener Ideen kennen.
- Die Teilnehmenden erwerben grundlegende Kenntnisse der Unfallverhütungsvorschriften im Umgang mit den genutzten Werkzeugen am Beispiel eines Lötkolbens.
- Die Teilnehmenden lernen Grundlagen des Lötens, welches der Umsetzung eigener Ideen und Projekte dienen kann.

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 sind sich den Vorteilen beim Programmieren bewusst.
- Die Teilnehmenden kennen das Grundprinzip von Mengenoperationen, wie sie bei Parametrischem Design angewendet werden.
- Die Teilnehmenden kennen die Grundlagen des Lötens und sind in der Lage selbstständig zwei Kontakte fachgerecht zusammenzulöten.
- Die Teilnehmenden kennen zudem die Unfallverhütungsvorschriften und berücksichtigen diese.

Zu Physical Computing:

- Die Teilnehmenden vertiefen Programmierkonzepte wie Schleifen, Variablen und Mengenoperationen.
- Die Teilnehmenden stellen eine Verknüpfung zwischen einem Programmcode und dem Mikrocontroller her.
- 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

Parametrisches Design:

Regelmäßige Formen und Muster treten in der Natur häufig auf und der Mensch empfindet sie intuitiv als schön. Aus diesem Grund werden sie oft bei künstlerischen Objekten, wie beispielsweise bei Vasen, verwendet. Zusätzlich finden sie vor allem in der Architektur ihre Anwendung. Diese regelmäßigen Formen lassen sich mit parametrischem Design herstellen. Das Merkmal von solch einem parametrischen Design ist, dass seine Form durch eine Kombination von Parametern und entsprechenden Zahlenvariablen zur Größenbeschreibung erzeugt wird. Die Parameter werden gespeichert. So können Abhängigkeiten und Beziehungen zwischen ihnen hergestellt werden. Durch eine Veränderung dieser Abhängigkeiten entstehen verschiedene Formen. Ein Quader wird beispielsweise durch die Parameter Höhe, Breite und Tiefe beschrieben.

Die Erstellung solcher parametrischen Formen kann mit sogenannten CAD-Systemen (aus dem Englischen: computer-aided design) vorgenommen werden. Die damit erstellten Modelle können anschließend mit einem 3D-Drucker ausgedruckt werden.

In dem in diesem Workshop verwendeten CAD-System *BlocksCAD* können diese Formen durch verschiedene Blöcke erstellt werden. Im folgenden Beispiel wurde mithilfe des Blockes „Würfel“ ein Würfel mit der Seitenlänge 10 erstellt, der nicht zentriert ist. *BlocksCAD* arbeitet mit einem dreidimensionalen Koordinatensystem mit den drei Achsen X, Y und Z. Wenn die Figur nicht zentriert ist, wird im Ursprung (0,0,0) begonnen. Wie in dem Beispiel wird der Würfel mit jeweils 10

Längeneinheiten in X-, Y- und Z-Richtung konstruiert. Wenn die Einstellung zentriert gewählt wird, wird der Würfel gleichmäßig um den Ursprung konstruiert.

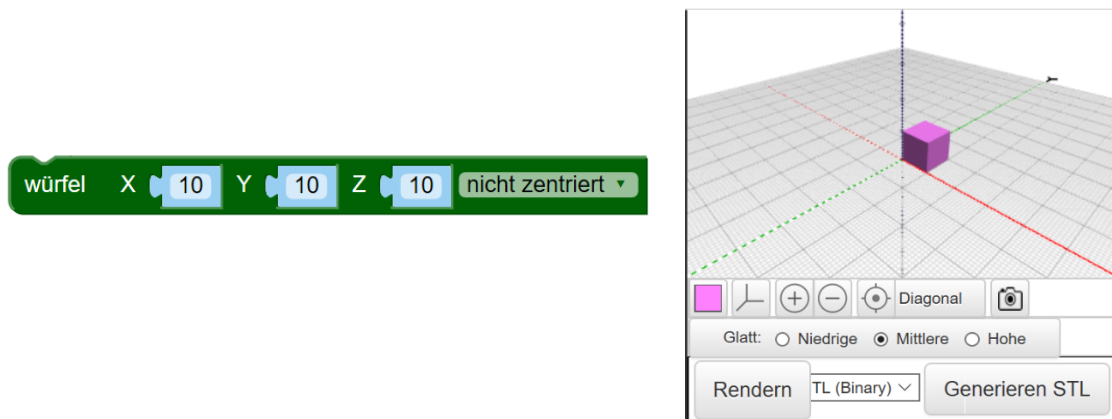


Abbildung 1: Erstellen eines Würfels mit dem „Cube“-Block in *BlocksCAD*.

Die Teilnehmenden entwickeln nach und nach mit *BlocksCAD* eine Blume, indem sie Eigenschaften des parametrischen Designs dafür kennenlernen und somit anwenden. Sie vertiefen ihr Wissen durch das konstruieren einer Blume nach eigenen Vorstellungen. Es ist nicht zwingend notwendig, wird jedoch empfohlen, das erstellte Modell nach Abschluss der Programmierung mit einem 3D-Drucker anfertigen zu lassen, damit die Teilnehmenden ihr eigenes Ergebnis auch anfassen und begreifen können.

Physical Computing:

Beim Physical Computing werden informatische Systeme betrachtet, die Software und Hardware nutzen und in Verbindung mit der physikalischen Umwelt stehen. Hierbei geht es vor allem um Systeme, die durch entsprechende Sensoren die Umwelt erfassen und dann mittels Aktoren entsprechend darauf reagieren. Diese Sensoren können beispielsweise die Temperatur oder die Luftfeuchtigkeit erfassen. Ein Aktor kann zum Beispiel ein Geräusch erzeugen, wenn der Sensor eine bestimmte Temperaturüberschreitung gemessen hat.

In diesem Workshop soll durch einen Temperatursensor eine LED entsprechend der Temperatur, die über die Openweathermap-API (siehe folgendes Kapitel) aus dem Internet geholt wird, in verschiedenen Effekten leuchten. Es wird hierfür der Mikrocontroller ESP8266 mithilfe der *ArduinoIDE* programmiert. Im folgenden Beispiel werden mittels des Beispielcodes „Blink“ für den ESP8266 einige Grundprinzipien für diese Programmierung erläutert:

```

void setup() {
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS)
}

void loop() {
    leds[0] = CRGB::Red;
    FastLED.show();
    delay(500);

    leds[0] = CRGB::Black;
    FastLED.show();
    delay(500);
}

```

Bild 2: Beispielcode „Blink“ für den ESP8266

Der dargestellte Code beschreibt ein Programm, bei dem die angeschlossene LED in einem Intervall von 0,5 Sekunden rot blinkt.

In der *setup*-Funktion werden die grundlegenden Definitionen vorgenommen, wie beispielsweise die Anzahl der LEDs und die Stellen, an denen sie angeschlossen sind. Die *setup*-Funktion wird beim Einschalten bzw. Zurücksetzen des Mikrokontrollers genau einmal ausgeführt.

Die *loop*-Funktion stellt eine Schleife dar und wird somit iterativ immer wieder ausgeführt. Nachdem die *setup-Funktion* erstmalig durchlaufen wurde, wird das in diesem Teil beschriebene Programm somit vom Mikrokontroller immer wieder durchlaufen. Die „{,“ beschreibt den Anfang der Schleife und die „}“ beendet jene. Die *show*-Funktion innerhalb der Schleife lässt die angeschlossene LED die in der Zeile darüber definierte Farbe anzeigen. Die Farbdefinition wurde hier mit „CRGB::Red“ vorgenommen. Es gibt aber auch die Möglichkeit, die Farben mittels des additiven Farbsystems darzustellen, welches die Grundfarben Rot, Grün und Blau mit jeweils unterschiedlicher Stärke miteinander mischt. Die *delay*-Funktion lässt das Programm für die in Klammern angegebenen Millisekunden pausieren, bevor das Programm den nächsten Schritt innerhalb der Schleife ausführt. Somit werden immer 500 Millisekunden (0,5 Sekunden) abwechselnd rot und schwarz bei der LED angezeigt, was den Eindruck eines Blinkens erzeugt.

Internet-of-Things und Openweathermap-API:

Mithilfe der OpenWeatherMap API (<https://openweathermap.org/api>), über die man sich kostenlos API Keys generieren lassen kann, die sich dann zur Weitergabe an die Workshopteilnehmenden eignen, soll der ESP8266 internetfähig gemacht werden. Über HTTP-Requests und ein Parsen der übermittelten JSON-Strings spielt der Mikrokontroller je nach Wetterlage (sonnig, bewölkt, gewittrig, ...) unterschiedliche Lichteffekte ab, die die Teilnehmenden selbst in Kleingruppen erarbeiten.

Bezüglich der Programmierung sei an dieser Stelle auf den Arduino-Beispielcode im Digitalen Archiv verwiesen.

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 eine eigene Blüte, die danach mit einem 3D-Drucker gedruckt wird.	
Einarbeitung in Physical Computing	Die Teilnehmenden programmieren einen Mikrocontroller. Ziel ist die Anzeige einer bestimmten Farbe entsprechend einer angezeigten Temperatur. Dieser Mikrocontroller wird anschließend mit der selbstgestalteten Blume verbunden.	
Sicherung der Ergebnisse	Die Teilnehmenden präsentieren ihre erstellte smarte Blume in einem selbstgedrehten Tutorial oder Werbevideo.	

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 einer bestimmten Farbe entsprechend einer angezeigten Temperatur. 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, die danach mit einem 3D-Drucker gedruckt werden kann.	

Materialliste

4.4 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.

4.5 Lern-Materialien

Nr.	Titel	Beschreibung	Benötigt für Variante(n)
1	Arbeitsblatt	Enthält die angeleiteten Arbeitsaufträge zum Erstellen einer beispielhaften Blume.	1, 2, 3

5 Verlaufspläne

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 „BlocksCAD Tutorial“
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 „BlocksCAD Tutorial“
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 „BlocksCAD Tutorial“
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.	

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 5x5x5cm 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.	

Einarbeitung in Physical Computing

Zeit	Phase	Inhalt	Material
10 Min.	Einführung	Um den Teilnehmenden zu zeigen, um was es gehen soll, wird ihnen der Mikrocontroller ESP8266 gezeigt mit dem fertigen Programm, dass eine LED fortlaufend verschiedene Farben anzeigt. Zudem wird erklärt, was ein Mikrocontroller ist und dass dieser Mikrocontroller eine vereinfachte Version eines Computers ist.	
30 Min.	Hinführung	Um die Idee der Programmierung eines Mikrocontrollers zu verstehen, wird gemeinsam das Beispiel „Blink“ durchgegangen. Wenn die Teilnehmenden noch keine Erfahrung im Bereich Programmierung haben, dann sollte der Code Zeile für Zeile durchgegangen werden und hierbei unter anderem auf die Kommentarfunktion, Funktionen im Allgemeinen und im Speziellen für das	

		Beispiel, geschweifte Klammern und Schleifen eingegangen werden. Nachdem der Code gemeinsam besprochen wurde, sollen die Teilnehmenden ihn auf ihren Mikrocontroller hochladen. Als Aufgabe sollen die Teilnehmenden anschließend versuchen, dass die Lampe immer wieder 5 Sekunden an ist und dann wieder 5 Sekunden aus.	
60 Min.	Erarbeitung	Nachdem die Teilnehmenden die grundlegende Idee der Programmierung verstanden haben, wird eine LED an den Mikrocontroller angeschlossen (Schwarz: Minus-Pol der LED, Rot: Plus-Pol der LED, Gelb: Daten-Pin) und sich für das Paket „FastLED“, welches in der Arduino-Bibliothek zu finden ist, ebenfalls das „Blink“-Programm gemeinsam anzusehen und wir in der Hinführung den Code durchzugehen und die Teilnehmenden versuchen lassen, die einzelnen Komponenten in dem Code erklären zu lassen. Als Aufgaben dienen hier das Ändern der Farben, mehrere Farben hintereinander anzeigen zu lassen (indem entsprechende Zeilen kopiert werden), aber auch die Darstellung der Farbe als RGB-Befehl. An dieser Stelle sollte auf das additive Farbmodell eingegangen werden und die Äquivalenz zu der Schreibweise der Farbe als Wort aufgezeigt werden.	
60 Min.	Vertiefung	Als Weiterführung wird nun auf den Mikrocontroller ein Display gesteckt (hierbei ist darauf zu achten, dass die Ein- und Ausgänge aufeinander gesteckt werden). In diesem Teil lernen die Teilnehmenden also, dass nicht nur LEDs programmiert werden können, sondern auch auf einem Display entsprechender Text angezeigt wird, wenn er im Programm so festgelegt wurde.	
30 Min.	Sicherung	Da das am Ende tatsächlich verwendete Programm zu komplex ist, wird es den Teilnehmenden zur Verfügung gestellt (siehe Anhang) und die wichtigsten Teile des Codes gemeinsam besprochen. Dieser Code wird auf den Mikrocontroller hochgeladen.	
60 Min.	Zusammensetzen der Blume	Die Teilnehmenden stecken in diesem Abschnitt ihre komplette Blume zusammen. Hierfür wird der Mikrocontroller mit der LED mit Heißkleber an den Boden angeklebt und danach der vorgefertigte Kasten aufgesetzt. Danach können sich die Teilnehmenden entscheiden, ob sie ihren Stängel aus Plexiglas abschmirlen möchten oder glatt lassen wollen und welche Länge er haben soll und dann auf die gewünschte Länge sägen. Die erstellte Blüte mit dem Loch unterhalb der Knospe wird dann auf den Stängel aufgesetzt.	

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.	

6 „Lessons learnt“

Nach der ersten Durchführung des Workshops ist aufgefallen, dass die Teilnehmenden mit dem Verstehen eines Programmcodes Schwierigkeiten hatten und er ihnen sehr abstrakt erschien. Dadurch konnten sie die meisten Dinge eher nachmachen. Die Programmierung mit den Blöcken aus *BlocksCAD* hingegen schien etwas intuitiver zu sein. Die Schülerinnen und Schüler haben mehr von sich aus ausprobiert. Es empfiehlt sich daher, eine explizite Verbindung zwischen der blockbasierten und der textbasierten Programmiersprache zu schaffen, indem Beispiele eines Codes beider Programmierarten miteinander verglichen 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).