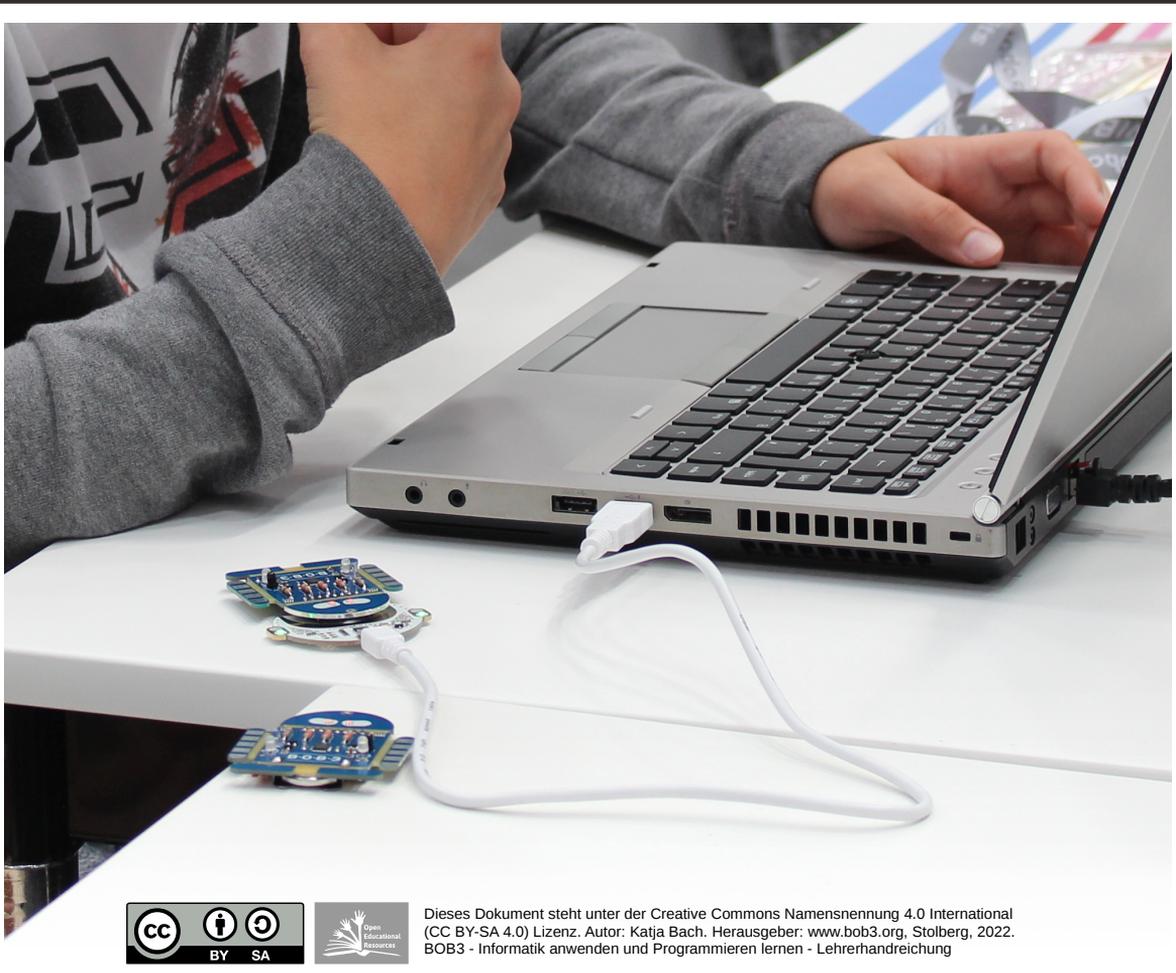


#Informatik

# BOB3

## Informatik anwenden und Programmieren lernen

Lehrerhandreichung – Sekundarstufe I



Dieses Dokument steht unter der Creative Commons Namensnennung 4.0 International (CC BY-SA 4.0) Lizenz. Autor: Katja Bach. Herausgeber: [www.bob3.org](http://www.bob3.org), Stolberg, 2022. BOB3 - Informatik anwenden und Programmieren lernen - Lehrerhandreichung

#Bildung in der digitalen Welt



# BOB3

## Informatik anwenden und Programmieren lernen

### Lehrerhandreichung

Version 12

Liebe Lehrerinnen und Lehrer,

die folgenden Seiten vermitteln Ihnen eine kompakte Grundlage für die praktische Durchführung des Unterrichts mit BOB3. Zusätzlich werden Details zu den Lernzielen, zu den angestrebten Kompetenzen und zum didaktischen Konzept dargestellt.

Da insbesondere im Bereich der Informationstechnologie eine große Heterogenität in den Vorkenntnissen der SuS vorhanden ist, war die Umsetzung der Wissensvermittlung in Form eines individuellen, differenzierten Unterrichts bei der Entwicklung ein zentrales Thema. Das BOB3-Programmier-Tutorial wurde so ausgelegt, dass es von den SuS selbstbestimmt, interaktiv und in individuellem Lerntempo bearbeitet werden kann. Insbesondere wurde darauf geachtet, dass schwächere Schüler die Motivation bei der Durchführung nicht verlieren, und dass leistungsstarken Schülern anspruchsvolle Aufgaben im Bereich der Vertiefungstutorials geboten werden. Da die Gruppe der SuS vom Tutorial angeleitet und motiviert wird, ergibt sich für die Lehrkraft die Möglichkeit, den gewonnenen Freiraum zur individuellen Förderung einzelner SuS einzusetzen. Weitere Informationen zum Konzept unter: [www.bob3.org/de/mint](http://www.bob3.org/de/mint)

#### **Bildungsziele:**

- Die SuS sollen die Bedeutung des Begriffs ‚Programmierung‘ verstehen und die Tätigkeit des ‚Programmierens‘ selbst durchgeführt haben.
- Die SuS sollen erkennen, welche Möglichkeiten sich durch Programmierung ergeben. Sie sollen die Potentiale Informationstechnischer Systeme erkennen und überlegen wo ihre Grenzen sind.
- Die SuS sollen lernen, wie Informationstechnologie entwickelt wird und welchen Einfluss diese auf die Gesellschaft ausübt.

- Die SuS sollen lernen wie man Strategien zur Problemlösung entwickelt und modelliert und sollen diese dann mittels Algorithmen und passenden digitalen Werkzeugen umsetzen.
- Die SuS sollen verstehen, was Programmcode bedeutet und wie man ihn verwendet. Spezielle Kontrollstrukturen wie bedingte Anweisungen und Schleifen-Konstrukte sollen erlernt werden.
- Leistungsstarke SuS sollen zusätzlich einen vertiefenden Einblick in die Prinzipien der Programmierung bekommen, ohne dabei zu stark auf eine spezielle Syntax, theoretische Konstrukte oder Algorithmen einzugehen - dies soll später systematisch im regulären Informatik-Unterricht geschehen, der dabei dann auf die erworbenen Grundkenntnisse aufbauen kann.
- Das Interesse der SuS an Technik, Mathematik & Informatik soll durch das Projekt geweckt und positiv verknüpft werden.

### **Technische Voraussetzungen:**

- PC-Arbeitsplätze mit Windows, Linux oder MacOS Betriebssystem oder iPads / Tablets und Internet-Zugang
- 1 ProgBob / BobDock und 1 BOB3 pro Arbeitsplatz
- Installiertes BobDude Übertragungsprogramm / installierte App

Hinweis: Bei der Durchführung in der Sekundarstufe I über 8 UE sollten die Lernziele von Intro I, Intro II und Intro III von den meisten SuS erreicht werden.



# Einführung: [1 UE]

## Voraussetzungen:

- SuS sollten rudimentäre Kenntnisse im Umgang mit einer Computer-Maus, einer Tastatur und in der Bedienung eines Web-Browsers haben.

## Lernziele:

- SuS sollen sich einen Account in der Web-Oberfläche anlegen und die Zugangsdaten notieren
- SuS sollen wissen unter welcher URL das Tutorial zu finden ist
- SuS sollen BOB3 kennen lernen

**Prozessbezogene Kompetenzen:** Bedienung der Oberfläche; Kenntnisse in der Bedienung eines Computers vertiefen

**Konzeptbezogene Kompetenzen:** Digitale Umgebungen und Werkzeuge zum persönlichen Gebrauch anpassen

**Soziale Kompetenzen:** gegenseitige Hilfestellung bei der Bedienung des Computers

## Vorbereitungen für die erste Unterrichtseinheit:

Benötigt werden:

- Ein Computer / iPad / Tablet pro Schüler (alternativ pro zwei SuS)
- Internetzugang + Webbrowser
- Ein Programmier-Helm „ProgBob“ pro Computer bzw. ein Programmieradapter „BobDock“ pro iPad
- Ein Roboter „BOB3“ pro Computer / iPad / Tablet



### Schritt 1:



Zunächst wird auf jedem Computer das Programm „**BobDude**“ installiert. Das Programm steht hier für alle Betriebssysteme (Windows, Mac OS X, Linux) kostenfrei zur Verfügung: <http://dude.bob3.org>

Auf iPads / Tablets wird die BOB3-App „**BobDude**“ installiert.

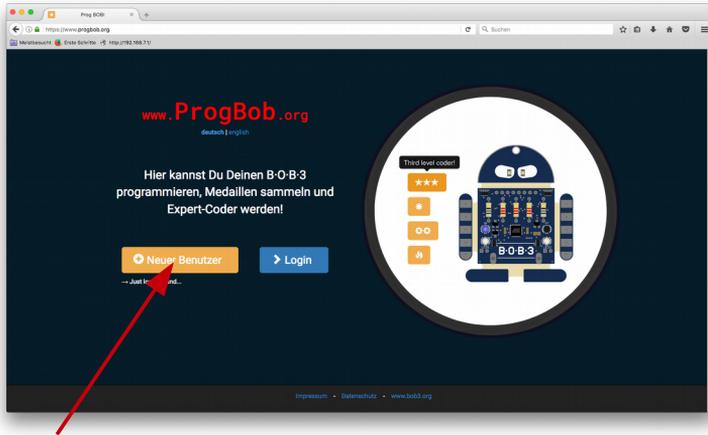
### Schritt 2:

Nach der Installation kann die LK einen **ersten Test** durchführen. Dazu wird der Programmier-Helm auf den BOB3 aufgesteckt und per USB-Kabel an den Computer angeschlossen. Bei dem Windows Betriebssystem wird nun automatisch der Gerätetreiber installiert, dies kann einige Minuten dauern. Sobald die LEDs am Helm **grün leuchten** ist der BOB3 einsatzbereit! Im Fall von iPads / Tablets reicht es, die App zu starten.

## Erste Unterrichtseinheit:

### → Erstellen der Benutzer-Accounts:

Die SuS starten den Webbrowser und gehen auf die Seite <http://www.ProgBob.org>:



Per Klick auf den Button „**Neuer Benutzer**“ öffnet sich der Dialog „*Neuen Account anlegen*“:

Die SuS denken sich jeweils einen eigenen **Benutzernamen** und ein **Passwort** aus.

#### **Benutzername:**

Beispiel: **LuckyLuke**,  
**BieneMaja** etc.  
(mindestens 4 Zeichen)

#### **Passwort:**

Beispiel: **Geheimnis13**,  
**Schueler27** etc.  
(mindestens 6 Zeichen)

#### **Email [optional]:**

Die Angabe einer Email-Adresse ist **optional** möglich: Falls man sein Passwort vergessen hat, kann es per Email angefordert werden.

**Tip:** Die SuS schreiben sich die Benutzerdaten auf.

# Tutorial „Intro I“: [1-3 UE, 13 Mikrolerneinheiten, 4 AB]

Intro I ✓

## Voraussetzungen:

- SuS sollten die Lernziele aus der Einführung erreicht haben

## Lernziele:

- SuS sollen die Web-Oberfläche der ProgBob Tutorials kennen lernen und bedienen können
- SuS sollen Programm-Quelltext selbstständig compilieren können und auf den Mikrocontroller übertragen
- SuS sollen zunächst vorgegebene kleine Änderungen am Quelltext vornehmen
- SuS sollen den Funktionsaufruf zur Veränderung der Farbe der LEDs kennen lernen und verstehen
- SuS sollen die `delay()` Funktion kennen lernen um Blinkeffekte mit den LEDs zu realisieren.
- SuS sollen jetzt selbstständig Änderungen am Quelltext vornehmen (Farbe der LEDs frei verändern, Parameter der `delay()`-Funktion, eigene Blinkroutinen)

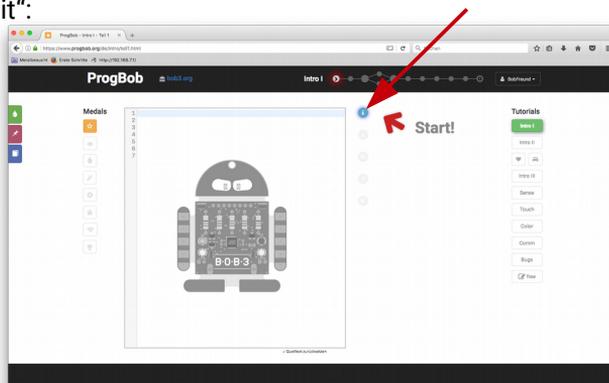
**Prozessbezogene Kompetenzen:** Bedienung der Oberfläche; compilieren eines Programms; Übertragung der Software auf den Roboter; Änderungen im Programmtext durchführen; Vertiefung der Kenntnisse in der Tastaturbedienung (Sondertasten, Groß-Kleinschreibung)

**Konzeptbezogene Kompetenzen:** Grundlegende Prinzipien und Funktionsweisen der digitalen Welt identifizieren, kennen und verstehen und bewusst nutzen; eine strukturierte algorithmische Sequenz zur Lösung eines Problems planen und verwenden

**Soziale Kompetenzen:** Gemeinsame Theoriebildung über die Funktionsweise des Roboters; gegenseitige Hilfestellung bei der Bedienung des Computers

## Starten des Tutorials:

Die SuS bestätigen den Dialog „Neuen Account anlegen“ mit „OK“ und gelangen so direkt zum Hauptfenster des Tutorials. Sie starten das Tutorial per Klick auf die erste „Info-Einheit“:



Das Tutorial ist **interaktiv** aufgebaut und **selbsterklärend**, so dass die SuS nun selbstständig und im jeweils eigenen Tempo arbeiten können.

 Die blauen „Info-Einheiten“ bieten Lerninhalte, z.B.:

**\_\_info\_\_**

 Jetzt lernen wir eine **neue** Art von **Bedingungen** kennen:

Im vorigen Beispiel haben wir zwei Zahlen auf **Gleichheit** überprüft. Jetzt wollen wir auf **größer als** und auf **kleiner als** prüfen:

`3 > 1`

`>` bedeutet "größer als"

`3 > 1` ist **wahr**

`1 > 4` ist **falsch**

 Die grauen „Aufgaben-Einheiten“ bieten Aufgaben, die von den SuS bewältigt werden müssen, z.B.:

**\_\_aufgabe\_\_**

 Ändere die Bedingung:

`100 > 1000`

 Compiliere das neue Programm und teste es auf deinem BOB3.

Was passiert jetzt?

 Die roten „Quiz-Einheiten“ dienen zur Wissensüberprüfung und müssen beantwortet werden, z.B.:

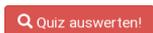
**\_\_quiz\_\_**

1: Wahr oder falsch: `5 < 4`  
 wahr  falsch

2: Wahr oder falsch: `5 != 4`  
 wahr  falsch

3: Wahr oder falsch: `5 != 5`  
 wahr  falsch

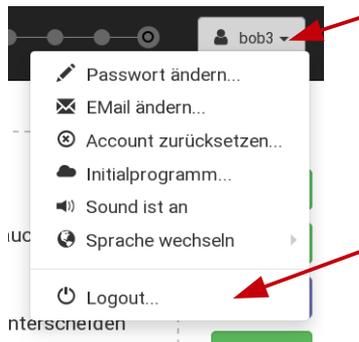
4: Wahr oder falsch: `1000 > -1000`  
 wahr  falsch

 Quiz auswerten!

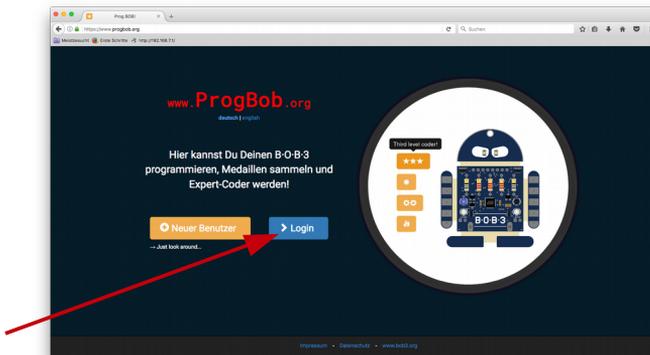
Per Klick auf den Button „Quiz auswerten!“ bekommen die SuS direktes Feedback zu ihrer Lösung.

## Ende einer Unterrichtseinheit:

Am Ende der Unterrichtseinheit klicken die SuS rechts oben auf den jeweiligen Benutzernamen und dann auf „Logout“.



Der **Lernstand** des einzelnen Schülers ist nun **gespeichert** und die SuS können in der nächsten Unterrichtseinheit über den Button „Login“ weiterarbeiten:



### Voraussetzungen:

- SuS sollten die Lernziele aus dem Intro I Tutorial erarbeitet haben

### Lernziele:

- SuS lernen Bedingungen wie 'gleich', 'ungleich' oder 'kleiner als' im Kontext einer Programmiersprache kennen
- SuS setzen die Bedingungen mit dem 'if/else' Konstrukt der Programmiersprache ein und experimentieren damit
- SuS lernen das Konzept einer (Ganzzahl-) Variablen kennen
- SuS speichern den Wert eines Sensors in einer Variablen und werten diese mit dem 'if/else' Konstrukt aus
- SuS lernen den IR-Reflex-Sensor und den Temperatur-Sensor kennen
- SuS erlernen den Sinn und Zweck einer 'for'-Schleife zur wiederholten Durchführung
- SuS lernen wie die Parameter der 'for'-Schleife eingesetzt werden können
- SuS lernen wie man Fehler im Programm-Code lokalisieren kann und wie man diese behebt

**Prozessbezogene Kompetenzen:** Suchen und Beheben von Fehlern im Programm

**Konzeptbezogene Kompetenzen:** Funktionsweise von analogen Sensoren, Bedingungen; Konzept von Verzweigungen, Ganzzahl-Variablen, Schleifen, Technische Probleme identifizieren; Probleme formalisiert beschreiben, Problemlösestrategien entwickeln und dazu eine strukturierte algorithmische Sequenz planen, diese auch durch Programmieren umsetzen und die gefundene Lösungsstrategie beurteilen

**Soziale Kompetenzen:** Gegenseitige Unterstützung bei der Suche nach Fehlern im Programmcode.

Zur Motivation der SuS werden im Anschluss die beiden optionalen Kurztutorials „Freundschaftstester“ und „Polizeiblinklicht“ angeboten.



## Voraussetzungen:

- SuS sollten die Lernziele aus dem Intro II Tutorial erarbeitet haben

## Lernziele:

- SuS vertiefen das Prinzip und die Anwendungsmöglichkeiten von Variablen
- SuS lernen die Touch-Sensoren als digitale Eingabeelemente kennen
- SuS verwenden Variablen, um die Sensorwerte der Touch-Sensoren abzuspeichern und während des Programmablaufs zur Verfügung zu stellen
- SuS erlernen als neue Kontrollstruktur die Verwendung des 'switch/case'-Konstrukts kennen und verwenden dies zur Fallunterscheidung
- SuS vertiefen ihr Wissen zur Kontrollstruktur ,if / else' anhand einer differenzierten Leistungsüberprüfung
- SuS erlernen des Sinn und Zweck von Funktionen und erarbeiten den Unterschied von Funktionen mit und ohne Parameter
- SuS erarbeiten den Unterschied von Funktionen mit und ohne Rückgabewert
- SuS erfahren wie eine selbstdefinierte Funktion angelegt werden kann und wie man diese aufruft
- SuS experimentieren mit Funktionsaufrufen zur Vereinfachung von wiederholten Abläufen

**Prozessbezogene Kompetenzen:** Verringerung des Programmieraufwands durch Verwendung von selbst-definierten Funktionen

**Konzeptbezogene Kompetenzen:** Funktionsweise von digitalen Sensoren; Konzept von Verzweigungen; Funktionsaufrufe; eine strukturierte algorithmische Sequenz zur Lösung eines Problems planen und verwenden

Zur Motivation der SuS werden im Anschluss die beiden optionalen Kurztutorials „Photo-Flash“ und „Alarmanlage“ angeboten.



# Vertiefungstutorials: [6-12 UE, 40 Mikrolerneinheiten, 2 AB]

## Voraussetzungen:

- SuS sollten die Lernziele aus dem Intro III Tutorial erarbeitet haben

## Lernziele:

- SuS lernen die verschiedenen Sensoren des Roboters im Detail kennen
- SuS experimentieren mit Arrays
- SuS lernen Funktionen mit Rückgabewerten kennen
- SuS lernen das Grundprinzip der Ereignisbehandlung anhand der Touch-Sensoren kennen
- SuS lernen die Grundlagen des RGB Farbmodells kennen
- SuS lernen die Grundlagen des HSV Farbmodells kennen
- SuS lernen die Grundlagen der Informationsübertragung zur Kommunikation zwischen technischen Geräten kennen

**Konzeptbezogene Kompetenzen:** Funktionsweise von Arrays; Farbmodelle zum Mischen von Farben; Funktionen mit Rückgabewerten; Algorithmische Muster und Strukturen in verschiedenen Kontexten erkennen, nachvollziehen und reflektieren

**Soziale Kompetenzen:** gemeinsames Arbeiten an einem Programm zur Übertragung von Informationen

Zusätzlich zu den einführenden Tutorials werden vertiefende Tutorials angeboten, die insbesondere die Experten der Klasse beschäftigen können. In diesen Tutorials werden einzelne Sensoren und Fähigkeiten des Roboters tiefergehend behandelt.

Sense ✓

Verstehen und Arbeiten mit dem IR-Sensor des Roboters, Experimente mit verschiedenen Lichtarten.

Touch ✓

Verstehen und Arbeiten mit dem Multifeld-Touch-Sensoren des Roboters, Lernen wie die Arme angesteuert und abgefragt werden können.

Color

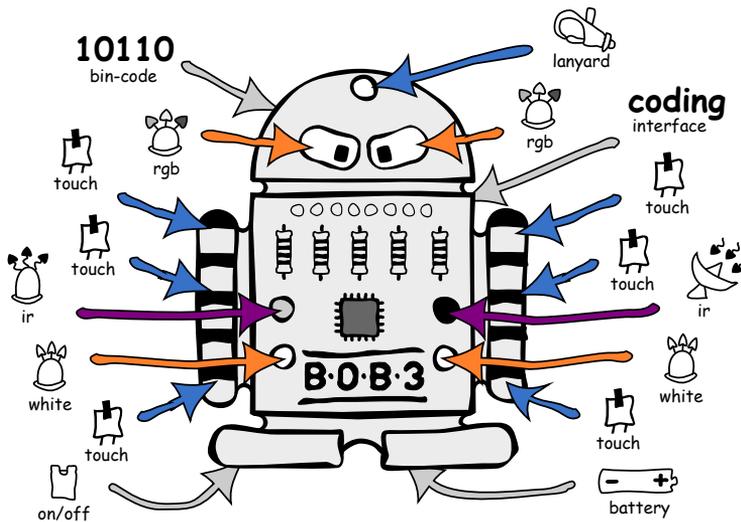
Erlernen des RGB- und des CMYK-Farbmodells. Wie entstehen eigentlich Farben? Experimente mit Farbüberblendeffekten. Verstehen und Anwenden anhand der beiden Multicolour-LEDs des Roboters.

Comm

Erlernen der drahtlosen Kommunikation zwischen zwei Roboter-Einheiten. Datenübertragung begreifbar machen anhand des „Alice und Bob“-Modells. Experimente mit Code-Übertragung. Verstehen und Anwenden anhand der IR-Sensorik des Roboters.

## BOB3 – Technische Übersicht

Von blinkenden LEDs zur drahtlosen Kommunikation - BOB3 ist ein kleiner Roboter der merkt, ob seine Arme berührt werden und wenn ja, wo! Er kann Freunde erkennen, seine weißen Scheinwerfer einschalten, seine Augen in allen Farben blinken lassen, nah und fern unterscheiden, drahtlos Botschaften senden und empfangen und einiges mehr! Die Schülerinnen und Schüler können ihn frei programmieren, ihm einen eigenen binären Code geben oder ihn mit einer Knopfzelle und dem beiliegenden Lanyard als blinkendes Gadget um den Hals tragen.



Die wichtigsten Elemente von BOB3 sind:



Augen-LEDs: können in allen Farben leuchten



Körper-LEDs: weiße, helle Scheinwerfer



Multifeld-Touch-Sensoren an den Armen, mit denen BOB3 auf Berührungen reagiert



IR-Sensor, mit dem BOB3 nah und fern unterscheiden kann

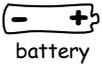


Computer-Gehirn: Ein kleiner Mikroprozessor der den Programmcode ausführt



on/off

Ein-/Aus-Schalter



Standard-Knopfzelle zum Herumtragen (während BOB3 programmiert wird, bekommt er den Strom über den USB-Anschluss)

**coding**  
interface

Die Anschlussbuchse für den ProgBob-Helm, damit BOB3 programmiert werden kann

**10110**  
bin-code

Persönlicher Binärcode zwischen 0 und 31, damit die SuS die Bobs in der Klasse unterscheiden können

Weiterhin verfügt BOB3 noch über:



Datenübertragung zwischen zwei BOB3 Einheiten:  
Infrarot Sender und Empfänger



Temperatursensor (im Mikroprozessor integriert)



lanyard

Ein Lanyard mit dem man sich BOB3 als blinkendes Gadget umhängen kann...

## Abschlusseinheit: [1 UE]

### Voraussetzungen:

- SuS sollten im Rahmen ihrer Entwicklungsstufe möglichst viele Einheiten bearbeitet haben

### Lernziele:

- SuS sollen die Parallelen von BOB3 zu anderen, alltäglichen Embedded Systemen erkennen und diskutieren
- SuS sollen sich Gedanken über die technischen Möglichkeiten und Einschränkungen von Computerprogrammen machen und diese gemeinsam reflektieren

**Konzeptbezogene Kompetenzen:** Einflüsse von Algorithmen und Auswirkung der Automatisierung von Prozessen in der digitalen Welt beschreiben und reflektieren

**Soziale Kompetenzen:** gemeinsames Diskutieren der Ergebnisse

Am Anfang dieser Einheit sollen die SuS zunächst reflektieren, welche neuen Erkenntnisse sie erworben haben und diese im Klassenverband diskutieren:

- Wie kommt das Programm auf den Roboter?  
[Compiler→BobDude→ProgBob→BOB3]
- Was ist eine Programmsequenz? [Die geplante Abfolge von einzelnen Anweisungen]
- Was ist eine Verzweigung? [if/else – in Abhängigkeit von einer Bedingung wird ein bestimmter Programmzweig ausgeführt]
- Was ist eine Schleife? [for – eine bestimmte Programmsequenz wird mehrfach ausgeführt]

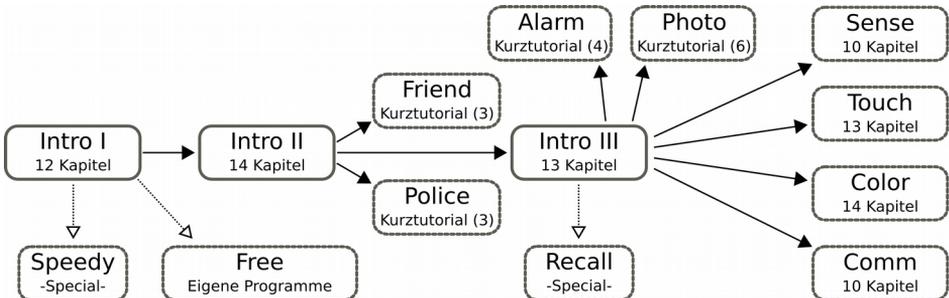
Anschließend sollen die SuS überlegen, in welchen alltäglichen Geräten heutzutage Computer-Systeme eingebaut sind, die man nicht direkt erkennt. Dies sind z. B.: Elektrische Küchengeräte, Waschmaschinen, Fernseher, Kühlschränke, digitale Fieberthermometer, Fensterheber in der Autotür oder Spielzeuge. Dabei sollen die SuS erkennen, dass für all diese Systeme spezielle Programme geschrieben wurden:

- Diskutiert was passieren würde, wenn man den Programmcode einer Waschmaschine fälschlicherweise auf den Mikrocontroller einer Spülmaschine übertragen würde.
- Überlegt in welchen alltäglichen Geräten ähnliche Mikrocontroller-Systeme eingesetzt werden.
- Überlegt welche Sensoren diese Geräte haben und erstellt daraus eine Liste. Vergleicht eure Listen in der Klasse.
- Überlege diesmal, welche Aktoren diese Geräte haben, erstellt daraus eine Liste und vergleicht diese in der Klasse.

Jetzt sollen die SuS überlegen, welche Parallelen es zwischen Embedded-Systemen wie dem BOB3 und Computern allgemein gibt, und anschließend reflektieren welche Möglichkeiten und Einschränkungen sie daraus ableiten können:

- Überlegt mit euren Erfahrungen, was BOB3 gut und was er nicht so gut kann.
- Vergleicht die Fähigkeiten und Einschränkungen von BOB3 mit normalen Computerprogrammen – was können diese gut und was nicht so gut.

## Lerninhalte / Überblick Mikrolerneinheiten:



Das Programmier-Tutorial ist in verschiedene Sub-Tutorials unterteilt, die **Grundlagen-Tutorials** (*Intro I*, *Intro II* und *Intro III*) sollen von allen Lernenden bearbeitet werden, die **Vertiefungs-Tutorials** (*Sense*, *Touch*, *Color* und *Comm*) sind zur inneren Differenzierung für leistungsstarke Schüler konzipiert. Im Folgenden wird ein Überblick über die Inhalte gegeben:

### Intro I

13 Kapitel

- Prinzip der Programmierung eines Embedded-Systems verstehen und anwenden
- Begriffsklärungen (Quellcode, Compilieren, Programm)
- Bedeutung der Programmier-Syntax erlernen
- Programmier-Oberfläche kennenlernen und verwenden
- Unterschiede und Verwendung der Funktionen `setup()` / `loop()` verstehen
- Roboter-LEDs ansteuern und Farben der LEDs verändern
- Ein einfaches Blinken als Sequenz mit der `delay()` Funktion erzeugen und Varianten entwickeln

### Intro II

18 Kapitel

- Wahrheitswerte verstehen und anwenden (`true` / `false`)
- Arbeiten mit mathematischen Bedingungen (`==`, `>`, `<`, `!=`, ...)
- Erlernen von Verzweigungen anhand der Programmstruktur „if / else“
- „if / else“ konkret zur Auswertung des IR-Sensors einsetzen
- Variablen kennenlernen und einsetzen
- Prinzip und Anwendung von for-Schleifen kennenlernen - Anwenden des zuvor Erlernen:
  - Ein-/Ausschalten der LEDs
  - `delay()` verwenden
  - Variablen
  - Bedingungsprüfungen
  - einfache Fehlersuche (Klammersetzung, Semikolon, ...)

### **Intro III** 12 Kapitel

- Sensoren entdecken, ansteuern und auswerten
- Variablen verwenden, um Sensorwerte abzuspeichern und zu verarbeiten
- switch / case Abfragen erlernen und anwenden
- Differenzierte Leistungsüberprüfung zur Kontrollstruktur ‚if / else‘
- Funktionen mit und ohne Parameter unterscheiden können
- Funktionen mit und ohne Rückgabewert unterscheiden können
- Verwendung von Funktionen mit Rückgabewert mittels Sensorik
- Eigene Funktionen definieren und deren Potential verstehen

Einheiten zur Vertiefung:

### **Sense** 10 Kapitel

- Verschiedene Experimente zur Analyse der Messergebnisse der IR-Sensorik

### **Touch** 13 Kapitel

- Ereignisbehandlung durch Auswertung der Multifeld-Touch-Sensoren

### **Color** 11 Kapitel

- Farbentheorie anhand der beiden Farbmodelle RGB und HSV, konkrete Experimente mit den Multicolor LEDs wie z.B. Farbüberblendeffekte

### **Comm** 10 Kapitel

- Kommunikation und Datenübertragung zwischen mehreren Roboter-Einheiten



## Kursverwaltung mit Lernstandskontrolle

- optional einsetzbar -

Die ProgBob Lernumgebung bietet speziell für Lehrerinnen und Lehrer eine Kursverwaltung an. Interessierte Lehrer können auf Anfrage ihren Benutzer-Account auf 'Kursleiter' upgraden, dazu bitte eine kurze E-Mail an [teacher@bob3.org](mailto:teacher@bob3.org) senden.

### Kursleitern stehen die folgenden Funktionen zur Verfügung:

- Zugang zu allen Tutorials für den Kursleiter freischalten
- Teilnehmer können ohne Angabe der persönlichen Daten im Block angelegt werden
- Passwörter der Teilnehmer können zurückgesetzt werden
- Übersicht über den Lernfortschritt der Teilnehmer
- Übersicht über die Zugangsdaten zum Ausdrucken und Verteilen vor dem ersten Termin

Als Kursleiter bekommt man im Benutzermenü zusätzlich den Menüpunkt „Kursverwaltung...“ angezeigt.



The screenshot shows the ProgBob interface. At the top, there is a dark header with the ProgBob logo, the URL bob3.org, and a 'Kursverwaltung' button. Below the header, the user's role is identified as 'Kursleiter'. A blue button labeled 'zur Kursübersicht...' is visible. A red arrow points to a checkbox labeled 'Zugang zu allen Tutorials für Kursleiter freischalten', which is currently checked.

Der **Kursleiter** hat dort die Möglichkeit, **alle Tutorials** für den eigenen Account freizuschalten.

Über den Button „Zur Kursübersicht“ gelangt man zur **Übersicht** über die bereits **angelegten Kurse**. Zusätzlich können dort **neue Kurse** angelegt werden. Solange ein Kurs noch keine Teilnehmer hat, kann er dort auch wieder gelöscht werden.

ID	Kursbezeichnung	eingerichtet am	Teilnehmer
wok	Testkurs - Klasse 6a 17/18	2018-01-10	6
XYZ	Testkurs - Klasse 6b 17/18	2018-01-06	0
n5b	Testkurs - Klasse 6c 17/18	2018-01-02	28

Durch Klick auf die **Kurs-ID** (z.B. **XYZ**) kommt man zum **Übersichtsbildschirm** über die **Kursteilnehmer**. Ein neu angelegter Kurs hat noch keine Teilnehmer:

Kurs-ID: XYZ      Kursbezeichnung: Testkurs - Klasse 6b 17/18      Kursleiter: nils

← Zurück

### Kursteilnehmer

Benutzername	Lernfortschritt	letzter Zugriff
noch keine Teilnehmer...		

Um neue Kursteilnehmer anzulegen, kann man hier einen Benutzernamen pro Zeile eintragen! Beispiel:

- lena
- jonas
- tim
- mia
- bernd
- helena

+ Neue Benutzer anlegen!

In dieser Übersicht können **neue Kursteilnehmer** angelegt werden.

- Pro Zeile muss ein Benutzername eingegeben werden.
- Neu angelegte Teilnehmer bekommen automatisch erzeugte, zufällige Passwörter. Diese können durch die Teilnehmer später beliebig geändert werden.

- Beim Anlegen der Benutzer wird den Benutzernamen automatisch das Kurskürzel vorangestellt: Lena → XYZ-Lena
- Kursteilnehmer, die sich noch nie eingeloggt haben, können auch wieder gelöscht werden.

Bei Bedarf können hier die **Passwörter** der einzelnen Schüler **zurückgesetzt** werden!

ProgBob
bob3.org
Kursverwaltung nils

Kurs-ID: n5b      Kursbezeichnung: Testkurs - Klasse 6c 17/18      Kursleiter: nils

[← Zurück](#)   [Teilnehmerliste](#)

### Kursteilnehmer

Benutzername	Lernfortschritt	letzter Zugriff	
n5b-Amelie	21           1	heute	<a href="#">Passwort zurücksetzen</a>
n5b-Anna	6           0	vor 2 Tagen	<a href="#">Passwort zurücksetzen</a>
n5b-Ben	21           1	vor 2 Tagen	<a href="#">Passwort zurücksetzen</a>
n5b-Elias	9           2	heute	<a href="#">Passwort zurücksetzen</a>
n5b-Emilie	7           0	vor 2 Tagen	<a href="#">Passwort zurücksetzen</a>
n5b-Emma	6           0	vor 2 Tagen	<a href="#">Passwort zurücksetzen</a>
n5b-Felix	6           0	vor 2 Tagen	<a href="#">Passwort zurücksetzen</a>

Sobald die **Schüler** die Aufgaben bearbeiten, werden in der Übersicht die Anzahl der erworbenen Medaillen und die **abgeschlossenen Tutorials** angezeigt.

21
I
II
♥
🚗
III
✍️
👉
💧
📶
🐛
1

In diesem Beispiel hat Amelie insgesamt 21 Medaillen erarbeitet. Sie hat die Tutorials „Intro I“ und „Intro II“, die beiden Specials „Friend-Detection“ und „Police-Lights“, das „Intro III“ Tutorial und das „Color“ Tutorial fertig bearbeitet und ein eigenes Programm angelegt.

Durch Klick auf ein Tutorial-Symbol werden die einzelnen Lernstände der Schüler im jeweiligen Tutorial angezeigt:

Benutzername	Lernfortschritt - Intro II												
n5b-Amelie	if_else_1	if_else_2	if_else_3	ir_sensor_1	ir_sensor_2	temperatur	settled	for_1	for_2	for_3	for_4	for_5	final
n5b-Anna	if_else_1	if_else_2	if_else_3	ir_sensor_1	ir_sensor_2	temperatur	settled	for_1	for_2	for_3	for_4	for_5	final
n5b-Ben	if_else_1	if_else_2	if_else_3	ir_sensor_1	ir_sensor_2	temperatur	settled	for_1	for_2	for_3	for_4	for_5	final
n5b-Elias	if_else_1	if_else_2	if_else_3	ir_sensor_1	ir_sensor_2	temperatur	settled	for_1	for_2	for_3	for_4	for_5	final
n5b-Emilie	if_else_1	if_else_2	if_else_3	ir_sensor_1	ir_sensor_2	temperatur	settled	for_1	for_2	for_3	for_4	for_5	final
n5b-Emma	if_else_1	if_else_2	if_else_3	ir_sensor_1	ir_sensor_2	temperatur	settled	for_1	for_2	for_3	for_4	for_5	final
n5b-Felix	if_else_1	if_else_2	if_else_3	ir_sensor_1	ir_sensor_2	temperatur	settled	for_1	for_2	for_3	for_4	for_5	final

In diesem Beispiel hat Emma 3 Lektionen zum Lernziel „if-else“, 2 Lektionen zum „IR-Sensor“ und eine Lektion zum „Temperatursensor“ fertig bearbeitet. Die Lektionen zu den Lernzielen „for-Schleife“ und „setLed()-Funktion“ muss sie noch bearbeiten.

### Zugangsdaten der Teilnehmer:

Durch Klick auf den Button „Teilnehmerliste“ kommt man zum Übersichtsbildschirm über die **Zugangsdaten**:

ProgBob
Kursverwaltung nls

---

Kurs-ID: n5b      Kursbezeichnung: Testkurs - Klasse 6c 17/18      Kursleiter: nls

← Zurück
Drucken...

**Kursteilnehmer** ✕

Webseite: <a href="http://www.progbob.org">www.progbob.org</a> > Login	Name: n5b-Amelie	Passwort: abc123
Webseite: <a href="http://www.progbob.org">www.progbob.org</a> > Login	Name: n5b-Anna	Passwort: cde234
Webseite: <a href="http://www.progbob.org">www.progbob.org</a> > Login	Name: n5b-Ben	Passwort: def345
Webseite: <a href="http://www.progbob.org">www.progbob.org</a> > Login	Name: n5b-Elias	Passwort: efa456
Webseite: <a href="http://www.progbob.org">www.progbob.org</a> > Login	Name: n5b-Emilie	Passwort: fab567



Die Liste lässt sich ausdrucken und in Streifen schneiden. Diese können dann an die einzelnen Kursteilnehmer verteilt werden. Hier als Beispiel die Anmeldedaten für Anna:

Webseite: [www.progbob.org](http://www.progbob.org) > Login      Name: n5b-Anna      Passwort: **f813db**



## Klausur-Modul

- optional einsetzbar -

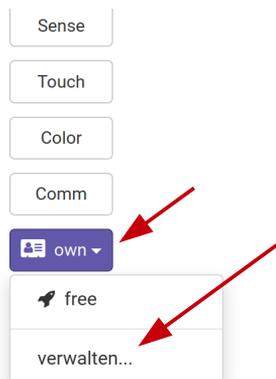
Um das Klausur-Modul nutzen zu können, muss der jeweilige Benutzer-Account wie auch bei der Kursverwaltung als '**Kursleiter**' freigeschaltet sein. Dazu reicht es, eine kurze E-Mail an [teacher@bob3.org](mailto:teacher@bob3.org) zu senden.

### Fallbeispiel:

Das folgende Fallbeispiel verdeutlicht die **Erstellung**, die **Durchführung** und die **Auswertung** einer Klausur mit zwei Aufgaben am Beispiel einer fiktiven Schulklasse ‚Klasse 7a‘ der Lehrerin Frau Theobald mit ihren neun Schülern:

- |           |          |         |
|-----------|----------|---------|
| * Anna    | * Lena   | * Bernd |
| * Luisa   | * Helena | * Mia   |
| * Hendrik | * Tim    | * Jonas |

Frau Theobald loggt sich mit ihrem **Kursleiter-Account** auf [www.ProgBob.org](http://www.ProgBob.org) ein. Ganz unten in der Liste der Tutorials auf der rechten Seite klickt sie den Button „own“ und „verwalten...“ an.



Frau Theobald gelangt zu folgender Ansicht:

 Eigene Tutorials ✕

Tutorial	Titel (lang)	Kapitel
 free	Free Tutorial	1



In dieser Eingabemaske können **eigene Tutorials** und **Klausuren** erstellt werden.

Die Lehrerin tippt als Name ‚Klausur‘ und als Titel ‚Klausur\_7a\_2020‘ ein und klickt auf ‚+ Add‘:

 Eigene Tutorials ✕

Tutorial	Titel (lang)	Kapitel
 free	Free Tutorial	4



Nun können der neu angelegten Klausur die gewünschten Aufgaben hinzugefügt werden. Dafür klickt Frau Theobald auf den neu entstandenen Button ‚Klausur‘:

The screenshot shows a window titled 'Eigene Tutorials' with a close button (X) in the top right. Below the title bar is a table with three columns: 'Tutorial', 'Titel (lang)', and 'Kapitel'. The first row contains a 'free' button, the text 'Free Tutorial', and the number '4'. The second row is highlighted in light blue and contains a 'Klausur' button, the text 'Klausur\_7a\_2020', and the number '1'. Below the table, there are three input fields: a dropdown menu with a document icon, a text field containing 'Klausur', a text field containing 'Klausur\_7a\_2020', and an 'OK' button with a checkmark icon. A red arrow points to the 'Klausur' button in the second row of the table.

Tutorial	Titel (lang)	Kapitel
free	Free Tutorial	4
Klausur	Klausur_7a_2020	1

Below the table, there are input fields: a dropdown menu with a document icon, a text field containing 'Klausur', a text field containing 'Klausur\_7a\_2020', and an 'OK' button with a checkmark icon.

At the bottom right of the window, there is a blue 'Schließen' button.

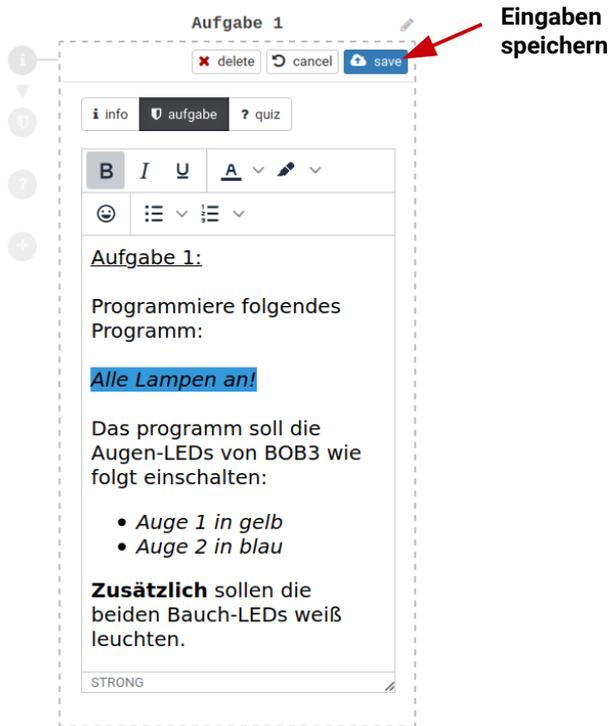
Jetzt befindet sich Frau Theobald im ersten Kapitel ihrer neu angelegten Klausur. Dort in der Eingabemaske klickt sie auf den Button „edit“:

The screenshot shows a window titled 'Neuer Sketch #10' with an 'edit' button in the top right corner. The main area contains the text 'info' and 'Dein neues Programm...'. A red arrow points to the 'edit' button.

Nun ändert sie den Titel in ‚Aufgabe 1‘ und wählt den Reiter ‚Aufgabe‘ aus, um die erste Aufgabe für die Klausur erstellen.

The screenshot shows a window titled 'Aufgabe 1' with a 'delete' button, a 'cancel' button, and a 'save' button in the top right corner. The main area contains the text 'eine neue Aufgabe ...'. A red arrow points to the 'aufgabe' tab in the top left corner.

Frau Theobald erstellt den Aufgabentext und speichert ihre Eingaben mit „save“:



Im Quellcodefenster sind bereits die Präprozessoranweisung „#include <BOB3.h>“ und die beiden Funktionen „setup()“ und „loop()“ eingetragen. Dies kann beliebig geändert und ergänzt werden.

Sobald man die gewünschten Änderungen im Quellcodefenster vorgenommen hat, klickt man auf „Compile“ um den Quellcode zu **kompilieren** und zu **speichern** und hat damit die Aufgabe 1 fertig erstellt.



Um ein **zweites Kapitel der Klausur** anzulegen, klickt Frau Theobald oben rechts auf das „**Plus**“:



Wie bei der ersten Aufgabe ändert sie wieder zuerst den Titel, diesmal in **„Aufgabe 2“**. Dann gibt sie den Aufgabentext für die Schüler ein. Anschließend nimmt sie dieses Mal auch **Änderungen im Quellcodefenster** vor und speichert diese mit dem Button **„Compile“**.

A screenshot of the ProgBob IDE interface. The top bar shows 'ProgBob bob3.org' on the left and 'Klausur' with navigation icons and a user profile 'bobini' on the right. The main area is divided into three sections: 1. A left sidebar with 'Medals' and various tool icons. 2. A central code editor showing C++ code for a Bob3 robot: 

```
1 #include <BOB3.h>
2
3 void setup() {
4
5 }
6
7
8 void loop() {
9
10 // Licht einschalten, wenn Arm 1 berührt wird:
11 if (bob3.getArm(1) != 0) {
12 // LED3 und LED4 weiß einschalten
13
14
15 // Auge 1 und Auge 2 in CYAN einschalten
16
17 }
18
19 // Licht ausschalten, wenn Arm 2 berührt wird:
20
21
22 }
```

 3. A right sidebar for 'Aufgabe 2' (Task 2) with an 'edit' button. The task title is 'aufgabe'. Below it, the text reads: 'Ergänze das vorgegebene Programm: Allen-Taschenlampe'. The description says: 'Bob soll als Taschenlampe mit grellblauem Licht programmiert werden: Sobald Arm 1 berührt wird, sollen die Bauch-LEDs in weiß und die Augen in CYAN eingeschaltet werden. Mit Berührung von Arm 2 sollen alle LEDs ausgeschaltet werden.' At the bottom, there is a 'Compile' button and a file name 'sketch-10.bob3'.

Frau Theobald geht in die **Kursverwaltung**, wählt ihren Kurs **„Klasse 7a“** aus und überprüft noch einmal die Teilnehmerliste. Den Kurs und die zugehörigen Schüler hat sie bereits wie im vorherigen Kapitel beschrieben angelegt.

## Klausur-Tag:

Um die Klausur zu starten, wählt Frau Theobald in der **Kursverwaltung** ihren Kurs ‚7a\_2020‘ aus und klickt dort auf den Button ‚Kurs-Tutorials / Prüfungen‘:

**ProgBob** bob3.org Kursverwaltung bobini

Kursleiter / Übersicht / Kurs 5mm

Teilnehmerliste **Kurs-Tutorials / Prüfungen**

Kurs-ID: 5mm Kursbezeichnung: 7a\_2020 Kursleiter: bobini

### Kursteilnehmer

Benutzername	Lernfortschritt	letzter Zugriff
5mm-Anna		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Bernd		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Helena		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Hendrik		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Jonas		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Lena		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Luisa		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Mia		noch nie <input type="button" value="Passwort zurücksetzen"/>
5mm-Tim		noch nie <input type="button" value="Passwort zurücksetzen"/>

Im folgenden Dialog-Fenster wählt sie ihre erstellte Klausur aus und klickt auf den Pfeil, um die Klausur den Schülern zur Bearbeitung zur Verfügung zu stellen:

### Tutorials - 7a\_2020

Tutorials für Teilnehmer:

verfügbare Tutorials:

- free - Free Tutorial
- Klausur - Klausur\_7a\_2020**

Tutorial:

Da es für Kursleiter auch möglich ist, eigene Programmier-Tutorials für die Schüler zu erstellen, werden im folgenden Dialog einige Optionen angeboten: ‚als normales Tutorial kopieren‘, ‚als normales Tutorial kopieren, zunächst versteckt‘ oder ‚als Prüfung‘ kopieren. Die Lehrerin wählt die Option ‚als Prüfung‘ aus und bestätigt die Auswahl mit ‚OK‘:

Wie soll das Tutorial kopiert werden? ✕

- als normales Tutorial
- als normales Tutorial, jedoch zunächst versteckt
- als Prüfung

✓ OK ✕ Abbruch

Die Klausur erscheint nun im linken Fenster-Teil:

Tutorials für Teilnehmer:

**Klausur - Klausur\_7a\_2020** 👁

verfügbare Tutorials:

**free - Free Tutorial**

**Klausur - Klausur\_7a\_2020**

Tutorial: 👁 verstecken ✕ entfernen

Um die Klausur zu starten, wird die Klausur ausgewählt und anschließend wird der Button ‚starten‘ gedrückt:

Tutorials für Teilnehmer:

**Klausur - Klausur\_7a\_2020** 👁

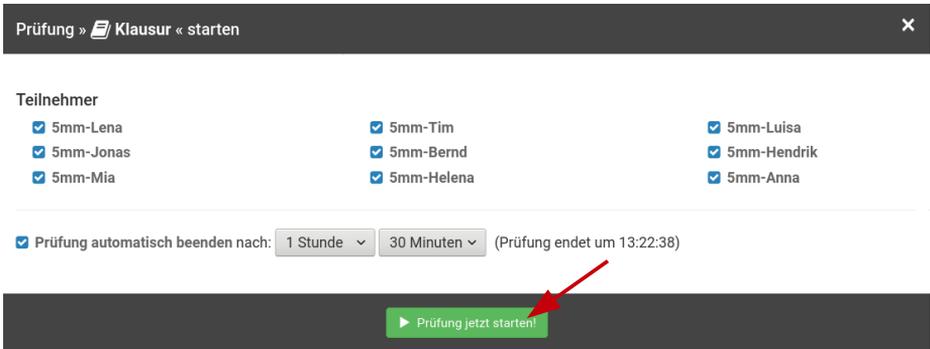
Prüfung: ▶ starten 🔗 umwandeln ✕ entfernen

verfügbare Tutorials:

**free - Free Tutorial**

**Klausur - Klausur\_7a\_2020**

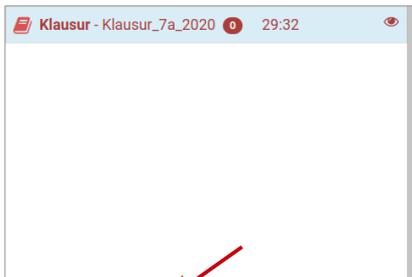
Im folgenden Dialog wählt Frau Theobald die Schülerinnen und Schüler aus, die an der Klausur teilnehmen sollen, legt eine Zeit fest nach der die Klausur automatisch enden soll und klickt auf den Button **„Prüfung jetzt starten!“**:



Jetzt haben ihre Schüler genau eine Stunde und dreißig Minuten Zeit, um die Aufgaben zu bearbeiten. Die Schüler loggen sich auf der Plattform ein und beginnen mit der Arbeit.

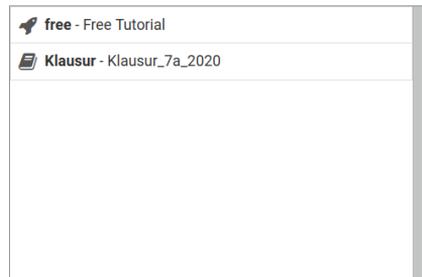
Die Klausur kann jederzeit mit dem Button **„beenden“** beendet werden:

Tutorials für Teilnehmer:



Prüfung:

verfügbare Tutorials:



Nachdem die Klausurzeit abgelaufen ist oder die Klausur von der Lehrerin manuell beendet wurde, werden automatisch alle zu diesem Zeitpunkt vorhandenen Schülerlösungen abgespeichert und die SuS können sich nicht mehr in die Klausurumgebung einloggen.

Zu Hause loggt sich Frau Theobald erneut ein und geht in der Kursverwaltung zu ihrem Kurs:

## Kursleiter

 zur Kursübersicht...

Zunächst drückt sie sich alle **Schülerlösungen** aus. Zu jedem Schüler wird eine **.pdf-Datei** bereitgestellt, die die jeweiligen Lösungen enthält:

### Kursteilnehmer



Benutzername	Lernfortschritt
 5mm-Anna	
 5mm-Bernd	
 5mm-Helena	
 5mm-Hendrik	
 5mm-Jonas	
 5mm-Lena	
 5mm-Luisa	
 5mm-Mia	
 5mm-Tim	

Die Lehrerin bekommt zusätzlich die Möglichkeit, den **Quellcode** der einzelnen Schülerlösungen zu verändern, zu kompilieren und konkret am BOB3 **auszuprobieren**. Dies erleichtert die Fehlersuche und hilft bei der Bewertung. Alle Änderungen können auf die Schülerlösung zurückgesetzt werden!

## Begleitendes OER-Arbeitsmaterial

Begleitend zu den digitalen Selbstlern-Einheiten auf [www.ProgBob.org](http://www.ProgBob.org) stehen auf [www.bob3.org](http://www.bob3.org) frei verfügbare OER-Lernmaterialien zur Verfügung:

# Modul 1

## Einführung & Grundlagen

Angefangen bei ersten Programmierschritten und dem Verstehen des Prinzips der Programmierung von Hardware (Compilieren, Übertragen) lernen die SuS in Modul 1 einzelne Aktionen zu programmieren, erzeugen einfache Sequenzen, erlernen das Prinzip der Verzögerung mit der delay()-Funktion, programmieren diverse LED-Blinklicht-Varianten, experimentieren mit verschiedenen Farbkonstanten und lernen Funktionen mit Parametern kennen.

### Materialien:

 [Lehrerbegleitheft Modul 1: AB 1 - AB 4](#)

 [Arbeitsblatt 1 - Roboter](#)

 [Lösungen zu AB 1](#)

 [Arbeitsblatt 2 - Benutzeraccount](#)

 [Lösungen zu AB 2](#)

 [Arbeitsblatt 3 – Klammern & Co.](#)

 [Lösungen zu AB 3](#)

 [Arbeitsblatt 4 – Programmieren](#)

 [Lösungen zu AB 4](#)

 [zu den Programmier-Einheiten ...](#)

Das Modul 1 umfasst insgesamt 6 Unterrichtseinheiten mit je ca. 45 Minuten, 11 Programmier-Einheiten und 4 Arbeitsblätter.

# Modul 2

## Sensoren & Kontrollstrukturen

Die Schüler lernen Verzweigungen anhand der Kontrollstruktur ‚if / else‘ kennen und arbeiten mit Vergleichsoperatoren und Wahrheitswerten. In kleinen Programm-beispielen wird die Verzweigung konkret zur Auswertung des IR-Sensors eingesetzt. In diesem Zusammenhang lernen die Schüler das Konzept von Variablen kennen und anwenden. Sie lernen die einzelnen Hardware-Bestandteile des Roboters und die jeweiligen Funktionen kennen. Anhand der Software-Bibliothek des BOB3 setzen sie sich mit Methoden, Funktionen und Parametern auseinander und erlernen als weitere Kontrollstruktur das Prinzip und die Anwendung von ‚for-Schleifen‘. Zum Abschluss der Lerneinheit erfolgt eine einfache Fehlersuche, bei der die Schüler lernen, Fehler im Programmcode zu entdecken und zu beheben.

### Materialien

-  [Lehrerbegleitheft Modul 2: AB 5 - AB 8](#)
-  [Arbeitsblatt 5 - if-else-Anweisung](#)
-  [Lösungen zu AB 5](#)
-  [Arbeitsblatt 6 – BOB's Hardware](#)
-  [Lösungen zu AB 6](#)
-  [Arbeitsblatt 7 – BOB's Software](#)
-  [Lösungen zu AB 7](#)
-  [Arbeitsblatt 8 – for-Schleife](#)
-  [Lösungen zu AB 8](#)
-  [zu den Programmier-Einheiten ...](#)

Das Modul 2 umfasst insgesamt 7 Unterrichtseinheiten mit je ca. 45 Minuten, 16 Programmier-Einheiten und 4 Arbeitsblätter.

# Modul 3

## Funktionen & Sensorik

Die Schüler vertiefen das Prinzip und die Anwendungsmöglichkeiten von Variablen und setzen das erlernte Wissen in kleinen Programmbeispielen um. In verschiedenen Experimenten mit den Multifeld-Touch-Sensoren verwenden sie Variablen, um die jeweils aktuellen Sensorwerte abzuspeichern und während des Programmablaufs zur Verfügung zu stellen. In diesem Zusammenhang beschäftigen sie sich intensiver mit der genauen Funktionalität der Sensoren und setzen sich mit dem Zeit-Multiplex-Verfahren auseinander. Als weitere Kontrollstrukturen lernen die Schüler ‚Switch-Case-Verzweigungen‘ und ‚While-Schleifen‘ kennen und vertiefen ihr Wissen zur Kontrollstruktur ‚if / else‘ anhand einer Leistungsüberprüfung. Die Schüler erlernen den Sinn und Zweck von Funktionen und erarbeiten den Unterschied von Funktionen mit und ohne Parameter. In verschiedenen Experimenten mit dem IR-Sensor werden Funktionen mit Rückgabewert konkret zur Auswertung eingesetzt. In diesem Zusammenhang erwerben die Schüler weiteres Wissen zur Funktionalität des IR-Sensors, insbesondere zum Reflexionsverfahren. Anschließend definieren die Schüler eigene Funktionen und diskutieren das aus diesen Möglichkeiten resultierende Potential. Zum Abschluss der Lerneinheit werden offene Aufgaben, wie z.B. die Programmierung eines Parksensors angeboten.

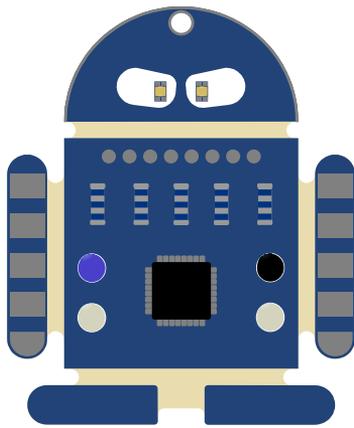
### Materialien:

-  [Lehrerbegleitheft Modul 3: AB 9 - AB 14](#)
-  [Arbeitsblatt 9 - Variablen](#)
-  [Lösungen zu AB 9](#)
-  [Arbeitsblatt 10 – Touch-Sensoren](#)
-  [Lösungen zu AB 10](#)
-  [Arbeitsblatt 11 – switch-case](#)
-  [Lösungen zu AB 11](#)
-  [Arbeitsblatt 12 – Funktionen](#)
-  [Lösungen zu AB 12](#)
-  [Arbeitsblatt 13 – while-Schleife](#)
-  [Lösungen zu AB 13](#)
-  [Arbeitsblatt 14 – IR-Sensor](#)
-  [Lösungen zu AB 14](#)
-  [zu den Programmier-Einheiten ...](#)

Modul 3 umfasst insgesamt 13 Unterrichtseinheiten mit je ca. 45 Minuten, 25 Programmier-Einheiten und 6 Arbeitsblätter.







**B-O-B-3**