Crashkurs für Lehrkräfte

Logo Digital.Point NEU.jpg

Der Digital.Point Crashkurs

Unsere **Digital.Point Crashkurse** gibt es für alle unsere Materialien. Sie wurden von uns eigenhändig erstellt und sollen allen Lehrkräften einen kleinen Einstieg in die Materialien ermöglichen. Dabei achten wir darauf, dass der Kurs in etwa **60 Minuten** umgesetzt werden kann. So lernen erfahrene Lehrkräfte die besonderen Funktionen kennen, während Anfänger sich mit den Grundfunktionen vertraut machen.

Sie schlüpfen dabei selbst in die Rolle der Schüler und können hoffentlich zu Hause oder am Schreibtisch ein wenig Spaß beim Knobeln haben oder eigene Programme entwickeln – vielleicht sogar bessere als die von uns vorgeschlagenen.

Wir wünschen Ihnen viel Spaß!

Ihr Digital.Point Team

DER BOB-3

Willkommen beim **Digital.Point Hildesheim**!

In diesem Crashkurs sollen Sie lernen wie der BOB 3 funktioniert, wie Sie ihn programmieren und wie Sie ihn im **Digital.Point** einsetzen können.
Der BOB 3 scheint zunächst einer der langweiligeren Roboter in unserer Auswahl zu comic-bob-bw-v4-mit-weiss.png sein, da er sich im Gegensatz zu den anderen nicht bewegen kann. Dennoch bietet der BOB 3 eine Vielzahl an Möglichkeiten ihn für Schüler*innen interessant und abwechslungsreich zu nutzen.

Durch seine verschiedenen Sensoren wie Touch, Licht und Abstandssensoren lassen sich spannende Experimente, kleine Spiele und interaktive Aufgaben gestalten. Außerdem kann er über seine LEDs Emotionen oder Zustände darstellen, was ihn besonders für kreative Programmieraufgaben attraktiv macht.
Mit einfachen Programmierbefehlen können Lernende schnelle Erfolgserlebnisse erzielen und gleichzeitig erste Programmlogiken wie Bedingungen, Schleifen oder Variablen kennenlernen.

Inhaltsverzeichnis:

1. Erste Schritte

1.1 Grundlagen zum BOB3

1.2 Programme auf den BOB3 übertragen

2. Programmieren des BOB3

2.1 Grafisches Programmieren mit BOB Blocks

2.2 Textbasiertes Programmieren mit Prog.Bob

3. Übung für Lehrkräfte

Erste Schritte

GRUNDLAGEN ZUM ROBOTER:

Der BOB3 ist ein kleiner, programmierbarer Roboter, der Kindern und Jugendlichen das Programmieren auf spielerische Weise näherbringt. Er kann erkennen, ob seine Arme berührt werden und sogar, an welcher Stelle oben, in der Mitte oder unten dies geschieht. Außerdem ist er in der Lage, andere BOB3-Roboter als Freunde zu erkennen, seine weißen Scheinwerfer einzuschalten, seine Augen in verschiedenen Farben blinken zu lassen und zwischen nahen und fernen Objekten zu unterscheiden. Der Roboter kann frei programmiert werden, eigene binäre Codes erhalten oder mit einer Knopfzelle und Lanyard als blinkendes Gadget um den Hals getragen werden.

Der BOB3 besteht aus mehreren wichtigen Bestandteilen:Screenshot BOB 3 aufbau .png

LEDs, Leuchtdioden
Der Roboter verfügt über zwei RGB-LEDs als Augen, die in allen Farben leuchten können, darunter rot, grün, blau, gelb, lila und viele weitere. Am Bauch befinden sich zwei superhelle weiße LEDs, die als Scheinwerfer dienen und den Roboter zum Beispiel als Taschenlampe nutzbar machen.

Multifeld-Touch-Sensoren
Beide Arme des BOB3 sind mit Touch-Sensoren ausgestattet, die erkennen, ob sie berührt werden. Dank der Multifeld-Technologie kann der Roboter sogar unterscheiden, an welcher Stelle der Arm berührt wird. Insgesamt verfügt er über sechs Tastsensoren, die angesteuert oder abgefragt werden können.

IR-Sensor
Der BOB3 besitzt einen Infrarot-Sensor, der aus einer violetten IR-Sende-LED und einem schwarzen IR-Empfänger besteht. Dieses System ermöglicht es dem Roboter, zwischen nahen und fernen Objekten zu unterscheiden, Hindernisse wie Hände oder Papier zu erkennen oder Nachrichten an andere BOB3-Roboter zu senden.

Mikrocontroller
Das Gehirn des BOB3 ist ein Mikrocontroller, der den Programmcode ausführt und die gesamte Elektronik steuert. Er bildet die zentrale Recheneinheit des Roboters und koordiniert alle Sensoren und Ausgabeelemente.

Im Digital.Point könen sie B-O-B-3 EduSets nutzen. In diesen finden sie jeweils 12 für Sie vorgebaute Roboter, durch den ProgBob-Programmier-Helm über ein USB-Kabel mit dem Computer mit Programmen bespielt werden können. Image (1).jpgImage.jpg

---

PROGRAMME AUF DEN BOB3 ÜBERTRAGEN

Wenn sie die BOB3 mit den in den Boxen beigefügtem USB zu Micro-USB verbunden haben müssen sie nun die BobDude bobdude-mac-inst-09.png Software herunterladen. Wenn sie nun Programme über die App am IPad oder den Browser erstellt haben können sie diese herunterladen und dann über das blaue Feld im Programm abspielen.

WINDOWS-INSTALLATION:

In der folgenden Anleitung wird die manuelle Installation der BobDude Software unter Windows (treiberlos, ab Windows 10) beschrieben:

1.1. Die folgenden Dateien müssen aus einer bestehenden Installation auf das Ziel-System kopiert werden:

```
C:\Program Files\BobDude\bin\bobdude\orca_service.exe
C:\Program Files\BobDude\bin\bobdude\bobdude.exe
C:\Program Files\BobDude\bin\bobdude\bobdude.conf
```

1.2. In der Shell als Admin Service einrichten:

```
SC.EXE CREATE "ORCA" binpath="C:\Program Files\BobDude\bin\bobdude\orca_service.exe"
SC.EXE START "ORCA"
```

Programmieren in BOB Blocks

Der BOB3 lässt sich sehr einfach über die BOB3 App oder die Website programmieren. Für die im Digital.Point vorliegenden Materialien wird dabei mit BOB-Blocks gearbeitet. Wenn sie mit Schülern der Grundschule mit dem BOB 3 arbeiten möchten kann Primary-Blocks eine etwas vereinfachte Vorstufe von BOB-Blocks genutzt werden. Mit ProbBoB ist neben den beiden Visuellen Programmier Interfaces auch ein Textuelles Interface für den BOB 3 verfügbar, welches durch eine Interaktive Lerneinheit einen eigenständigen Übergang ins Code-Schreiben in C++ ermöglicht.

> Öffnen sie nun die Website oder die App und machen sich mit der Umgebung vertraut.

Startseite BOB§ App.jpgScreenshot 2025-11-13 111424.png

Die BOB 3 App *Die Programmierumgebung von BOB Blocks erinnert Programmiersprachen wie Scratch*

UNSER ERSTES PROGRAMM:

Zunächst wollen wir erst einmal sprachlich beschreiben was unser Programm können soll:

> Wird das Programm gestartet leuchten die Augen des BOB3 in blau.

> Wird der Linke Arm des BOB3 berührt leuchten seine Augen in grün.

> Wird der rechte Arm des BOB3 berührt leuchten die Augen in rot.

> Wird der BOB3 nicht berührt wartet er 0,5 Sekunden und setzt dann die Augen wieder auf blau.

AUFGABE: PROGRAMMIEREN SIE EIN PROGRAMM DAS GENAU DIESE FUNKTIONEN ERFÜLLT.

> Der Linke Arm wird als Arm1 bezeichnet und der rechte Arm bezeichnet den Arm2.

LÖSUNG

Zunächst überlegen wir uns, was unter den **Mache am Anfang** und was unter den **Mache immer wieder Befehl** gestellt werden muss.

Mache am Anfang: Zu Beginn des Programms sollen die Augen auf blau gesetzt werden, darum nutzen wir das Puzzleteil: **setzte Augen auf:** und setzten es an den **Mache am Anfang** Befehl.

Programm Anfang .png

Mache immer wieder: Als nächstes wollen wir, dass es drei Zustände gibt:

1. Der linke Arm wird berührt
2. Der rechte Arm wird berührt
3. kein Arm wird berührt

Nun gibt es viele unterschiedliche Möglichkeiten, wie wir diese Zustände einführen können. Zur Darstellung sollen an dieser Stelle zwei Möglichkeiten vorgestellt werden.

Version 1

Zunächst unterscheiden wir zwischen Befehlen, die nur beim Start ausgeführt werden sollen – also dem **Setup** des Programms und solchen, die kontinuierlich wiederholt werden, also im **Loop**. Für unser Programm wollen wir im Setup zu Beginn die Augen auf die Grundfarbe Blau einstellen.	Programm Anfang .png
Zunächst wird im Loop die Grundstruktur des Programms erstellt. Dafür werden zwei Wenn–Sonst-Blöcke eingefügt: Einer soll später überprüfen, ob Arm1 etwas berührt, und der andere reagiert auf Berührungen von Arm2.	Falls sonst schleife .png

Im nächsten Schritt wird diese Struktur mit Leben gefüllt. Wenn Arm1 etwas berührt, wechseln die Augenfarben des Roboters zu Grün. Wird Arm1 hingegen nicht berührt, wartet der Roboter kurz und setzt die Augen danach wieder auf Blau. Anschließend wird dasselbe für Arm2 umgesetzt: Bei einer Berührung leuchten die Augen Rot, und wenn keine Berührung erkannt wird, wartet der Roboter erneut kurz und stellt die Augen auf Blau zurück. Dadurch reagiert der Roboter kontinuierlich auf Berührungen seiner beiden Arme und zeigt diese deutlich über wechselnde Augenfarben an.	Falls Schlaufe ganz.png

Version 2

Anstatt das Programm mit zwei If-Else-Schleifen aufzubauen, kann es auch mit drei If-Schleifen umgesetzt werden. Zunächst sollen die Armberührungen programmiert werden. Wenn der linke Arm berührt wird, leuchtet das Auge grün, wenn der rechte Arm berührt wird, leuchtet das Auge rot.	Version2 1 .png
Als letztes wird nun noch die Bedingung eingeführt, dass nichts berührt wird, dann schaltet es die Augen wieder auf blau und wartet auf einen neuen Input.	Version 22.png

Textuelles Programmieren in Prog.Bob

Wenn nicht nur Visuell sondern auch textuell Programmiert werden soll kann der BOB3 über die Website: www.ProbBob.org auch mit Code programmiert werden. Wir wollen nun genau die selben Funktionen mit Code Programmieren. screenshot-2025-11-20-145736.png

Um auf Prog.Bob programmieren zu können müssen sie zunächst das Intro I durchführen bevor sie ins freie Programmieren starten dürfen. Dies sollte ungefähr 20-30 Minuten dauern.

ProbBob Accounts können auch ohne E-Mail Adresse erstellt werden, erstellen sie für Ihre Klasse Accounts und Passwörter sodass der Fortschritt nicht verloren geht.

Mache einmal am Anfang wird im textuellen codieren zu **setup.**
Mache immer wieder wird im Code als **loop** bezeichnet.

AUFGABE: PROGRAMMIEREN SIE WIEDER EIN PROGRAMM DAS DIE SELBEN ANFORDERUNGEN ERFÜLLT.

TIPPS

Augen eine Farbe geben: **bob3.setEyes(LINKS, RECHTS)**

Berührungssensor am Arm: **bob3.getArm(ARM (1,2))**

LÖSUNG

```
#include <BOB3.h>

void setup() {
  bob3.setEyes(BLUE, BLUE);
}

void loop() {
  int links = bob3.getArm(1);
  int rechts = bob3.getArm(2);

  if (links != 0) {
    bob3.setEyes(GREEN, GREEN);
  } else if (rechts != 0) {
    bob3.setEyes(RED, RED);
  } else {
    bob3.setEyes(BLUE, BLUE); // Standardfarbe
  }
}
```

Übungsaufgabe für Lehrkräfte

Zuletzt soll der BOB3 einfache Rollen übernehmen, mit denen Sie ihren Schülern das nutzen des BOB3 zeigen können.

BOB3 als Leselicht:

> Aufgabe: Die Augen und die LEDs des BOB`s sollen als Leselampe dienen.
> Der linke arm ist der "AN" Schalter der rechte Arm der "AUS" Schalter.

TIPPS

- Ein Auge auf Schwarz zu setzten schaltet es ab.
- der test ob der Arm berührt wird wird mit bob3.getArm(1,2) ausgeführt

LÖSUNG BLOCKS

LEselampe Programm .png

LÖSUNG CODE

```
#include <BOB3.h>

void setup() {
  // wird einmal beim Start ausgeführt
  bob3.enableArms(true);   // Arme aktivieren, damit Berührungen erkannt werden
}

void loop() {
  // Arm 1 berührt?
  if (bob3.getArm(1) > 0) {
    bob3.setEyes(rgb(255,255,255), rgb(255,255,255));  // Augen weiß
    bob3.setWhiteLeds(1, 1);                  // Bauch-LEDs an
  }

  // Arm 2 berührt?
  if (bob3.getArm(2) > 0) {
    bob3.setWhiteLeds(0, 0);                  // Bauch-LEDs aus
    bob3.setEyes(rgb(0,0,0), rgb(0,0,0));          // Augen schwarz
  }
}
```

BOB3 als dimmbare Taschenlampe

Aufgabe: Als nächstes soll die Lampe nicht nur ein und ausgeschaltet werden können sondern auch die Intensität des Lichts angepasst werden. Dafür soll die Anzahl der leuchtenden Lampen 0-4 eingestellt werden können.

TIPPS

Tipps TaschenLampe .png

LÖSUNG BLOCKS

Taschenlampe Programm .png

LÖSUNG CODE

```
#include <BOB3.h>

void setup() {

}

void loop() {
  int Links = bob3.getArm(1);
  int Rechts = bob3.getArm(2);

  // Links steuert LEDs
  if (Links == 1) {
    bob3.setEyes(WHITE, WHITE);
    bob3.setWhiteLeds(ON, ON);
  } else if (Links == 2) {
    bob3.setEyes(WHITE, WHITE);
    bob3.setWhiteLeds(ON, OFF);
  } else if (Links == 3) {
    bob3.setEyes(WHITE, WHITE);
    bob3.setWhiteLeds(OFF, OFF);
  }

  // Rechts überschreibt evtl. Links
  if (Rechts != 0) {
    bob3.setEyes(OFF, OFF);
    bob3.setWhiteLeds(OFF, OFF);
  }

  delay(50); // kleine Pause, um zu flackern zu vermeiden
}
```

BOB3 als Alarmanlage

> In dieser letzten Aufgabe wollen wir den BOB3 zur Alarmanlage programmieren.

Aufgabenbeschreibung:
Programmiere den BOB3-Roboter so, dass er als kleine Alarmanlage funktioniert:

- **Auslösen:** Sobald der IR-Sensor eine Annäherung erkennt (Wert > 8), startet ein blinkendes LED-Muster über die Augen- und weißen LEDs.
- **Signal:** Das Muster soll auffällig und rhythmisch blinken, um eine Alarmreaktion zu simulieren.
- **Deaktivieren:** Das Blinkmuster endet erst, wenn beide Arme des Roboters gleichzeitig bewegt werden (Profi-Reset).
- **Pause:** Nach dem Stoppen des Signals wartet der Roboter kurz, bevor er wieder auf neue Annäherungen reagiert.

TIPPS

- LEDs steuern mit `bob3.setEyes(FarbeLinks, FarbeRechts)` und `bob3.setWhiteLeds(links, rechts)`.
- Sensoren abfragen mit `bob3.getIRSensor()` und `bob3.getArm(1/2)`, um Auslösen und Stoppen zu kontrollieren.

LÖSUNG BLOCKS

Programm Alamarm .png

LÖSUNG CODE

```
#include <BOB3.h>

void loop() {

 if (bob3.getIRSensor() > 8) {

  while (true) {
   bob3.setEyes(ORANGE, OFF);
   bob3.setWhiteLeds(ON, ON);
   delay(50);

   bob3.setEyes(OFF, OFF);
   bob3.setWhiteLeds(OFF, OFF);
   delay(50);

   bob3.setEyes(OFF, ORANGE);
   bob3.setWhiteLeds(ON, ON);
   delay(50);

   bob3.setEyes(OFF, OFF);
```

```
      bob3.setWhiteLeds(OFF, OFF);
      delay(50);

      // Profi-Reset-Funktion über Armbewegung
      int Links = bob3.getArm(1);   // Arm 1 abfragen
      int Rechts = bob3.getArm(2);  // Arm 2 abfragen

      if (Links != 0 && Rechts != 0) {  // beide Arme aktiv
        break;  // Schleife verlassen
      }

    }

    delay(2000);
  }

}
```

WIR FREUEN UNS, DASS SIE UNSEREN CRASHKURS GENUTZT HABEN. NUTZEN SIE NUN IHR WISSEN, UM SPANNENDE UND KREATIVE STUNDEN MIT IHREN SCHÜLERN IM DIGITAL.POINT ZU GESTALTEN. VIEL ERFOLG UND VIEL FREUDE BEIM AUSPROBIEREN!

Ihr Digital.Point Team