2.2 B0B 3

Herzlich Willkommen im Digital.Point.
Der BOB3 ist ein kleiner programmierbarer Lernroboter, mit dem Schülerinnen und Schüler spielerisch Programmieren und technisches Denken lernen können. Durch seine LEDs und Sensoren bietet er anschauliche Möglichkeiten, eigene Programme zu testen und zu verstehen.
Es stehen davon 3x12er Sets im Digital.Point bereit bereit.

2.2.1 Crashkurs

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:

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

2.2.2 Unterrichtseinheiten

Lernstationsarbeit bis Klasse 6

Logo Digital.Point NEU.jpg

> Dies ist die Lernstationsarbeit des Herstellers. Informationen und Materialien sind in dieser Form übernommen.

Die Lernkarten sind zur Durchführung eines offenen Unterrichts in Form des Stationenlernens konzipiert. Dabei bearbeiten die Schüler individuell oder in kleinen Gruppen die einzelnen Stationen. Die jeweilige Station ist auf den Lernkarten oben links als Buchstabe aufgeführt, innerhalb einer Station sind die Karten durchnummeriert. Zur Differenzierung sind die Karten oben rechts mit einem bis drei Sternen markiert. Anspruchsvolle Karten haben beispielsweise drei Sterne. Zusätzlich gibt es bei einigen Karten einzelne Aufgaben mit mehreren Sternen. Diese sind ebenfalls zur Differenzierung gedacht. Jede Lernkarte hat zusätzlich zur besseren Übersicht eine Titelzeile als Überschrift, in der das Lernziel der Karte grob beschrieben wird. Zu jeder Station gibt es zusätzlich einen Werkstattplan in dem die SuS ihren Lernfortschritt dokumentieren.

Lernkarten_GS_BOB3_A.png

DIE EINFÜHRUNG (LERNSTATION A)

Die Bearbeitung der Lernstation A ist die Voraussetzung für alle anderen Lernstationen und sollte somit von allen Schülerinnen und Schülern als Erstes bearbeitet werden! In dieser Station lernen die SuS zunächst die grundlegende Bedienung der Open Roberta Oberfläche und die Übertragung der Programme auf den BOB3. Später lernen sie erste Befehlsblöcke zur Veränderung der Farbe der Augen kennen und verwenden diese.

Schwierigkeitsgrad: Einfach
Zeitbedarf: ca. 20–40 Minuten

Material:

Stationskartensatz A
Werkstattplan Station A

POLIZEIBLINKLICHT (LERNSTATION B)

Lernkarten_GS_BOB3_B.png

An dieser Lernstation programmieren die SuS zunächst ein einfaches Blinkprogramm mit dem BOB3: Das linke Auge wird rot eingeschaltet, es wird kurz gewartet und wieder ausgeschaltet. Die SuS erweitern das Programm dann so, dass beide Augen abwechselnd blinken und können dabei mit verschiedenen Farben und Geschwindigkeiten experimentieren. Die SuS lernen, wie man mit Befehlsblöcken zur Änderung der Farbe und mit Befehlsblöcken zur kurzen Verzögerung Blinkmuster erzeugen kann, indem sie diese zu Sequenzen verbinden. Zur Differenzierung können die SuS zusätzlich ihr Programm zu einem Polizeiblinklicht mit einer komplexeren Sequenz erweitern.
Schwierigkeitsgrad: Einfach bis Mittel
Zeitbedarf: ca. 15–30 Minuten

Material:

Stationskartensatz B

Werkstattplan Station B

TASCHENLAMPE (LERNSTATION C) lernkarten-gs-bob3-c.png

An dieser Lernstation soll der Bob als Taschenlampe programmiert werden. Bei Berührung des linken Arms sollen die Augen-LEDs und später auch die Körper-LEDs weiß eingeschaltet werden. Mit Berührung des rechen Arms werden dann alle LEDs wieder ausgeschaltet. Dazu lernen die SuS das Konzept von Verzweigungen und Bedingungen kennen. Sie verwenden einen „Wenn-Mache“-Block und kombinieren diesen mit einem Block für die Eingabewerte der Touch-Sensoren, um auf Benutzereingaben reagieren zu können. Zur Differenzierung muss die Lösung bei der Stationkarte 5 von den SuS selbstständig aus den Hinweisen erarbeitet werden.

Schwierigkeitsgrad: Einfach bis Mittel
Zeitbedarf: ca. 15–30 Minuten

Material:

Stationskartensatz C

Werkstattplan Station C

REGENBOGEN (LERNSTATION D) Lernkarten_GS_BOB3_D.png

An dieser Lernstation experimentieren die SuS mit den sechs Touch-Sensoren von BOB3: Je nachdem welcher Touch-Sensor ausgelöst wird, sollen die Augen-LEDs in einer bestimmten Farbe leuchten. Wird danach ein anderer Touch-Sensor ausgelöst, soll die Farbe sich verändern. Die SuS können dazu ihre sechs Lieblingsfarben verwenden. Sie erwerben vertiefte Kenntnisse in der Benutzung des „Wenn- Mache“-Blocks. Sie lernen, wie dieser Block zum „Wenn-Mache-Sonst-Wenn-Mache“ Block erweitert werden kann, und wie man dadurch auf verschiedene Fälle unterschiedlich reagieren kann.

Schwierigkeitsgrad: Mittel
Zeitbedarf: ca. 15–30 Minuten

Material:

Stationskartensatz D

Werkstattplan Station D

ALARMANLAGE (LERNSTATION E) Lernkarten_GS_BOB3_E.png

An dieser Station wird der Bob zunächst als Alarmsensor und dann als Alarmanlage programmiert. Die SuS experimentieren mit dem IR-Sensor und lernen, dass der Bob auch berührungslos Objekte, wie zum Beispiel eine Hand, wahrnehmen kann. Diese Station ist speziell für leistungsstarke SuS konzipiert. Die SuS erwerben

Kenntnisse in der Benutzung des „Wenn-Mache-Sonst“-Blocks. Sie lernen Sensorwerte kennen und vergleichen diese mit einem Referenzwert. Später lernen sie auch Endlosschleifen und deren Abbruch kennen, um den Alarm-Zustand anzuzeigen und zu beenden.

Schwierigkeitsgrad: Mittel bis Hoch
Zeitbedarf: ca. 30–60 Minuten

Material:

Stationskartensatz E
Werkstattplan Station E

ZUFALL (LERNSTATION F) Lernkarten_GS_BOB3_F.png

An dieser Station programmieren die SuS Bob als Zufallsgenerator für Ja/Nein Entscheidungen (Zufallsexperiment Münzwurf). Nach Berührung des rechten Arms blinken die LEDs zunächst in Gelb. Je nach Ergebnis des Zufallsexperiments leuchten die LEDs anschließend in Rot oder in Grün auf. Die SuS lernen dabei eine Methode zur wiederholten Ausführung von Programmblöcken kennen, und erwerben erste Kenntnisse über die Funktion zur Generierung von Zufallszahlen. Zum Abschluss erstellen die SuS eine Statistik über die Zufallsergebnisse.

Schwierigkeitsgrad: Mittel
Zeitbedarf: ca. 30–60 Minuten

Material:

Stationskartensatz F
Werkstattplan Station F

WÜRFEL (LERNSTATION G) Lernkarten_GS_BOB3_G.png

An dieser Station programmieren die SuS zunächst ein einfaches Würfelprogramm. Sie weisen einer Variablen den Zufallswert zu und verwenden den Wiederhole-N-mal-Block, um mit den LEDs das Ergebnis des Wurfs anzuzeigen. Im zweiten Teil wird das Programm so abgeändert, das ein Wurfergebnis zwischen Eins und Vier simultan mit den LEDs angezeigt wird. Zur Fallunterscheidung verwenden die SuS einen Wenn-mache-sonst-wenn-mache-Block. Zum Abschluss experimentieren die Schüler mit dem Zufallsgenerator und erstellen eine Statistik über die Ergebnisse.

Schwierigkeitsgrad: Hoch
Zeitbedarf: ca. 45–90 Minuten

Material:

Stationskartensatz G
Werkstattplan Station G

FOTO-BOB (LERNSTATION H) Lernkarten_GS_BOB3_H.png

An dieser Station wird Bob so programmiert, dass er den Selbstauslöser und das Blitzlicht von einem Fotoapparat simuliert. Am Anfang zeigt Bob die Berührung der Arme an. Bei Auslösung des Sensors wird das Auge auf der jeweiligen Seite gelb eingeschaltet. Anschließend verwenden die SuS den UND-Operator um bei gleichzeitiger Berührung beider Arme den Selbstauslöser zu starten, der Selbstauslöser wird dabei durch eine eigene Funktion realisiert. In dieser Funktion wird ein Wiederhole-N-mal-Block verwendet, um die Augen in der Wartezeit schnell Blinken zu lassen. Anschließend programmieren die SuS eine weitere Funktion um das Blitzlicht zu simulieren: Dazu werden alle LEDs kurz, weiß eingeschaltet und anschließend wieder ausgeschaltet.

Schwierigkeitsgrad: Hoch
Zeitbedarf: ca. 45–90 Minuten

Material:

Stationskartensatz H
Werkstattplan Station H

Offene Stationen

AMPEL (LERNSTATION I) Lernkarten_GS_BOB3_I.png

An dieser Station überlegen sich die Schüler, wie BOB3 als Ampel eingesetzt werden kann. Dazu diskutieren sie was eine Ampel ist, analysieren was diese genau macht und überlegen in welchen Farben sie in welcher Situation leuchtet. Danach konzipieren sie einen eigenen Algorithmus je nach individuellen Fähigkeiten, welchen sie anschließend in ein reales Programm umsetzen und mit BOB3 in der Praxis testen. Die Lehrkräfte bekommen exemplarisch drei differenzierte Lösungsvorschläge (Lehrerkarten: manuelle Ampel, automatische Ampel, Ampel mit Auslöser) mit unterschiedlichem Komplexitätsgrad.

Schwierigkeitsgrad: Hoch
Zeitbedarf: ca. 45–90 Minuten

Material:

Stationskartensatz I
Werkstattplan Station I

BAUSTELLENLICHT (LERNSTATION J) Lernkarten_GS_BOB3_J.png

Die SuS prorammieren Bob als Baustellenwarnlicht. Am Anfang zeigt Bob die Berührung der Arme an. Bei Auslösung des Sensors wird das Auge auf der jeweiligen Seite gelb eingeschaltet. Anschließend verwenden die SuS den UND-Operator um bei gleichzeitiger Berührung beider Arme den Selbstauslöser zu starten, der Selbstauslöser wird dabei durch eine eigene Funktion realisiert. In dieser Funktion wird ein Wiederhole-N-mal-Block verwendet, um die Augen in der Wartezeit schnell Blinken zu lassen. Anschließend programmieren die SuS eine weitere Funktion um das Blitzlicht zu simulieren: Dazu werden alle LEDs kurz, weiß eingeschaltet und anschließend wieder ausgeschaltet.

Schwierigkeitsgrad: Hoch
Zeitbedarf: ca. 45–90 Minuten

Material:

Stationskartensatz J
Werkstattplan Station J

Klassen 9-10 Textbasiertes Programmieren.

Logo Digital.Point NEU.jpg

1Rdimage.png

Für Schülerinnen, die statt mit Blocks textbasiert programmieren wollen und können, steht für den B-O-B-3 das ProgBob-Interface bereit. In diesem können Schülerinnen ihre Kenntnisse in einer an C bzw. C++ orientierten Programmiersprache erweitern und dabei die Funktionen des B-O-B-3 kennenlernen. Dabei kann es hilfreich sein, den Roboter zuvor mit Blocks kennengelernt zu haben, ist aber nicht notwendig. Vom Hersteller steht eine auf sechs mal 45 Minuten ausgelegte Unterrichtseinheit mit einer Einführung ins Programmieren zur Verfügung, die Sie auch in der Digital.Library finden können. In dieser auf das Wesentliche verkürzten Version dieses Kurses lernen die SuS zunächst den Bot und seine Bestandteile kennen, erstellen sich dann einen ProgBob-Account und arbeiten anschließend selbstständig mit Laptop und B-O-B-3 in der interaktiven Lerneinheit.

Benötigtes Material:

- -Genügend B-O-B-3 Roboter
- -Laptops mit der BobDude Programm zum bespielen der B-O-B-3

DER B-O-B-3 ROBOTER

Arbeitsblatt_GS_BOB3_3_Bob.png

PROGRAMMIEREN MIT PROGBOB

Um programmieren zu können, braucht man immer eine Programmiersprache und eine Umgebung. Für den B-O-B-3 ist die Sprache eine eigene Version der Programmiesprache C bzw C++, die Umgebung wird ProgBob genannt. Ihr findet sie unter folgendem Link: https://www.progbob.org/Als Erstes müsst ihr euch einen Account erstellen. Diesen benötigt ihr, damit ihr eure Fortschritte speichert und beim nächsten Mal genau dort weitermachen könnt, wo ihr aufgehört habt. Um die Zugangsdaten nicht zu vergessen, schreibt sie hier auf:

User: _____

Code: _____

Nun startet ihr direkt in das Intro und macht euch damit vertraut. Wenn ihr an einem Punkt seid, an dem ein Programm erstellt (compiliert) werden soll, drückt den **Compile**-Knopf. Anschließend erhaltet ihr eine B-O-B-3-Datei. Ladet diese über **Bob-Dude** auf euren B-O-B-3 und probiert das Programm aus.

Das war’s schon! Viel Spaß mit der interaktiven Lerneinheit. punkte-oben.png

2.2.3 Materialisten und Handouts

Materialiste BOB 3

Logo Digital.Point NEU.jpg

IN JEDER UNSERER B-O-B-3 KISTEN SIND :

- **12 BOB 3 ROBOTER**
- **12 PROGBOB USB-ADAPTER MIT USB KABEL**
- **1 LEHRHANDREICHUNG**
- **MODULBESCHREIBUNGEN 1, 2 UND 3**
- **KOPIERVORLAGEN MIT LÖSUNGEN**

4f0234fa-1da9-4d0a-bcf9-0c04b268a4f5.jpg

2.2.4 Öffentliche Dokumente und Links

Für den BOB3 Roboter stehen von Hersteller einige Unterrichtsmaterialien zur Verfügung

Legende für Arbeitsblätter:

B = Basisinformationen

W= Wissen

E= Experiment

Das Arbeitsblatt: Klassen 5-6 E1 Taschenlampe ist also ein Experiment für die Klassenstufen 5-6 und es wird dabei der BoB als Taschenlampe genutzt.

Klasse 5-6 B1 - Los geht's!

logo-digital-point-neu.jpg

B1 - Los geht's!

Der Start

In dieser Unterrichtseinheit lernen die Schülerinnen und Schüler die grundlegende Funktion der Programmieroberfläche und des Roboters kennnen. Sie erstellen ein erstes, einfaches Programm, compilieren es und übertragen das Ergebnis auf den Roboter.
"Eine ‚Anweisung' ist eine Aufforderung an den Mikrocontroller, etwas auszuführen."
Die SuS lernen den «Mache einmal am Anfang» und den «setze Augenfarben»-Block kennen und kombinieren die beiden Blöcke zu einem ersten Programm. Dieses Programm übertragen sie auf den Roboter und testen die Funktionsweise aus. Anschliessend verändern sie die Augenfarbe und probieren das Programm wieder praktisch aus. Zum Schluss lernen sie noch den «schalte Bauchleds»-Block kennen und probieren diesen wieder am Roboter aus.
Thema: Erste Schritte
Bereich: Basics/Grundlagen
Lernziele: Überblick über die Programmierumgebung, Compilieren und Übertragen, erste Programmierschritte, Augen-Leds ansteuern, Farben ändern, Anweisungen, Bauch-Leds ansteuern, Parameter ändern
Anspruch: Einfach
Aufgaben: A1-A5
Zeitbedarf : ca. 20 Minuten

bob basics 1.jpg
bobs basics 2.jpg bobs basics 3.jpg bobs basics 5.jpg bobs basics 6.jpg bobs basics 7.jpg

Download als PDF

Klasse 5-6 B2 - Erste Programme

logo-digital-point-neu.jpg

B2 - Erste Programme

Sequenzen

In dieser Unterrichtseinheit lernen die Schülerinnen und Schüler, wie man durch Kombination von Blöcken zur Verzögerung und Blöcken zum Schalten der Leds Blinksequenzen erzeugen kann.
Eine ‚Sequenz' ist eine Abfolge von einzelnen Anweisungen, die nacheinander ausgeführt werden.
Die SuS lernen zunächst den «warte x Millisekunden»-Block kennen und kombinieren dienen mit den bereits bekannten Blöcken zum Ein- und Ausschalten der Leds. Da diese Sequenz zum «Mache einmal am Anfang»-Block gehört, wird die Sequenz nur einmal, direkt nach der Programmierung bzw. nach dem Einschalten ausgeführt. Im nächsten Schritt wird der «Mache einmal am Anfang»-Block durch den «Mache immer wieder»-Block ausgetauscht, dadurch ergibt sich ein dauerhaftes Blinken. Danach wird der Zeitparameter verändert und die Schüler beobachten die Auswirkungen der Veränderungen. Anschliessend wird ein abwechselndes Blinken der Augen implementiert. Zur Differenzierung können die SuS das abwechselnde Blinken der Augen auf die Bauch-Leds erweitern und zu einem 'Überkreuz-Blinken' variieren.
Thema: Sequenzen
Bereich: Basics/Grundlagen
Lernziele: Erste eigene Programme, Sequenzen, Prinzip der Verzögerung, «Mache immer wieder»-Block, Blinklichter erzeugen, Varianten entwickeln
Anspruch: Einfach
Aufgaben: A1-A10
Differenzierung: A11+A12
Zeitbedarf : ca. 30 Minuten
Erte Programme 1 .jpgErste Programme 2 .jpgErste Programme 3 .jpgErste Programme 4.jpg
Erste Programme 5.jpgErste Programme 6.jpg

Download als PDF

Klasse 5-6 E1 - Taschenlampe

logo-digital-point-neu.jpg

E1 - Taschenlampe

Mehrfachverzweigung und Touch-Sensoren

In dieser Unterrichtseinheit programmieren die Schülerinnen und Schüler eine Taschenlampe mit einstellbarer Helligkeit. An Arm1 des Roboters kann die Helligkeit eingestellt werden, mit Arm 2 lässt sich die Taschenlampe wieder ausschalten.
"Beide Arme vom BOB3 sind Touch-Sensoren. Die Arme „merken“ also, ob sie berührt werden, oder nicht!"
Die SuS bekommen zunächst eine kleine Einführung zum Messprinzip der Touch-Sensoren. Anschließend ändern sie den Parameter des «wird Arm1 irgendwo berührt?»-Blocks in 'oben' und in 'unten' und testen die neuen Programmvarianten. Dann erweitern sie das Programm um einen Ausschalter an Arm 2, und setzen zusätzlich die Bauch-Leds für mehr Licht ein. Im Zweiten Teil starten die SuS mit einem neuen Taschenlampen Programm bei dem sie die Mehrfachverzweigung einsetzen. Dazu erweitern sie den regulären «falls dann»-Block um mehrere «sonst falls dann»-Zweige. In den verschiedenen Zweigen werden dann unterschiedliche Helligkeitsstufen implementiert. Im letzten Zweig werden nach Berührung von Arm 2 alle Leds ausgeschaltet.
Thema: Mehrfachverzweigung
Bereich: Wissen
Lernziele: Bedeutung und Anwendung von Verzweigungen, Vergleichsoperatoren, Wahrheitswerte, «falls-dann»-Block
Anspruch: Einfach
Aufgaben: A1-A12
Zeitbedarf : ca. 30 Minuten

Download-PDF

E11.jpgE12.jpgE13.jpgE14.jpgE15.jpgE16.jpgE17.jpgE18.jpgE19.jpg

Download-PDF

Klasse 5-6 E2 - Vogelwarnsystem

logo-digital-point-neu.jpg

E2 - Vogelwarnsystem

Hinderniserkennung für ein Flugzeug

In dieser Unterrichtseinheit programmieren die Schülerinnen und Schüler eine Hinderniserkennung, dabei lernen sie den Infrarot-Reflexions-Sensor und die Verwendung von Variablen kennen.
"Eine Variable ist ein Speicherort für Zahlen, Zeichen oder sonstige Daten."
Implementation einer Hinderniserkennung für Flugzeuge: Bob analysiert mit seinem IR-Sensor den Reflexionswert. Der aktuelle Wert in einer Variablen gespeichert, bei Überschreitung eines Grenzwertes aktiviert er ein Warnblitzlicht.
Thema: Sensoren
Bereich: Experimente
Vorraussetzung: Station B2
Lernziele: Implementation einer Hinderniserkennung: Bob analysiert mit seinem IR-Sensor den Reflexionswert. Bei Überschreitung eines Grenzwertes aktiviert er ein Warnblitzlicht. Fallunterscheidung, Variable und Operator.
Anspruch: Einfach
Aufgaben: A1-A8
Differenzierung: A9
Zeitbedarf : ca. 40 Minuten
E21.jpgE22.jpgE23.jpgE24.jpgE25.jpgE26.jpgE27.jpgBobBlockly_Sicherungsblatt_E2-V1_2-web (1).jpg

Downloads:

Lernkarten-PDF Sicherungsblatt-PDF

Klasse 5-6 E3 - Bob's smart-home

logo-digital-point-neu.jpg

E3 - Bob's smart-home

Automatische Indoor-Beleuchtung

In dieser Unterrichtseinheit programmieren die Schülerinnen und Schüler eine automatische Indoor-Beleuchtung, dabei lernen sie den Infrarot-Umgebungslicht-Sensor kennen.
Die Abkürzung „IR“ steht für „Infrarot“. Infrarotlicht ist eine spezielle Lichtart.
Die SuS lernen den Infrarot-Umgebungslicht-Sensor kennen. Mit einer Fallunterscheideung in Kombination mit dem «hole IR-Helligkeits-Wert»-Block kann der Roboter je nach Umgebungslicht unterschiedlich reagieren: Bei Helligkeit sollen alle Leds ausgeschaltet werden, bei Dunkelheit sollen alle Leds weiß eingeschaltet werden. Die SuS sollen das Programm in der Praxis ausprobieren indem sie den Roboter an dunklen und hellen Stellen platzieren.
Thema: Sensoren
Bereich: Experimente
Lernziele: Implementation einer automatischen Indoor Beleuchtung: Bob analysiert mit seinem IR-Sensor den Helligkeitswert der aktuellen Tageslichtsituation und schaltet ab einem Schwellwert automatisch alle Leds ein.
Anspruch: Einfach
Aufgaben: A1-A3
Differenzierung: A4
Zeitbedarf : ca. 20 Minuten
E31.jpgE32.jpgE33.jpgE34.jpgE35.jpg

Download-PDF

Klasse 5-6 E4 - Alice und Bob

logo-digital-point-neu.jpg

E4 - Alice und Bob

Kommunikation

In dieser Unterrichtseinheit programmieren die Schülerinnen und Schüler eine Datenübertragung zwischen den beiden Robotern 'Alice' und 'Bob'. Alice sendet bei Berührung ihrer Arme eine Zahl, Bob schaltet je nach empfangenem Wert seine Augen ein.
"Der «sende Wert»-Block kann eine beliebige Zahl zwischen 0 und 255 senden."
Die SuS schreiben zunächst das Programm für Alice und verwenden hierzu den «sende Wert»-Block. Um ein direktes Feedback zu bekommen wenn ein Wert gesendet wird, schalten sie bei Alice auch ein Auge ein. Als nächstes schreiben sie das Programm für Bob. Bob wartet zunächst mit dem «empfange Wert, warte bis zu 1000 ms»-Block auf eine empfangene Übertragung. Danach überprüfen sie den zuletzt empfangenen Wert mit dem Block «zuletzt empfangener Wert» und schalten - je nachdem - die Augen ein. Zum Schluss testen die SuS die beiden Programme mit ihren Robotern.
Thema: Kommunikation
Bereich: Experimente
Lernziele: Implementation einer Infrarot-Datenübertragung zwischen zwei Robotereinheiten
Anspruch: Mittel
Aufgaben: A1-A10
Zeitbedarf : ca. 30 Minuten
E41.jpgE42.jpgE43.jpgE44.jpgE45.jpgE46.jpgE47.jpg

Download-PDF

Klasse 5-6 W1 - Verzweigung

logo-digital-point-neu.jpg

W1 - Verzweigung

Die Programmstruktur Verzweigung

In dieser Unterrichtseinheit lernen die Schülerinnen und Schüler die Verzweigungs Struktur in allen Varianten kennen. Dazu überprüfen sie Bedingungen und schalten je nachdem ob die Bedingungen wahr oder falsch ist die Leds unterschiedlich ein.
"Falls die Bedingung wahr ist, dann werden die Anweisungen der Sequenz ausgeführt."
Die SuS lernen zunächst die zweiseitige Verzweigung und die Vergleichs-Blöcke kennen und testen diese mit einfachen Wahrheitsaussagen aus. Im späteren Verlauf wird die Verzweigung um weitere «sonst falls dann» Zweige zur Mehrfachverzweigung erweitert. Die wichtige Erkenntnis hierbei ist, dass immer nur genau ein Zweig ausgeführt wird. Anschliessend wird noch der «falls dann»-Block ohne «sonst»-Zweig vorgestellt und damit eine einfache Taschenlampe implementiert. Zur Differenzierung gibt es noch eine Aufgabe in der die Taschenlampe um eine Blinkfunktion erweitert werden soll.
Thema: Verzweigung
Bereich: Wissen
Lernziele: Bedeutung und Anwendung von Verzweigungen, Vergleichsoperatoren, Wahrheitswerte, «falls-dann»-Block
Anspruch: Einfach
Aufgaben: A1-A8
Differenzierung: A9
Zeitbedarf : ca. 30 Minuten

0001.jpg

0001.jpg0002.jpg0003.jpg0004.jpg0005.jpg0006.jpg0007.jpg0008.jpg

Download als PDF

Klasse 5-6 W2 - Schleifen

logo-digital-point-neu.jpg

W2 - Schleifen

Wiederholungen mit Schleifen

In dieser Unterrichtseinheit lernen die Schülerinnen und Schüler die verschiedenen Typen der Kontrollstruktur Schleife kennen, dazu implementieren Schritt für Schritt eine kleine Auto-Diebstahlsicherung.
Mit der «wiederhole bis»-Schleife werden alle Anweisungen innerhalb der Schleife solange wiederholt, bis die Bedingung wahr ist.
Die SuS lernen zunächst die «wiederhole x-mal»-Schleife kennen, und verwenden diese um Bob mit den Augen x-mal zwinkern zu lassen. Anschliessend werden zwei Schleifen nacheinander als Sequenz abgearbeitet um zunächst 8 mal mit dem Auge zu zwinkern, und später 20 mal mit den Bauch-Leds zu blitzen. Im nächsten Schritt lernen die SuS die «wiederhole bis»-Schleife kennen, und setzen sie zum Warten auf eine Objektdetektion des IR-Sensors ein. Während der Wartezeit sollen die Augen immer wieder kurz rot aufglimmen. Wenn ein Objekt detektiert wurde blinken die Bauch-Leds durch Verwendung einer «wiederhole x-mal» 50 mal auf. Als nächstes ersetzen die SuS die «wiederhole x-mal» durch eine «wiederhole fortlaufend»-Schleife und lassen die Bauch-Leds hin und her blinken. In den letzten beiden Aufgaben lernen die SuS den «die Schleife abbrechen»-Block kennen, setzten diese als Alarm-Reset ein und testen den Unterschied zwischen der logischen «und» und der logischen «oder» Verknüpfung.
Thema: Schleifen
Bereich: Wissen
Lernziele: Bedeutung und Anwendung von Schleifen kennenlernen, «wiederholex-mal»-Schleife, «wiederhole bis»-Schleife, Endlosschleife, Abbruch von Schleifen
Anspruch: Mittel
Aufgaben: A1-A9
Zeitbedarf : ca. 30 Minuten
W2 1.jpgw22.jpgw23.jpgw24.jpgw25.jpgw26.jpgw27.jpgw28.jpg

Download-PDF

Klasse 5-6 W3 - Operatoren

logo-digital-point-neu.jpg

W3 - Operatoren

Rechnen mit Operatoren

In dieser Unterrichtseinheit lernen die Schülerinnen und Schüler Operatoren für Zahlen und Wahrheitswerte kennen. Themen sind Rechenoperatoren, Vergleichsoperatoren und logische Operatoren.
"Mit einem Operator lassen sich Konstanten und Variablen zu einem Ausdruck verbinden."
Zunächst bekommen die SuS eine Übersicht über Operatoren. Anschließend wird der Ergebnis-Typ eines Operators erklärt. Mit dem Programm 'Logik-Meister' evaluieren sie verschiedene Ausdrücke und verwenden dazu Operatoren. Im nächsten Schritt kombinieren sie mehrere Operatoren zu komplexeren Ausdrücken. Dazu entwickeln die SuS ein Programm, das Rechenergebnisse anzeigt und überprüfen damit die Rechnekünste von Bob. Zum Abschluss testen die SuS den Vergleichsoperator in Kombination mit Farb-Konstanten.
Thema: Operatoren
Bereich: Wissen
Lernziele: Bedeutung und Anwendung von Operatoren kennenlernen, mit Ganzzahl-Operatoren und dem Farb-Vergleichsoperator arbeiten, Wahrheitswerte kennenlernen und anwenden
Anspruch: Mittel
Aufgaben: A1-A15
Zeitbedarf : ca. 30 Minuten
w31.jpgw32.jpgw33.jpgw34.jpgw35.jpgw36.jpgw37.jpgw38.jpg

Download-PDF

Klasse 5-6 W4 - Variablen

logo-digital-point-neu.jpg

W4 - Variablen

Verwendung von Variablen

In dieser Unterrichtseinheit lernen die Schülerinnen und Schüler Variablen kennen. Die Themen sind lokale Variablen, Deklaration, Wertzuweisung und der Abruf des Variablenwertes.
Eine Variable ist ein Speicherort für Zahlen, Farben oder sonstige Daten.
Die SuS deklarieren zunächst eine lokale Ganzzahlvariable. Anschliessend implementieren sie eine Mehrfachverzweigung für 5 verschiedene Fälle: Der *Fall 1* (ein Eisbär) und der *sonst-Fall* (kein Eisbär) sind vorgegeben. Im nächsten Schritt erweitern die SuS das Programm für den *Fall 2* selbständig mit vorgegebenen Blöcken. Anschließend implementieren die SuS die *Fälle 3* und *4* komplett selbsständig. Mit dem Block «setze Variable auf Wert» testen sie ihr fertiges Programm mit unterschiedlichen Werten der Variablen aus. Als Differenzierungsaufgabe erweitern die SuS das Variablenkonzept auf Farbwerte.
Thema: Variablen
Bereich: Wissen
Lernziele: Bedeutung und Anwendung von Variablen kennenlernen, eine Ganzzahl-Variable deklarieren, initialisieren und im Programm verwenden, Zuweisung von neuen Werten verstehen und anwenden
Anspruch: Mittel
Aufgaben: A1-A8
Differenzierung: A9
Zeitbedarf : ca. 30 Minuten

w41.jpgw42.jpgw43.jpgw44.jpgw45.jpgw46.jpgw47.jpgw48.jpg

Download-PDF

2.2.B Explore Sciencenter

Test 1

2.2.A WvSS