

Hausarbeit im Seminar  
Physical-Computing-Projekte im Informatikunterricht

## Projekt Arduino-Gewächshaus

Malte Kruse, Malte Lundschien

09.07.2018



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>3</b>
<b>2. Beschreibung des Projektes</b>	<b>4</b>
2.1 Materialien	4
2.1.1 Gewächshaus	5
2.1.2 Sprühflasche	5
2.1.3 Arduino Nano V3.0	6
2.1.4 Steckbrett und Jumper-Kabel	6
2.1.5 Sensoren	7
2.2 Snap4Arduino	9
2.2.1 Benutzeroberfläche	9
2.2.2 Menüleiste	10
2.2.3 Block-Bibliothek	10
2.2.4 Programmbereich	11
2.2.5 Zeichenbereich	12
<b>3. Einordnung in den Rahmenlehrplan</b>	<b>12</b>
<b>4. Projektumsetzung</b>	<b>13</b>
4.1 Vorbereitung	13
4.2 Durchführung	13
<b>5. Erweiterungen</b>	<b>14</b>
<b>6. Erprobung durch Studierende</b>	<b>14</b>
<b>7. Reflektion</b>	<b>15</b>
<b>Abbildungsverzeichnis</b>	<b>17</b>
<b>Quellenverzeichnis</b>	<b>18</b>
<b>Anhang</b>	<b>19</b>
Vorbereitung	19
Sensorbeschreibungen	22
Arbeitsblatt	24
Musterlösung	27

# 1. Einleitung

Sensoren, die Türen öffnen, Heizungen steuern, Brandmelder oder Autos regulieren und somit autonomes Fahren ermöglichen, sind nur einige Beispiele, wie die Technik immer mehr in die verschiedenen Bereiche des Lebens Einzug findet. Es wird immer wichtiger, schon in frühen Kindertagen den Umgang, die Möglichkeiten und das Verständnis für den Umgang mit Technik zu schulen. Nicht zuletzt um Gefahren und Vorteile der Technik abschätzen zu können. Die rasanten Entwicklungen in dem Bereich erfordern zudem gute Weiterbildungsmöglichkeiten und ausgebildete Lehrkräfte, die diese Entwicklungen aufgreifen und begreifbar machen können. Leider sind viele Lehrkörper, die in diesem Bereich unterrichten, noch immer fachfremd und haben ggf. selbst Berührungsängste oder wissen nicht, wie sie das Thema der Digitalisierung der Gesellschaft angemessen aufbereiten sollen [Klemm, 2015].

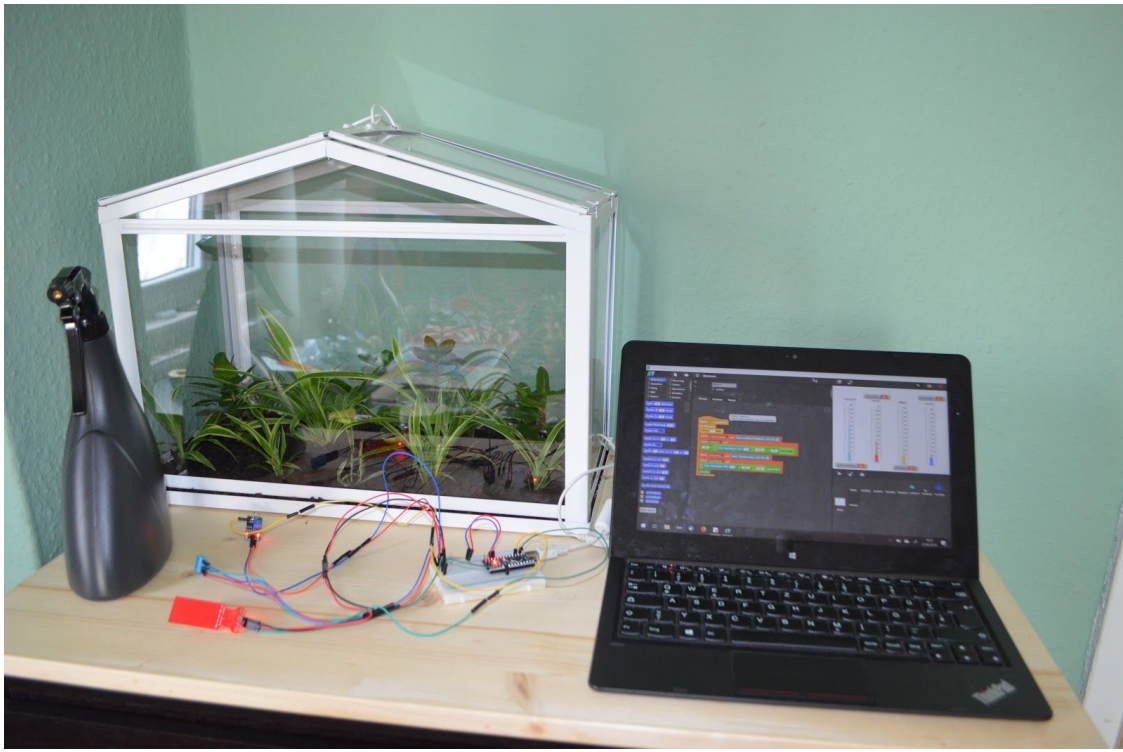
Seit 2015 findet sich unter anderem explizit das Themengebiet Physical-Computing im Berliner Rahmenlehrplan für Informatik wieder [Bildungsserver Berlin-Brandenburg, 2015a]. Dieser gibt eine Richtung vor, mit der man mit Schülerinnen und Schülern das Thema angehen kann und wie man in fächerübergreifenden Unterrichtseinheiten sowohl den Umgang mit der Technik, als auch fachspezifische Inhalte vermitteln kann. Durch günstige Preise und die - mittlerweile - große Auswahl an Mikrocontrollern stehen immer mehr Möglichkeiten bereit, innerhalb der Schule eigene Projekte zu entwickeln, zu programmieren und durchzuführen, die diese Ansprüche abdecken.

Das in dieser Hausarbeit beschriebene Projekt "Arduino-Gewächshaus" soll hierbei einerseits eine Hilfestellung bieten, Lehrkräften die Möglichkeiten des fächerübergreifenden Unterrichts mit Physical-Computing näher zu bringen und aufzuzeigen. Andererseits soll dies für die bereits fachlich ausgebildeten Lehrkräfte eine Idee und einen Anreiz für weitere Projekte bieten, ohne von Grund auf alles selbst vorbereiten zu müssen.

Im Folgenden werden wir das interdisziplinäre Projekt zwischen dem Nawi- und Informatikunterricht genauer beschreiben und eine genauere Eingrenzung in die Themenfelder des Rahmenlehrplans der 6./7. Klasse diskutieren. Anschließend werden die für das Projekt benötigten Komponenten näher erläutert. Dies soll das bessere Verständnis der nachfolgenden Projektumsetzung ermöglichen. Die an dieser Stelle beschriebene Projektdurchführung wird dabei um Hinweise zu den im aktuellen Schritt zu vermittelnden Kenntnisse ergänzt. Es folgt die Beschreibung einiger Erweiterungsmöglichkeiten, die an dem Projekt vorgenommen werden können. Zum Schluss werden die Erprobung durch die Studierenden des Seminars beschrieben, sowie das Projekt und die Erprobung reflektiert.

Zusätzlich dazu werden im Anhang zu dieser Arbeit ausführliche Materialien zur Vorbereitung des Projektes, sowie die zwei Arbeitsblätter zur Durchführung inklusive deren Musterlösung, bereitgestellt.

## 2. Beschreibung des Projektes



*Abb. 1 - Ein beispielhafter Projektaufbau*

Die Schüler und Schülerinnen lernen anhand eines kleinen, selbst bestückten Gewächshauses den Lebensraum und die Eigenschaften von Pflanzen kennen. Dazu gehören z.B. der Wasserbedarf der Pflanzen, die Anpassungsfähigkeit der Pflanzen an ihre Umgebung, sowie das Grundprinzip der Photosynthese. Um diese Eigenschaften besser erklären und verdeutlichen zu können, werden den Schülerinnen und Schülern in Gruppen je ein Arduino, sowie verschiedene Sensoren bereitgestellt, welche die Schüler und Schülerinnen selbst mittels der Software *Snap4Arduino* (S4A) programmieren sollen. Dabei sollen sich die Schülerinnen und Schüler selbstständig überlegen, welche Sensoren sinnvoll sind, wo diese angebracht werden und wie diese ausgelesen werden. Die Schüler und Schülerinnen müssen nun die Ergebnisse interpretieren und eventuell eine geeignete Darstellung für die ausgelesenen Werte finden. Ein beispielhafter Projektaufbau wird dabei in Abb. 1 dargestellt.

Das Gewächshaus bietet dabei eine gute Möglichkeit, die Sensorwerte vergleichend zu messen. Ein Beispiel wäre z.B. die Luftfeuchtigkeit oder die Temperatur innerhalb und außerhalb des Gewächshauses. Bereits bei diesen Werten wird ein sichtbarer Unterschied deutlich.

Im Folgenden wird auf die benötigten Materialien sowie S4A eingegangen.

### 2.1 Materialien

Dieser Abschnitt bietet einen kurzen Überblick über die einzelnen Materialien und deren Zusammenspiel. Dazu werden die meisten Komponenten ausführlicher beschrieben. Da jedoch einige

Komponenten nachfolgend keine eigene Beschreibung erhalten, wird hier eine kurze Zusammenfassung aller verwendeten Materialien bereitgestellt.

Folgende Materialien werden mindestens einmal für die ganze Klasse benötigt:  
Gewächshaus, Blumenerde, Pflanzen, Sprühflasche mit Wasser

Folgende Materialien werden mindestens einmal für jede Schülergruppe benötigt:  
Arduino Nano V3.0, Steckbretter, Set Jumper-Kabel, Set Sensoren (z.B. Licht-, Flüssigkeits- bzw. Tropfen-, Temperatur-, Luftfeuchtigkeitssensor), 1x 10k $\Omega$ -Widerstand, Computer mit *Snap4Arduino*

### 2.1.1 Gewächshaus



*Abb. 2 - Das Gewächshaus, beispielhaft bepflanzt mit Grünstilben und flammenden Kästchen*

Zur Durchführung des Projektes wird ein kleines Zimmergewächshaus benötigt, wie in Abb. 2 beispielhaft dargestellt. Hierbei sollte darauf geachtet werden, dass es sich von mindestens einer Seite öffnen lässt, ohne den Korpus vollständig abnehmen zu müssen, um Werte wie die erhöhte Luftfeuchtigkeit oder eine höhere Temperatur innerhalb des Gewächshauses messen zu können.

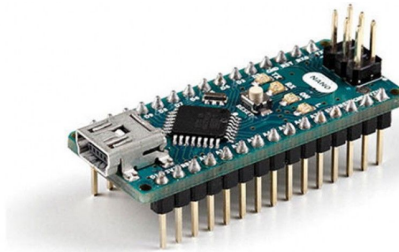
Das von uns verwendete Gewächshaus hatte die Möglichkeit, den Korpus zur Bepflanzung zu entfernen und anschließend wieder zu montieren. Für die weitere Pflege der Pflanzen kann anschließend das Dach geöffnet werden.

Das Gewächshaus sollte dabei mit ausreichend Blumenerde und pflegeleichten Pflanzen bestückt sein, die auch kurzzeitig mit einer eventuellen Überwässerung auskommen können. Andernfalls empfiehlt es sich, Ersatzblumenerde bereitzuhalten.

### 2.1.2 Sprühflasche

Um in kurzer Zeit die Luft- und Bodenfeuchtigkeit erhöhen zu können, empfiehlt es sich eine mit Wasser gefüllte Sprühflasche bereitzustellen. Durch diese können die Schülerinnen und Schüler bei Bedarf durch einen feinen Sprühnebel das Gewächshaus bewässern. Auf diese Weise wird zudem ein zu schnelles und unvorsichtiges Überwässern weitestgehend vermieden.

### 2.1.3 Arduino Nano V3.0



*Abb. 3 - Der Arduino Nano V3.0*

Quelle: <https://store.arduino.cc/usa/arduino-nano>, 08.07.18

Der Arduino Nano V3.0 ist ein sogenannter Mikrocontroller (siehe Abb. 3). Mikrocontroller sind Kleinstcomputer, die über Ein- und Ausgänge in Form von Pins verfügen. Über diese können sie an weitere Geräte wie z.B. Sensoren, Motoren, Bildschirme oder LEDs angeschlossen werden. Dabei lässt sich der Mikrocontroller frei durch den Benutzer über bereitgestellte Schnittstellen programmieren. Zur Programmierung und zum Bespielen des Mikrocontrollers ist in der Regel eine Verbindung zu einem Computer notwendig. Über diese Verbindung wird zudem der Austausch zwischen den Sensoren und anderen Softwareanwendungen auf dem Computer ermöglicht.

Wird eine externe Stromquelle für den Mikrocontroller bereitgestellt und dieser einmalig programmiert, kann der Mikrocontroller anschließend das Programm eigenständig wiederholen, ohne den Austausch mit dem Computer zu benötigen. Somit können eigenständige, interaktive Objekte erstellt und gesteuert werden. Da der Arduino nur geringe Spannungen leitet und verträgt, sollte darauf geachtet werden, dass bei Benutzung keine Stromversorgung verwendet wird, die 5 Volt übersteigt, um einen Schaden am Arduino zu vermeiden.

Der Arduino Nano verfügt über 12 (D2 - 13) digitale sowie 8 analoge Pins (A0-7), welche sowohl als Eingang als auch als Ausgang verwendet werden können. Dabei leiten digitale Pins nur Signale der Werte **1** und **0**, also Spannung angelegt oder nicht. Analoge Pins können hingegen den Spannungsunterschied messen und auf diese Weise einen Bereich lesen, der durch Werte zwischen 0 und 1023 interpretiert werden kann. Je geringer dabei die ankommende Spannung ist, desto kleiner ist der jeweilige Wert.

Um Bauteile mit Strom zu versorgen, besitzt der Arduino zudem mehrere Pins zur Stromversorgung (5V und 3.3V) sowie "geerdet" Pins (GND) zum Schließen des Stromkreises. Neben konstanten Spannungen kann dabei auch Pulse Width Modulation, d.h. eine ständig zwischen 0 und 5 Volt auf- und absteigende Spannung, zur Steuerung der Komponenten angelegt werden. Dies kann z.B. als Taktfrequenz genutzt werden.

Die weiteren Pins besitzen ebenfalls Funktionalitäten, sind jedoch nicht für das Projekt von Relevanz und werden somit nicht beschrieben. Eine ausführliche Beschreibung findet sich auf der offiziellen Seite von Arduino (<https://store.arduino.cc/usa/arduino-nano>).

### 2.1.4 Steckbrett und Jumper-Kabel

Steckbretter können dazu verwendet werden, um den Mikrocontroller ohne zu löten mit den verschiedenen Sensoren zu verbinden. Somit ist eine flexible Verwendung der bereitgestellten Komponenten möglich und die Wiederverwendung der Bauteile für weitere Versuche gegeben.

Um die Distanzen zwischen dem Mikrocontroller und den einzelnen Sensoren bzw. Komponenten zu überbrücken, können zudem Jumper-Kabel verwendet werden. Jumper-Kabel sind z.B. isolierte Kupferkabel mit männlichen und / oder weiblichen Enden. Dabei sind die Enden so verstärkt, dass durch ein einfaches Stecksystem die Kabel mit anderen Komponenten verbunden werden können. Sind genug Kabel vorhanden, können die Sensoren auch direkt an den Arduino angeschlossen werden, sofern keine Widerstände oder andere Komponenten zwischen den Sensor und den Mikrocontroller geschaltet werden müssen. Sobald jedoch Widerstände oder andere Komponenten zur Beschaltung notwendig sind, sollte ein Steckbrett verwendet werden.

## 2.1.5 Sensoren

### Luftfeuchtigkeits- und Temperatursensor

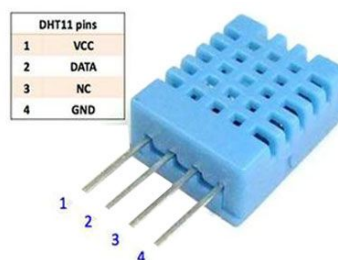


Abb. 4 - Der DHT11 Sensor zum Messen der Luftfeuchtigkeit und Temperatur

Quelle: [http://www.hobbyandyou.com/content/images/thumbs/0002133\\_dht11-digital-temperature-and-humidity-sensor-sensor-arduino.jpeg](http://www.hobbyandyou.com/content/images/thumbs/0002133_dht11-digital-temperature-and-humidity-sensor-sensor-arduino.jpeg), 08.07.18

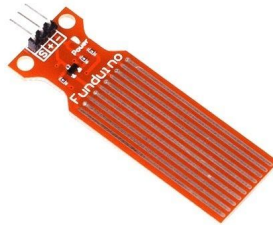
Der in Abb. 4 dargestellte Sensor DHT11 bietet die Möglichkeit, mit dem Arduino die Luftfeuchtigkeit im Bereich von ca. 20-95% mit einer Genauigkeit von 5% und die Temperatur im Bereich von ca. 0-50°C mit einer Genauigkeit von 2°C zu messen.

Die Kommunikation mit dem Arduino erfolgt über einen digitalen Pin. Dieser Pin dient sowohl als Ein- als auch als Ausgang. Der Arduino schickt über den entsprechenden Pin zuerst eine Anfrage an den Sensor, anschließend ändert sich die Einstellung des Pins und der Sensor kann über diesen Daten an den Arduino übermitteln. Da es sich um einen digitalen Sensor handelt, werden die Temperatur und Luftfeuchtigkeit mittels 8-Bit-Auflösung übertragen, d. h. es werden pro Wert 8 Signale über den digitalen Pin übermittelt. Da über digitale Pins nicht die eingehende Spannung gemessen werden kann, sondern nur die Signale HIGH (1) und LOW (0) gesendet werden können, wird der entsprechende Wert in einer binären Darstellung übermittelt. Der übermittelte Wert kann anschließend mittels der Spezifikation des Sensors in die entsprechende Temperatur oder Luftfeuchtigkeit übersetzt werden.

Da dies aufwändig zu implementieren ist, werden zur Verwendung des Sensors verschiedene Bibliotheken bereitgestellt, die direkt in die Arduino-IDE eingebunden werden können und alle notwendigen Funktionalitäten bieten (<https://github.com/adafruit/DHT-sensor-library>). Auch mehrere Portierungen für *Snap4Arduino* stehen zum Download bereit. [ITEAD Wiki, 2015]



## Flüssigkeits- bzw. Tropfsensor



*Abb. 5 - Der Flüssigkeits- bzw. Tropfsensor*

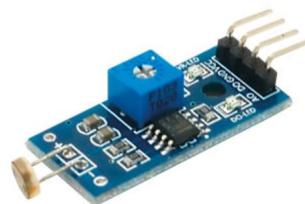
Quelle: [https://www.funduinoshop.com/WebRoot/Store14/Shops/78096195/5468/BEEE/65F5/CFDD/4BCB/C0A8/2BB9/E4A9/Tropfsensor\\_arduino\\_m.jpg](https://www.funduinoshop.com/WebRoot/Store14/Shops/78096195/5468/BEEE/65F5/CFDD/4BCB/C0A8/2BB9/E4A9/Tropfsensor_arduino_m.jpg), 08.07.18

Mit einem Flüssigkeits- bzw. Tropfsensor kann man Flüssigkeiten detektieren. Der in Abb. 5 gezeigte Sensor ist dazu mit mehreren Kontaktstellen ausgestattet, welche als glänzende Streifen erkennbar sind. Die Kontaktstellen haben dabei alternierend eine positive oder negative Spannung. Gelangt nun eine Flüssigkeit an den Sensor, werden die Kontaktstellen miteinander verbunden und der Stromkreis wird geschlossen. Dadurch entsteht eine Spannung am Ausgang des Sensors. Die Spannung wird dabei größer, je stärker der Sensor mit einer Flüssigkeit benetzt ist.

Die Verbindung zum Arduino erfolgt über einen analogen Pin. Auf diese Weise ist es dem Arduino möglich, die eingehende Spannung zu messen und diese auf einen entsprechenden Wert zwischen 0 (0V) und 1023 (5V) abzubilden.

Wichtig ist dabei, dass der Sensor nur über einen kurzen Zeitraum Messungen im Sekundenbereich durchführen sollte, um Schäden durch Elektrolyse am Sensor zu vermeiden. Ist also eine Verwendung über einen längeren Zeitraum geplant, sollte das Messintervall bei ca. 15 min liegen, um diesen Schäden vorzubeugen. [Funduino GmbH, o.J.]

## Helligkeitssensor



*Abb. 6 - Der Helligkeitssensor*

Quelle: [https://www.reichelt.de/?ARTICLE=224222&PROVID=2788&gclid=EAIaIQobChMI8bK0956S3AIVDdwZCh2Nrw96EAQYAyABEgLOXfID\\_BwE](https://www.reichelt.de/?ARTICLE=224222&PROVID=2788&gclid=EAIaIQobChMI8bK0956S3AIVDdwZCh2Nrw96EAQYAyABEgLOXfID_BwE)

Ein Helligkeitssensor, wie in Abb. 6 dargestellt, gibt Auskunft über den Lichteinfall an der Stelle, an der er angebracht ist. Dazu wird beim obigen Sensor das links über die Platine hängende Bauteil, ein Fotowiderstand, verwendet. Der Fotowiderstand ändert seinen Widerstand in Abhängigkeit von der einfallenden Lichtstärke. Je dunkler die Stelle, an welcher der Sensor angebracht ist, desto höher ist der Widerstand, den der Fotowiderstand bereitstellt. Nimmt die Helligkeit zu, sinkt der Widerstandswert des Fotowiderstandes. Dieser Effekt wird zusammen mit einem in Reihe geschalteten Widerstand dazu genutzt, die Helligkeit bzw. Dunkelheit anhand der am Fotowiderstand anliegenden Spannung zu ermitteln. Bei einer Reihenschaltung teilt sich die Gesamtspannung auf die Widerstände auf und nach dem Ohmschen Gesetz fällt dabei die höchste Spannung am größten



Widerstand ab. Das heißt, dass die 5V-Spannung, die vom Arduino bereitgestellt wird, im Stromkreis des Helligkeitssensors auf den Fotowiderstand und auf den verbauten Widerstand aufgeteilt wird. Also gilt: Je mehr Licht auf den Fotowiderstand fällt, desto geringer ist dessen Widerstand und somit auch die an ihm abfallende Spannung.

Um die am Fotowiderstand anliegende Spannung zu messen, wird der Sensor an einen analogen Pin des Arduinos angeschlossen. Der Arduino wandelt daraufhin den gemessenen Spannungswert in eine Zahl zwischen 0 (0V) und 1023 (5V) um. Der Wert 0 entspricht somit der höchsten messbaren und der Wert 1023 der geringsten messbaren Helligkeit. [Coding World UG, o.J.]

## 2.2 Snap4Arduino

*Snap4Arduino* ist eine Erweiterung der visuellen Programmiersprache *Snap!* zur Verwendung mit den Arduino Boards [Romagosa and Guillén, o.J.]. *Snap!* basiert auf der Programmiersprache Scratch und eignet sich daher besonders gut für die Einführung in die Informatik bzw. die ersten Erfahrungen in der Arbeit mit Arduinos. Dabei wurde *Snap!* um einige Features erweitert, so dass auch komplexere Strukturen wie z.B. Listen von Listen möglich werden (siehe [National Science Foundation, o.J.]).

Wie auch in Scratch erfolgt bei *Snap4Arduino* die Programmierung der Mikrocontroller durch visuelle Blöcke, die wie Puzzleteile ineinander gesteckt werden können. Durch das Puzzleteil-Prinzip passen nur solche Teile ineinander, die auch zusammen funktionieren. Dadurch werden von vornherein einige nicht funktionierende Verbindungen und somit eine unnötige Frustration in der Programmierung umgangen.

Im Folgenden wird eine kurze Erklärung der Programmoberfläche gegeben, weitere Informationen finden sich jedoch auf dem Internetauftritt des Programms.

### 2.2.1 Benutzeroberfläche

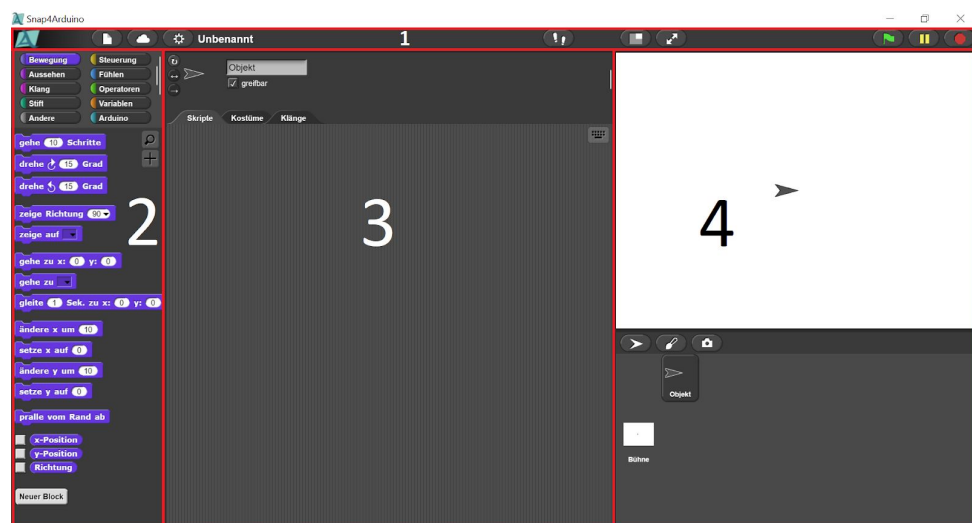






Abb. 7 - Benutzeroberfläche von Snap4Arduino nach dem Starten des Programms

Die in Abb. 7 dargestellte Benutzeroberfläche von *Snap4Arduino* lässt sich in 4 Bereiche unterteilen. Eine Menüleiste (1), über die die Programme geladen und gestartet werden können, die Block-Bibliothek (2), welche die Blöcke zur Programmierung eines Programmes bereitstellt, den

Programmbereich (3), in dem die Blöcke zur Ausführung zusammengesetzt werden und den Zeichenbereich (4), auf welchem verschiedene Daten und Objekte dargestellt werden können. Im Folgenden wird eine genauere Beschreibung der einzelnen Bereiche gegeben.

## 2.2.2 Menüzeile

Die Menüzeile enthält alle wichtigen Funktionen zum Erstellen, Laden und Speichern neuer *Snap4Arduino*-Projekte sowie der Konfiguration der IDE. Zudem lässt sich hier der Modus “Programmausführung verfolgen”  aktivieren, welcher einen Debugger darstellt. Hier kann der Ablauf der einzelnen Schritte im laufenden Programm nachvollzogen werden.

Auch das Starten , Pausieren  und Anhalten  der Programmausführung aller mit den entsprechenden Steuerungs-Blöcken versehenen Programme kann hier mit einem Klick gesteuert werden.

## 2.2.3 Block-Bibliothek

Da es sich bei *Snap4Arduino* um keine textuelle, sondern um eine visuelle Programmiersprache handelt, werden entsprechende Elemente benötigt, die visuell verbunden werden können. Diese finden sich in der Block-Bibliothek in verschiedenen Kategorien wieder. Dabei ist dem Benutzenden die Möglichkeit gegeben, in jeder Kategorie eigene Blöcke zu definieren, welche Funktionalitäten im Programmablauf übernehmen - ein Beispiel hierfür sei das Auswerten des Datenstroms des DHT11-Sensors, welches sonst nicht korrekt möglich wäre.

*Snap4Arduino* unterscheidet dabei in folgende Blockgruppen:

**Bewegung** - Mit Bewegungs-Blöcken kann die Zeichenfläche des Tools verändert werden. Man kann den darauf dargestellten Objekten somit eine Richtung und weitere Informationen mitgeben, damit diese in die entsprechende Richtung wandern. Ein Beispiel dafür ist das Zeichnen mittels eines Stiftes. Blöcke der Gruppe Bewegung werden dunkelblau dargestellt.

**Aussehen** - Ein Block der Gruppe „Aussehen“ ermöglicht es, die visuelle Gestaltung der Zeichenfläche zu beeinflussen. Man kann Objekte, Variablen oder anderweitige Ausgaben erscheinen oder verschwinden lassen. Zudem kann das Aussehen von Objekten angepasst, sowie die Zeichenreihenfolge der Objekte verändert werden. Aussehens-Blöcke werden lila dargestellt.

**Klang** - Um Sound durch die Programme ausgeben lassen zu können, wie z.B. Hintergrundmusik, können Klang-Blöcke benutzt werden. Blöcke der Gruppe „Klang“ sind dabei rosa / pink dargestellt.

**Stift** - Stift-Blöcke werden dazu verwendet, um durch das Programm eigene Linien zeichnen zu lassen. Man kann z.B. die Farbe der Linien bestimmen, sowie die verwendete Stärke. Um den Stift bewegen zu können, müssen die Stift-Blöcke mit Bewegungsblöcken kombiniert werden. Ein Stift-Block ist anhand seiner dunkelgrünen Farbe zu erkennen.

**Andere** - Die Gruppe „Andere“ steht dem Benutzer zur Verfügung, wenn er einen selbstdefinierten Block erstellen möchte, der sich in keine der anderen Gruppen einordnen lässt. Blöcke dieser Gruppe werden grau dargestellt.

**Steuerung** - Die gelben Blöcke der Gruppe Steuerung dienen dazu, den Programmablauf zu beeinflussen. Zum einen wird hier die Endlosschleife zur dauerhaften Ausführung der Programme bereitgestellt, zum anderen Kontrollstrukturen wie bedingungs-basierte Ausführung, andere Schleifen und die Startsignale des Programms.

**Fühlen** - Die Kategorie “Fühlen” umfasst jene Blöcke, die Tastatureingaben abfangen oder Interaktionen mit der Zeichenfläche auswerten können. Zudem können Links oder andere interaktive Objekte auf der Zeichenfläche dargestellt werden. Dieser Gruppe zugehörige Blöcke sind an ihrer hellblauen Farbe zu erkennen.

**Operatoren** - Die Gruppe der Operatoren ermöglicht es Berechnungen durchzuführen. Dazu gehört z.B. die Addition zweier Zahlen, aber auch die Ermittlung des Unix-Codes eines Zeichens. Zudem wird hier die Möglichkeit eröffnet, Javascript zu programmieren und ausführen zu lassen. Blöcke dieser Gruppe werden hellgrün dargestellt.

**Variablen** - Variablen können frei durch den Benutzer hinzugefügt werden. Jede Variable wird dabei als eigener Block dargestellt. Weiterhin können hier Operationen, die auf Variablen ausgeführt werden können, d.h. Zuweisungen von Werten o.ä., gefunden werden. Auch Listen werden hier zur Verfügung gestellt, welche mehrere Werte auf einmal enthalten können. Mit Variablen können zudem Werte zwischen verschiedenen Objekten und deren Programmen ausgetauscht werden. Ein Beispiel ist hier das Einlesen eines Sensorwertes in eine Variable durch den Arduino und das Auslesen der Variable durch ein Zeichenprogramm, das diesen Wert grafisch auf der Zeichenfläche darstellt. Variablen-Blöcke sind an ihrer orangen oder dunkelroten Farbe zu erkennen.

**Arduino** - Die Kategorie Arduino ermöglicht es, Arduino Boards mit *Snap4Arduino* zu verbinden bzw. zu trennen und enthält zudem alle Blöcke, die für das Auswerten der Arduino-Daten zuständig sind. Darunter fallen z.B. Funktionen wie das Auslesen analoger und digitaler Pins oder das Senden von Werten über analoge bzw. digitale Pins. Blöcke dieser Farbe werden türkis dargestellt.

## 2.2.4 Programmbereich

Um eine Programmsequenz für ein Objekt bereitzustellen, müssen die für das Programm benötigten Blöcke aus der Block-Bibliothek in den Programmbereich gezogen und zusammengesetzt werden. Es ist sogar möglich, mehrere Programmsequenzen pro Objekt bereitzustellen. Zudem wird hier die Möglichkeit gegeben, den einzelnen Objekten Namen zu geben, sowie deren Aussehen durch Kostüme - also eigene Grafiken - oder deren Sounds durch das Aufnehmen eigener Geräusche zu ändern.

### 2.2.5 Zeichenbereich

Der Zeichenbereich, oder auch Bühne genannt, wird dazu verwendet, um die gewünschten Objekte, Werte und Schaltflächen des Programms darzustellen. Dafür wird der Zeichenbereich in zwei Hälften unterteilt. Unten findet sich die Bühne, sowie die dort darzustellenden Objekte des Programms. Die Objekte werden dort erstellt und ausgewählt. Anschließend können dem ausgewählten Objekt über den Programmbereich das entsprechende Verhalten, Aussehen und Sounds zugewiesen werden.

Im oberen Teil des Zeichenbereichs findet sich hingegen zu Beginn eine weiße Fläche mit einem schwarzen Pfeil. Diese kann jedoch - je nach verwendeten Objekten und deren Kommunikation - angepasst werden und stellt die aktuelle Programmoberfläche dar. Dort werden Werte ausgegeben, Interaktionen vorgenommen und z.B. andere Grafiken dargestellt oder selbst gezeichnet.

## 3. Einordnung in den Rahmenlehrplan

Das Projekt "Arduino-Gewächshaus" richtet sich an 6./7. Klässler mit sehr geringem bis keinem Vorwissen in der Informatik und der Programmierung.

Das Projekt ist für den interdisziplinären Unterricht zwischen dem Nawi-Unterricht und der Informatik angedacht. Die Schüler und Schülerinnen nutzen in dem Projekt eine Vielzahl von Messgeräten, betrachten die Aussagekraft von ihren Messwerten und entwickeln selbstständig eine Skala zur einfacheren Interpretation der Messwerte. Diese Inhalte werden im Rahmenlehrplan Berlin 2015 für das Fach Nawi in das Themenfeld "3.1 Von den Sinnen zum Messen" aufgeführt.

Aus informatischer Sicht wird in dem Projekt vor allem auf die Kompetenz "2.2 Mit Informationen umgehen" aus dem Berliner Rahmenlehrplan 2015 hin gearbeitet. Es werden Variablen als Speicherort von Daten eingeführt und von den Schülerinnen und Schülern selbst erstellt. In folgenden Aufgaben kann anschließend auf die verschiedenen Datentypen für Texte, Zahlen und Wahrheitswerte eingegangen werden. Des Weiteren wird bei dieser Einführung in die Programmierung bereits das EVA-Prinzip (Eingabe-Verarbeitung-Ausgabe, Themenfeld "3.2 Informatiksysteme") angewandt. Die Schüler und Schülerinnen verarbeiten die Sensormessdaten und lassen sich diese anschließend grafisch ausgeben.

Erweitert man den Projektaufbau, um einen CO<sub>2</sub>-Sensor, so könnte alternativ mit dem Gewächshaus und den Sensoren auch der Treibhauseffekt (Themenfeld "3.3 Vom ganz Kleinen und ganz Großen" im Rahmenlehrplan 2015) im Nawi-Unterricht verdeutlicht werden. Uns fehlte lediglich die Zeit, den weiteren Sensor zu integrieren.

Wird das Projekt in einem größeren zeitlichen Rahmen als automatisiertes Gewächshaus durchgeführt, so können zusätzlich Inhalte aus dem Themenfeld "3.5 Pflanzen, Tiere, Lebensräume" behandelt werden. Die Schülerinnen und Schüler lernen anhand eines kleinen, selbst bestückten Gewächshauses den Lebensraum und die Wachstumsbedingungen von Pflanzen kennen.

Die Lernziele des Projektes sind die folgenden:

Die Schüler und Schülerinnen ...

- können die grundlegende Funktionsweise eines Feuchtigkeits- und eines Helligkeitssensors beschreiben.
- können die Aussagekraft von Messwerten für Außenstehende bewerten.
- können Messwerte von beliebigen Messgeräten mittels einer Prozentskala angeben.
- können Sensordaten aus einem Arduino mittels *Snap4Arduino* auslesen.

## 4. Projektumsetzung

### 4.1 Vorbereitung

Damit das Projekt und die Aufgaben im Unterricht durchgeführt werden können, muss durch die Lehrkraft die entsprechende Vorbereitung des Projektes durchgeführt werden. Zum einen müssen die Materialien besorgt werden, zum anderen sind einige wenige Schritte hinsichtlich des Arduino Nanos V3.0 notwendig, damit dieser zu *Snap4Arduino* kompatibel ist. Eine detaillierte Schritt-für-Schritt-Anweisung findet sich im Anhang im Abschnitt "Vorbereitung".

### 4.2 Durchführung

Die Durchführung des Projektes ist in einer Doppelstunde in der Klasse 6/7 angedacht.

Bevor das Projekt begonnen wird, sollte die Lehrkraft eine kurze Motivation geben, weshalb dieses Projekt durchgeführt wird und welche Relevanz dieses Projekt in der Realität hat. Eine solche Einführung könnte wie folgt aussehen:

"Sensoren umgeben uns in unserem Alltag fast überall und dienen der Vermessung der Umwelt. Anfängen von Assistenzsystemen in Autos, über Sensoren in unseren Smartphones bis hin zu Sensoren in Kühlschränken und anderen Haushaltsgeräten, trifft man Sensoren in nahezu allen Lebensbereichen an. Wir wollen uns heute mit einigen Sensoren und deren Messfähigkeiten beschäftigen, um ein Überwachungssystem für ein Gewächshaus zu entwickeln, das die Temperatur, die Luftfeuchtigkeit, die Bodenfeuchtigkeit und die Sonneneinstrahlung ausgibt."

Eventuell kann anschließend als Ausblick auf Erweiterungsmöglichkeiten folgendes hinzugefügt werden:

"In einem weiteren Projekt könnte das Überwachungssystem dann zu einem automatisierten Gewächshaus ergänzt werden, in dem die Helligkeit und Bodenfeuchtigkeit automatisch nach den Ansprüchen der Pflanzen angepasst werden."

Im nächsten Schritt sollte eine kurze Einführung in die verwendeten Sensoren und den Arduino Nano erfolgen. Dazu können die Beschreibungen im Anhang "Sensorbeschreibungen" z.B. zur Präsentation durch die Lehrkraft oder zur Erarbeitung und Präsentation durch Gruppen von Schülerinnen und Schülern genutzt werden.

Nun haben die Schüler und Schülerinnen die Voraussetzungen, um in Gruppen das Arbeitsblatt "Der erste Sensor" (siehe Anhang) zu bearbeiten. Auf diesem Arbeitsblatt wird den Schülerinnen und Schülern die Programmierumgebung *Snap4Arduino* vorgestellt und sie lernen in dieser Umgebung ihr

erstes kleines Programm zum Auslesen eines Sensormesswertes zu schreiben. Abschließend sollen die Schüler und Schülerinnen eine Ausgabe der Messwerte entwickeln, die auch für einen Außenstehenden ohne Vorwissen zu den Ausgabewerten der Sensoren interpretierbar ist (empfohlen wird auf dem AB die Angabe in Prozent).

Die entstandenen Programme sollten anschließend kurz von den Gruppen präsentiert werden, wobei darauf Wert gelegt werden sollte, dass zumindest eine Gruppe die Berechnung von Prozentwerten präsentiert, da diese Berechnung für das folgende Arbeitsblatt benötigt wird.

Im letzten Teil des Projektes bearbeiten die Schülerinnen und Schüler das Arbeitsblatt “Zusammenführung der Sensoren” (siehe Anhang), in dem sie in einem neuen Projekt mit grafischer Ausgabe der Sensormesswerte nun die Werte aller Sensoren auslesen sollen.

Zum Abschluss des Projektes sollen die Schüler und Schülerinnen mit den Sensoren experimentieren, um mögliche Positionierungen und Einsatzzwecke der Sensoren im Gewächshaus zu bestimmen.

Die Ergebnisse sollten abschließend im Plenum besprochen werden, sofern dies der zeitliche Rahmen noch erlaubt.

## 5. Erweiterungen

Neben der bereits erwähnten inhaltlichen Erweiterung zum Thema Treibhauseffekt, kann ein LCD Bildschirm zur Anzeige der Messwerte genutzt werden. Auch weitere Sensoren, die an den Arduino angeschlossen werden, sind möglich. Hier könnte man eine Vielzahl von Sensoren, wie z.B. einen CO2-Sensor oder ein Gyrometer verwenden und in das Projekt integrieren.

Ist ein größerer zeitlicher Rahmen gegeben, kann dieses Projekt auch zu einem automatisierten Gewächshaus erweitert werden (ähnlich zu [Stern, o.J.]).

Dazu könnten über die Messwerte der Sensoren eine lange LED, eine Wasserpumpe, ein Luftbefeuchter und ein Servomotor (zum Öffnen der Klappe) gesteuert werden, um die optimale Beleuchtung, Bewässerung, Luftfeuchtigkeit und Temperatur für die verwendeten Pflanzen sicherzustellen.

Eine Alternative zum automatischen Gewächshaus wäre zudem, das Gewächshaus mit den vorgestellten Sensoren (oder einigen Erweiterungen) über einen längeren Zeitraum im Klassenverband stehen zu haben und die Sensorwerte auf eine SD-Karte oder anderweitig regelmäßig abzuspeichern. So kann die Entwicklung der Werte über einen längeren Zeitraum analysiert werden.

## 6. Erprobung durch Studierende

Im Zuge des Seminars Physical Computing an der Humboldt Universität zu Berlin im Sommersemester 2018 haben wir unser Projekt mit Studierenden der Informatik erprobt. Wir haben den Studierenden dazu folgende Materialien bereitgestellt: Einen Arduino Nano mit angeschlossenem Feuchtigkeits-, Helligkeits- und DHT11 -Sensor, einen Laptop, der per USB-Kabel mit dem Arduino Nano verbunden war, ein bepflanztes Gewächshaus und unsere Arbeitsblätter.

Die Studierenden hatten rund 15 Minuten Zeit, um sich die grobe Funktionsweise der verwendeten Sensoren anhand unserer Beschreibungen zu erschließen, die Arbeitsblätter so weit wie möglich zu bearbeiten und anschließend mit den Sensoren zu experimentieren.

Mithilfe des ersten Arbeitsblattes sollten sich die Studierenden mit der Programmierumgebung *Snap4Arduino* vertraut machen, die Daten eines ersten Sensors auslesen und zuletzt die Ausgabe der Messwerte in eine Form bringen, die auch für Außenstehende sofort interpretierbar wäre.

Zusammen mit dem zweiten Arbeitsblatt wurde für die Studierenden ein Projekt bereitgestellt, in dem bereits eine grafische Darstellung der Sensorwerte mittels verschiedenfarbiger "Thermometer" vorbereitet war. Zudem waren die Variablen für die Temperatur, die Luftfeuchtigkeit, die Helligkeit und die Bodenfeuchtigkeit vorgegeben und die Aufgabe der Studierenden war nun das Auslesen der Messwerte und gegebenenfalls die Umrechnung der Messwerte in Prozentwerte.

Abschließend konnten die Studierenden die Sensoren innerhalb und außerhalb des Gewächshauses benutzen und die Veränderung der Sensormesswerte am Computer beobachten.

## 7. Reflektion

Die Studierenden schienen viel Spaß bei der Bearbeitung der Aufgaben und insbesondere bei den Experimenten mit den Sensoren zu haben. Nach Ablauf der 15 Minuten blieben beide Gruppen selbst nach Aufforderung zum Wechsel noch länger sitzen, um die Aufgaben zu beenden, beziehungsweise noch weitere Tests mit den Sensoren durchzuführen.

In einer Feedbackrunde mit den Studierenden wurden das mit Pflanzen ausgestattete Gewächshaus und die vielen Sensoren als anregend und motivierend beschrieben. Es wurde positiv aufgenommen, dass die Schülerinnen und Schüler auf dem ersten Blatt eine kleine Wahlmöglichkeit haben und die Aufgabenstellungen und Erklärungen wurden als klar und gut strukturiert bezeichnet.

Als problematisch wurde von den Studierenden der hohe Zeitaufwand eingeschätzt, da die Bearbeitung nach ihren Einschätzungen mehr als 45 Minuten in Anspruch nehmen würde. Dem stimmen wir zu, jedoch sehen wir darin nicht unbedingt ein Problem, da dieses Projekt aus unserer Sicht gut in einer Doppelstunde behandelt werden kann.

Ein weiterer Kritikpunkt waren die kurzen Kabel von den Sensoren zum Arduino Nano, durch die das Messen innerhalb des Gewächshauses schwierig war. Für einen nächsten Aufbau/Nachbau würden wir deshalb empfehlen, die Kabel ausreichend lang zu wählen, so dass die Sensoren im Gewächshaus verwendet werden können, ohne dass der Arduino dabei gleichzeitig in der Luft gehalten werden muss.

Zuletzt wurde kritisiert, dass auf unseren Sensor-Beschreibungen ein Touchsensor aufgeführt war, den wir schlussendlich nicht mehr in unser Projekt integriert haben. Das Fehlen des Sensors hat zu einer kleinen Irritation der Studierenden geführt.

Bei der Durchführung dieses Projektes sollte also darauf geachtet werden, dass ausschließlich die für das Projekt verwendeten Sensoren besprochen/aufgeführt werden.



Insgesamt würden wir die Erprobung als erfolgreich bewerten, da ohne jegliche Beschreibung allein das Gewächshaus bereits großes Interesse an dem Projekt hervorgerufen hat und das Feedback, abgesehen von einigen kleinen Verbesserungsvorschlägen, durchweg positiv ausfiel.

# Abbildungsverzeichnis

Abb. 1: Ein beispielhafter Projektaufbau	4
Abb. 2: Das Gewächshaus, beispielhaft bepflanzt mit Grünstilben und flammenden Käthchen	5
Abb. 3: Der Arduino Nano V3.0	6
Abb. 4: Der DHT11 Sensor zum Messen der Luftfeuchtigkeit und Temperatur	7
Abb. 5: Der Flüssigkeits- bzw. Tropfensensor	8
Abb. 6: Der Helligkeitssensor	8
Abb. 7: Benutzeroberfläche von Snap4Arduino nach dem Starten des Programms	9

# Quellenverzeichnis

[Bildungsserver Berlin-Brandenburg, 2015a] Bildungsserver Berlin- Brandenburg: *Rahmenlehrplan für die Jahrgangsstufen 7–10 der Berliner und Brandenburger Schulen. Teil C: Informatik. Wahlpflichtfach. Jahrgangsstufen 7–10.* 2015. – URL [http://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche\\_Fassung/Teil\\_C\\_Informatik\\_2015\\_11\\_10\\_WEB.pdf](http://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche_Fassung/Teil_C_Informatik_2015_11_10_WEB.pdf). – Zugriffsdatum: 08.07.18.

[Bildungsserver Berlin-Brandenburg, 2015b] Bildungsserver Berlin- Brandenburg: *Rahmenlehrplan für die Jahrgangsstufen 5–6 der Berliner und Brandenburger Schulen. Teil C: Naturwissenschaften. Jahrgangsstufen 5–6.* 2015. – URL [https://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche\\_Fassung/Teil\\_C\\_Nawi\\_5-6\\_2015\\_11\\_16\\_web.pdf](https://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche_Fassung/Teil_C_Nawi_5-6_2015_11_16_web.pdf). – Zugriffsdatum: 08.07.18.

[Bildungsserver Berlin-Brandenburg, 2015c] Bildungsserver Berlin- Brandenburg: *Rahmenlehrplan für die Jahrgangsstufen 7–10 der Berliner und Brandenburger Schulen. Teil C: Naturwissenschaften. Wahlpflichtfach. Jahrgangsstufen 7–10.* 2015. – URL [https://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche\\_Fassung/Teil\\_C\\_Nawi\\_7-10\\_2015\\_11\\_16\\_web.pdf](https://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche_Fassung/Teil_C_Nawi_7-10_2015_11_16_web.pdf). – Zugriffsdatum: 08.07.18.

[Coding World UG, o.J.] Coding World UG: LDR - Fotowiderstand - *Wie kann denn ein Foto Widerstand leisten?* – URL <http://codetrainer.de/de/trainer/view/220/228> – Zugriffsdatum: 08.07.18.

[Funduino GmbH, o.J.] Funduino GmbH: Nr.18 Tropfensensor. – URL <https://funduino.de/nr-17-tropfensensor> – Zugriffsdatum: 08.07.18.

[ITEAD Wiki, 2015] ITEAD Wiki: DHT11 Humidity Temperature Sensor Brick – URL [https://www.itead.cc/wiki/DHT11\\_Humidity\\_Temperature\\_Sensor\\_Brick#Communication\\_Process:\\_Serial\\_Interface\\_28Single-Wire\\_Two-Way.29](https://www.itead.cc/wiki/DHT11_Humidity_Temperature_Sensor_Brick#Communication_Process:_Serial_Interface_28Single-Wire_Two-Way.29) – Zugriffsdatum: 08.07.18.

[Klemm, 2015] Klemm, Klaus. "Lehrerinnen und Lehrer der MINT-Fächer: Zur Bedarfs- und Angebotsentwicklung in den allgemein bildenden Schulen der Sekundarstufen I und II am Beispiel Nordrhein-Westfalens." *Gutachten im Auftrag der Deutsche Telekom Stiftung* (2015): 1-13.

[National Science Foundation, o.J.] National Science Foundation: About Snap!. – URL <https://snap.berkeley.edu/about.html> – Zugriffsdatum: 08.07.18.

[Romagosa and Guillén, o.J.] Bernat Romagosa and Joan Guillén: Arduino goes lambda! – URL <http://snap4arduino.rocks> – Zugriffsdatum: 08.07.18.

[Stern, o.J.] Felix Stern: Automatisches Raspberry Pi Gewächshaus selber bauen. – URL <https://tutorials-raspberrypi.de/automatisches-raspberry-pi-gewaechshaus-selber-bauen/> – Zugriffsdatum: 08.07.18.

# Anhang



## Vorbereitung

Um die Arduino Nanos V3.0 für *Snap4Arduino* vorbereiten zu können, installieren Sie bitte die offizielle ArduinoIDE sowie *Snap4Arduino* auf Ihrem Computer. Danach müssen folgende Schritte ausgeführt werden:

1. Öffnen Sie das im ZIP-Archiv zu diesem Paper befindliche Arduino-Projekt "DHT11\_StandardFirmata.ino" mit der ArduinoIDE.
2. Installieren Sie die DHT11-Bibliothek von Adafruit
  - a. Laden Sie die Bibliothek von <https://github.com/adafruit/DHT-sensor-library> als ZIP-Datei herunter.
  - b. Gehen Sie in dem zuvor geöffneten Arduino-Projekt auf Sketch → Bibliothek einbinden → .ZIP-Bibliothek hinzufügen und wählen Sie die eben heruntergeladene ZIP-Datei aus.
3. Verbinden Sie nun den Arduino Nano V3.0 mit dem Computer und prüfen Sie den Sketch.
4. Sollten keine Probleme auftreten, laden Sie den Sketch auf den Arduino hoch.
5. Nun sollte der Arduino Nano V3.0 mit *Snap4Arduino* kommunizieren können. Um dies zu testen, sind folgende Schritte notwendig:
  - a. Öffnen Sie *Snap4Arduino* und wählen Sie die Block-Kategorie "Arduino" aus.
  - b. Wählen Sie nun "Mit Arduino verbinden". Wird nun der Arduino erkannt und erfolgreich eine Verbindung aufgebaut, war das Aufspielen des Sketches erfolgreich und der Arduino kann mit *Snap4Arduino* verwendet werden. Falls die Kommunikation fehlschlägt, wiederholen Sie die vorherigen Schritte erneut.
6. Wiederholen Sie die Schritte 3-5 für alle Arduinos, die an Schülergruppen bzgl. des Projektes ausgegeben werden.

Bei "DHT11\_StandardFirmata.ino" handelt es sich um eine angepasste Version der standardmäßig in der ArduinoIDE enthaltenen Datei "StandardFirmata.ino". Die Datei wurde dahingehend angepasst, dass der DHT11-Sensor durch spezielle Blöcke für *Snap4Arduino* ausgelesen werden kann. Die Anpassungen wurden dabei auf Basis des Projektes Robotics-Unleashed von chrisclaynz vorgenommen (<https://github.com/chrisclaynz/Robotics-unleashed>). Dabei wurden die Blockdefinitionen für *Snap4Arduino* übernommen und ins Deutsche übersetzt.

Anschließend installieren Sie *Snap4Arduino* auf den entsprechenden Computern, die durch die Schülerinnen und Schüler verwendet werden sollen. Um hier nun die Projektvorlagen verwenden zu können, die in der dem Paper zugehörigen ZIP-Datei im Ordner "Aufgaben" bereitliegen, gehen Sie wie folgt vor:

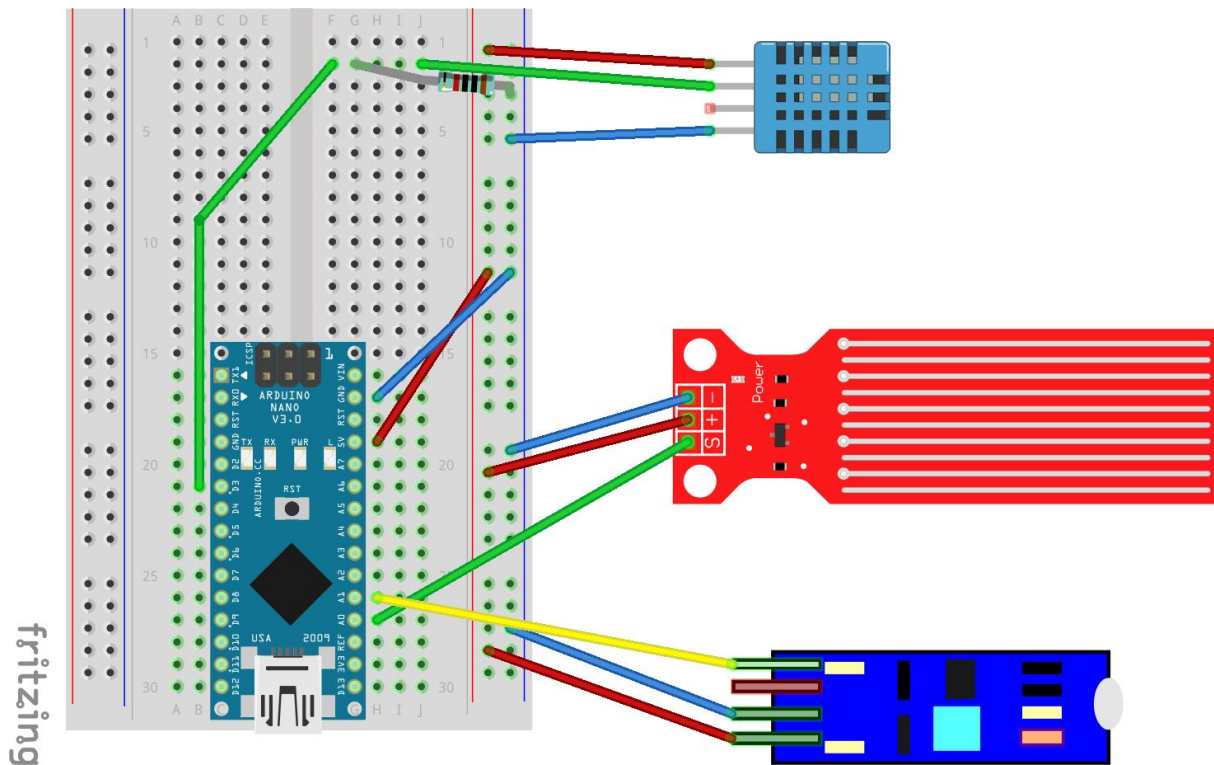
1. Starten Sie *Snap4Arduino* und wählen Sie  → Importieren... und wählen Sie eine der Dateien "Aufgabe\_Der\_erste\_Sensor.xml", "Aufgabe\_Zusammenführung\_der\_Sensoren.xml" und "Musterlösung.xml" aus.
2. Nachdem der Import erfolgreich durchlaufen wurde, gehen Sie auf  → Sichern...
3. Wiederholen Sie diese Schritte für alle drei Projekte.

Auf diese Weise werden alle notwendigen Daten, wie z.B. die Blöcke zum Auslesen des DHT11, die Grafiken und die vordefinierten Programmblöcke zum Updaten der Grafiken etc. in das Projekt geladen. Die Grafik des Thermometers basiert auf einer Wikimedia Commons Grafik ([https://commons.wikimedia.org/wiki/File:Thermometer\\_C.svg](https://commons.wikimedia.org/wiki/File:Thermometer_C.svg)). Zur Bearbeitung der Aufgaben können die Schüler und Schülerinnen nun das entsprechende Projekt laden und das Arduino-Objekt um die geforderte Funktionalität erweitern.



*Beispielhafte Zeichenroutine des Thermometerbalkens für die Luftfeuchtigkeit*

Der Code für das Arduino-Objekt wird dabei im Anhang "Musterlösung" ausführlich erklärt. Daher wird nur kurz auf die Zeichenroutine der Thermometerbalken eingegangen, hier exemplarisch am Beispiel des Luftfeuchtigkeitssensors. Das Programm startet dabei damit, dass die Zeichenfläche von allen vorher gezeichneten Objekten bereinigt wird - also evtl. bestehende Balken gelöscht werden. Anschließend werden Farbe und Stiftdicke konfiguriert. Damit der Balken kontinuierlich an die Werte angepasst werden kann, wird eine Dauerschleife benötigt. Zu Beginn der Schleife setzen wir die aktuelle Zeichenposition des Stiftes auf den Ausgangspunkt des Thermometers und geben an, in welche Richtung der Stift verschoben wird. Erst dann setzen wir den Stift auf der Zeichenfläche auf und berechnen die zu gehende Schrittzahl anhand des durch den Arduino in die Variable Luftfeuchtigkeit eingelesenen Wertes. Anschließend setzen wir den Stift wieder ab, damit wir den Zeichenvorgang wiederholen können. Hierbei sind die Zeichenroutinen für alle Thermometer gleich, lediglich in der ausgelesenen Variable, der Farbe und den Startkoordinaten unterscheiden sie sich.



*Steckplan des Steckbrettes für den Arduino und die Sensoren. Von oben nach unten sind der DHT11, der Flüssigkeitssensor sowie der Lichtsensor zu sehen.*

Nun bereiten Sie für jede Schülergruppe die Steckbretter entsprechend der oben gezeigten Abbildung vor. Achten Sie darauf, dass der Widerstand, der für den DHT11 benötigt wird, ein 10kΩ-Widerstand ist. Haben Sie auch diesen Schritt für jede Gruppe erledigt, sind alle technischen Vorbereitungen abgeschlossen.

Anschließend gilt es das / die Gewächshäuser vorzubereiten. Dieser Schritt kann jedoch auch im Rahmen der Projektdurchführung vorgenommen werden.

# Sensorbeschreibungen

## Arduino Nano v3.0

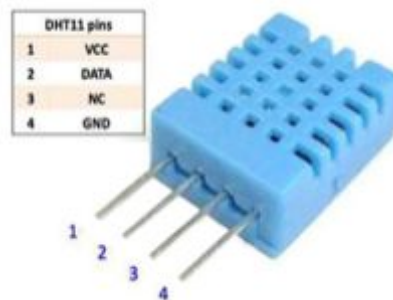


Quelle: <https://store.arduino.cc/usa/arduino-nano>

Der Arduino Nano ist ein Kleinstcomputer (auch Mikrocontroller genannt) mit Ein- und Ausgängen zum Anschluss von Geräten wie z.B. Sensoren, Motoren, Bildschirmen oder LEDs. Mit dem Arduino können eigenständige, interaktive Objekte gesteuert werden oder es können Daten mit Softwareanwendungen auf Computern ausgetauscht werden.

Der Arduino erhält über den USB-Anschluss Strom und kann gleichzeitig über diesen auch programmiert werden. Zum Anschluss von externen Geräten werden die Metallstifte (Pins) des Arduinos mit den Metallstiften der Geräte durch Kabel verbunden.

## Temperatur- und Luftfeuchtigkeitssensor (DHT11)



Quelle: [http://www.hobbyandyou.com/content/images/thumbs/0002133\\_dht11-digital-temperature-and-humidity-sensor-sensor-arduino.jpeg](http://www.hobbyandyou.com/content/images/thumbs/0002133_dht11-digital-temperature-and-humidity-sensor-sensor-arduino.jpeg)

Der Sensor DHT11 bietet die Möglichkeit, mit dem Arduino die Luftfeuchtigkeit im Bereich von ca. 20-80% (5% Genauigkeit) und die Temperatur im Bereich von ca. 0-50°C (2°C Genauigkeit) zu messen.



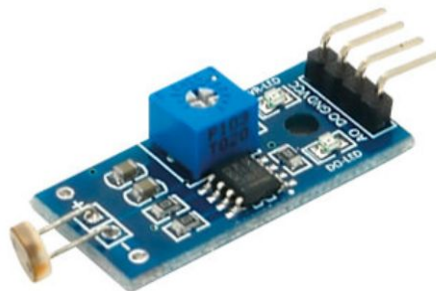
## Tropfensensor/Flüssigkeitssensor



Quelle: [https://www.funduinoshop.com/WebRoot/Store14/Shops/78096195/5468/BEEE/65F5/CFDD/4BCB/C0A8/2BB9/E4A9/Tropfensensor\\_arduino\\_m.jpg](https://www.funduinoshop.com/WebRoot/Store14/Shops/78096195/5468/BEEE/65F5/CFDD/4BCB/C0A8/2BB9/E4A9/Tropfensensor_arduino_m.jpg)

Mit einem Tropfensensor oder auch Flüssigkeitssensor kann man Flüssigkeiten detektieren. Das Ganze funktioniert wie folgt: An den langen Kontaktstellen, die den Sensor durchziehen, liegt eine Spannung an (entweder + oder -). Sobald eine Flüssigkeit, zum Beispiel durch einen Tropfen, zwei Kontakte verbindet, fließt ein kleiner Strom von einem Kontakt zum anderen. Dieser Strom wird an den Arduino weitergeleitet, der die Stromspannung in einen Zahlenwert umwandelt. 0 bis 5 Volt entsprechen dabei einem Zahlenwert von 0 bis 1023.

## Helligkeitssensor



Quelle: [https://www.reichelt.de/?ARTICLE=224222&PROVID=2788&gclid=EAlaIqobChMI8bK0956S3AIVDdwZCh2Nrw96EAQYAyABEgLOXfD\\_BwE](https://www.reichelt.de/?ARTICLE=224222&PROVID=2788&gclid=EAlaIqobChMI8bK0956S3AIVDdwZCh2Nrw96EAQYAyABEgLOXfD_BwE)

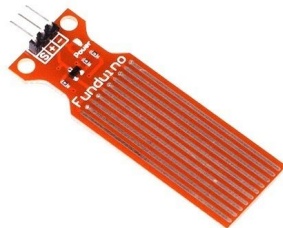
Im Helligkeitssensor wird ein Fotowiderstand verwendet, der im Bild rot markiert ist. Dieser ändert seinen Widerstand in Abhängigkeit von der Lichtstärke. Im Dunkeln ist der Widerstand hoch und er nimmt mit zunehmender Helligkeit ab. Diesen Effekt nutzt man, um die Helligkeit anhand der am Sensor anliegenden Spannung zu ermitteln. Denn je größer der Widerstand des Sensors ist, desto größer muss auch die daran anliegende Spannung sein, damit Strom fließen kann.

Der Arduino wandelt den gemessenen Spannungswert in eine Zahl um, die dann weiterverarbeitet werden kann. 0 Volt entspricht dabei der Zahl 0 und der höchsten messbaren Helligkeit. Der höchste Messwert von 5 Volt entspricht der Zahl 1023 und damit der niedrigsten Helligkeit.

# Arbeitsblatt

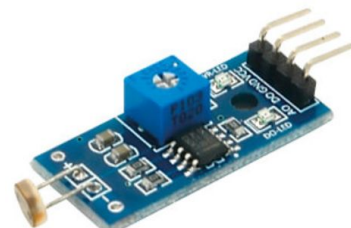
## Der erste Sensor

Wählt Euch **einen** der beiden unten stehenden Sensoren aus. Im Folgenden wollen wir dessen Werte auslesen und uns anzeigen lassen.



Feuchtigkeitssensor

Quelle: [https://www.funduinoshop.com/WebRoot/Store14/Shops/78096195/5468/BEEE/65F5/CFDD/4BCB/C0A8/2BB9/E4A9/Tropfensensor\\_arduino\\_m.jpg](https://www.funduinoshop.com/WebRoot/Store14/Shops/78096195/5468/BEEE/65F5/CFDD/4BCB/C0A8/2BB9/E4A9/Tropfensensor_arduino_m.jpg)

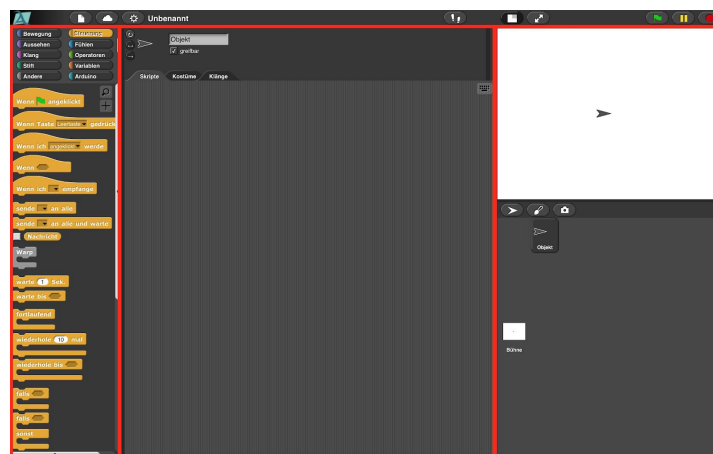


Lichtsensor

Quelle: [https://www.reichelt.de/?ARTICLE=224222&PROVID=2788&gclid=EAlalQobChMI8bK0956S3AIVDdwZCh2Nrw96EAQYAYABEgLOXfD\\_BwE](https://www.reichelt.de/?ARTICLE=224222&PROVID=2788&gclid=EAlalQobChMI8bK0956S3AIVDdwZCh2Nrw96EAQYAYABEgLOXfD_BwE)

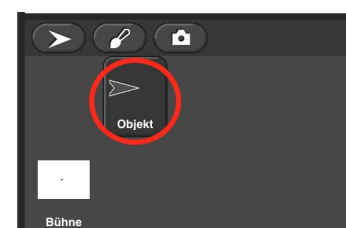
Zum Auslesen entwickeln wir mithilfe von *Snap4Arduino* ein kleines Programm.

1. **Öffnet** die Anwendung *Snap4Arduino*. Das geöffnete Fenster sollte ungefähr der unten stehenden Grafik entsprechen.



Auf der linken Seite befinden sich die Befehlsblöcke zum Programmieren, die in verschiedene Kategorien, wie z.B. Steuerung oder Fühlen unterteilt sind. Diese Befehlsblöcke werden in den mittleren Teil des Fensters bewegt, um das Programm zu schreiben. Auf der rechten, oberen Seite werden Euch Eure gemessenen Daten ausgegeben bzw. könnt Ihr in späteren Projekten hier eigene, bewegte Zeichnungen anfertigen.

2. Bevor wir mit der Programmierung beginnen, muss das aktuell offene Projekt mit dem Arduino verbunden werden. **Klickt** dazu **mit** einem **Rechtsklick** auf den Pfeil mit der Unterschrift "Objekt" und wählt "mit Arduino verbinden" aus.



- Wenn  angeklickt  
fortlaufend

- Drückt** nun die grüne Fahne zum Starten Eures Programms. Der Wert Eures Sensors sollte im weißen Teil des Fensters angezeigt werden.

- |  |
|--|
|  |
|  |
|  |

- (Tipp: Prozente)

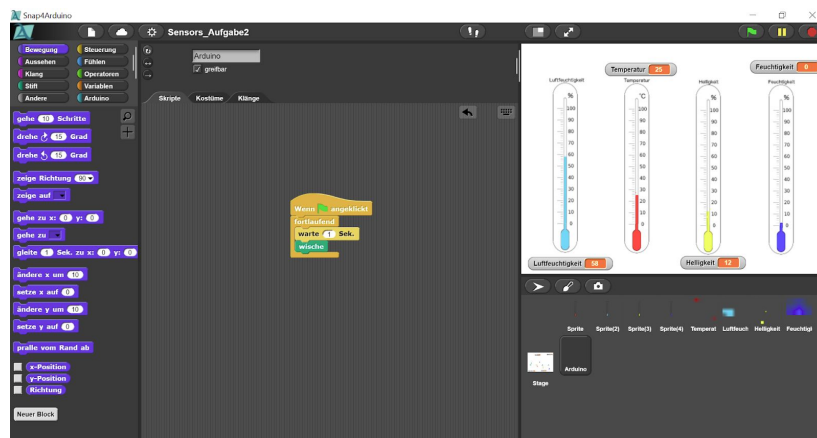
[illegible]

25

## Zusammenführung der Sensoren

Nun wollen wir die Werte aller drei Sensoren auslesen.

1. **Öffnet** dazu das teilweise fertige Projekt „Aufgabe\_Zusammenführung\_der\_Sensoren.xml“, indem Ihr auf das weiße Dokument links klickt, „Öffnen...“ auswählt und das Projekt „Aufgabe\_Zusammenführung\_der\_Sensoren“ auswählt. Euer Fenster sollte nun ungefähr der unten stehenden Grafik entsprechen.



2. Bevor wir mit der Programmierung beginnen, muss das aktuell offene Projekt wieder mit dem Arduino verbunden werden. **Klickt** dazu **mit einem Rechtsklick**, wie in Aufgabe 1 auf das Objekt „Arduino“ und wählt „mit Arduino verbinden“ aus.

**Setzt** in dieser Aufgabe bitte **alle Eure Blöcke zwischen** dem „**Warten**“-Block und dem „**Wischen**“-Block ein, da wir zwischen dem Auslesen der Daten nun kurz warten und die Anzeige leer wischen wollen.



In diesem Projekt sind die Variablen Temperatur, Luftfeuchtigkeit, Helligkeit und Feuchtigkeit bereits angelegt.

3. Eure Aufgabe ist es nun, die **Variablen** Helligkeit und Feuchtigkeit **auf den Prozentwert** des Messwertes zu **setzen**, wie in Aufgabe 7 des Arbeitsblatts „Der erste Sensor“. Der Sensor für die Helligkeit liegt am Pin 1 und der Sensor für die Feuchtigkeit liegt am Pin 0 an.
4. Lest nun die Daten für die **Temperatur-Variable (Pin 3)** und die **Luftfeuchtigkeits-Variable (Pin 3)** über die Blocks aus der Kategorie Arduino aus.

Nun könnt Ihr **mit den Sensoren experimentieren**, indem Ihr sie in und außerhalb des Gewächshauses messen lasst!

5. Der Helligkeitssensor zeigt jetzt bei hoher Helligkeit einen niedrigen Wert und bei Dunkelheit einen hohen Wert an. **Korrigiert** dieses Verhalten, indem Ihr die **Berechnung des Prozentwertes** in Eurem Programm **anpasst**!

## Musterlösung

Die hier dargestellten Musterlösungen sind jeweils nur beispielhaft für die Aufgaben. Aufgrund der vielen Möglichkeiten ist es nicht möglich, alle gültigen und funktionierenden Lösungen anzugeben.

### Der erste Sensor

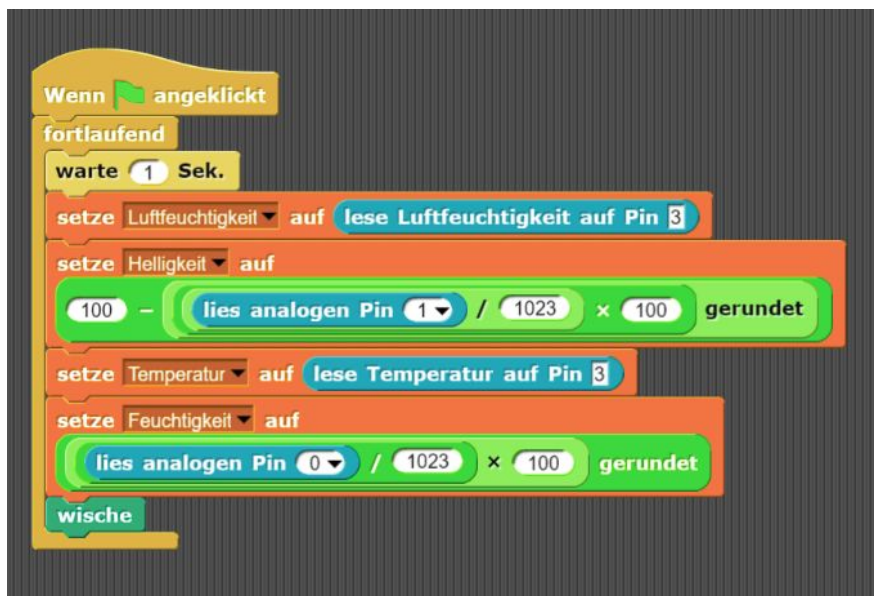


Zwei gültige Lösungsvorschläge für die Aufgabe "Der erste Sensor". Links: Lichtsensor, Rechts: Feuchtigkeitssensor

Die in der oberen Abbildung gezeigten Programme stellen eine minimale Lösung der ersten Aufgabe dar. Dabei ist links die Lösung für den Lichtsensor zu sehen und rechts die für den Feuchtigkeitssensor.

Beide Programme lesen fortlaufend den jeweiligen analogen Pin des Sensors aus und schreiben den Wert in die entsprechend für den Sensor definierte Variable. Diese Variable wird nun auf der Zeichenfläche ausgegeben und der Wert kann ausgelesen werden.

### Zusammenführung der Sensoren



Eine der gültigen Lösungsvorschläge für die Aufgabe "Zusammenführung der Sensoren"

Das oben abgebildete Programm liest jede Sekunde alle vier Sensorwerte nacheinander ein und speichert diese in der entsprechenden Variable, die für den Sensor bereitgestellt wurde.

Im Anschluss "wischt" das Programm die Zeichenfläche einmal leer, damit die Aktualisierung der Thermometerbalken im Zeichenbereich korrekt durchgeführt wird.

Da der digitale Sensor DHT11 bereits die korrekten Werte übermittelt, muss nur für die analogen Werte die entsprechende Umrechnung in Prozent vorgenommen werden. Hier ist bei dem Helligkeitssensor darauf zu achten, dass das Inverse des ermittelten Prozentwerts berechnet wird, da das Thermometer sonst nicht die prozentuale Helligkeit des Standortes, sondern die prozentuale Dunkelheit angeben würde. Weiterhin sollten beide analogen Werte gerundet werden, um so eine bessere Interpretierbarkeit der ermittelten Werte zu erreichen. Sollten bei dem Aufbau andere Pins verwendet werden, müssen diese entsprechend angepasst werden.