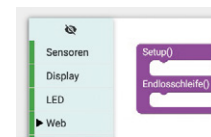


Beim Starten eines neuen Projekts besteht der Code aus einem großen Block, der wiederum aus den Teilen „Setup()“ (beim Start) und „Endlosschleife()“ (dauerhaft) besteht.



Die Programmblöcke, die sich innerhalb des „**Setup()**“-Blocks befinden, werden einmalig beim Starten des Programms auf der senseBox ausgeführt. Hier sollte alles initialisiert werden (z. B. der Bildschirm). Die Blöcke im „**Endlosschleife()**“-Block befinden sich in einer Dauerschleife und werden immer wieder ausgeführt, bis die senseBox ausgeschaltet wird. Durch Kontrollstrukturen wie Verzweigungen und Schleifen kann man die senseBox weiterhin „kontrollieren“. Der Sinn der Endlosschleife besteht darin, dass ein Gerät ständig nach Eingaben (z. B. Taste gedrückt) gucken kann und entsprechend darauf reagiert (z. B. Lampe blinkt). Man spricht hier vom **EVA-Prinzip** (Eingabe – Verarbeitung – Ausgabe).

Klickt man auf eine der Block-Kategorien, so zeigt sich eine Auswahl an Programmierblöcken. Um einen der Codeblöcke zu verwenden, muss dieser mit gehaltener linker Maustaste an die gewünschte Stelle gezogen werden.



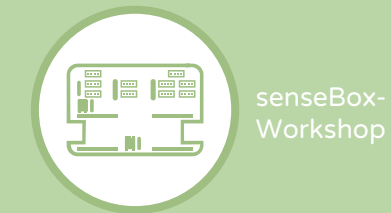
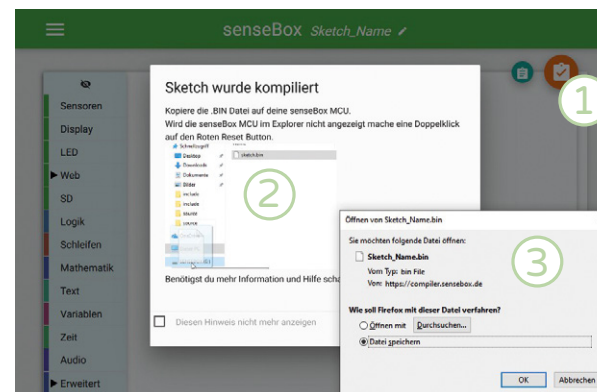
Die Form der einzelnen Blöcke gibt Aufschluss darüber, wo die Blöcke eingehängt werden können.



Um einen Block wieder zu löschen, muss dieser einfach zurück auf die Blockauswahl (egal welche Kategorie) gezogen werden. Alternativ kann der zu löschende Block unten rechts auf den Mülleimer gezogen werden.



Um das Programm auf die senseBox zu laden, klickt man oben rechts auf den orangenen Koffer (1). Wenn das Programm keine Fehler enthält, erscheint ein Pop-up-Fenster mit allen weiteren Anweisungen (2). Kurz darauf erscheint der Downloaddialog (3). Nun muss das Programm auf die per USB-Kabel angeschlossene senseBox gespeichert werden. Damit die senseBox am Computer angezeigt wird, muss zunächst der rote Resetbutton 2 x gedrückt werden.



# HOW TO SENSEBOX

Lehrerversion (mit Musterlösung)

Sekundarstufe 1 & 2

Niveaustufe: Einsteiger

Dauer: 1 Stunde

Sachmittel:  
 ☑ senseBox  
 ☑ PC oder Laptop  
 ☑ senseBox-Lernkarten  
 ([https://sensebox.de/docs/senseBox\\_Lernkarten\\_v2.2\\_online.pdf](https://sensebox.de/docs/senseBox_Lernkarten_v2.2_online.pdf))

Voraussetzungen: keine

## Was ist die senseBox?

Die senseBox ist ein Mikrocontroller, der speziell für den Einsatz in der Digitalen Bildung, Citizen Science und für professionelle Umwelt-Datenerhebung entwickelt wurde. Der Vorteil der senseBox gegenüber anderen Mikrocontrollern (wie z. B. dem Arduino) ist, dass viele verschiedene Sensoren direkt angeschlossen werden können, um eine individuelle Umweltmessstation zu erstellen. Vielseitige Module ermöglichen die Datenübertragung an eine Internetplattform von nahezu jedem Standort aus. So kann man direkt loslegen und frustrationsfrei schnell erste Ergebnisse erhalten. Die senseBox kann sowohl mittels Blocksprache als auch textuell programmiert werden. Dies kann auch während der Programmierung umgeschaltet werden.

## Welche senseBoxen gibt es?

- › Die preiswerte **senseBox:mini** enthält alles Nötige für das erste eigene IoT-Projekt und der enthaltene Umweltsensor ermöglicht die Erfassung einiger Umweltphänomene.
- › Die **senseBox:home** ist für den heimischen Gebrauch als Do-it-yourself-Umweltmessstation (inklusive Gehäuse) konzipiert worden und enthält weitere Umweltsensoren.
- › Die **senseBox:edu** bietet den größten Hardwareumfang und ermöglicht somit eine Vielzahl an Projekten und Experimenten. Alle Bauteile befinden sich in einem robusten Kunststoffkoffer mit Schaumstoffeinlage. Somit ist diese am besten für den Einsatz mit SuS geeignet.

## Was ist ein Mikrocontroller?

Vereinfacht ausgedrückt ist ein Mikrocontroller ein Computer in Miniaturausgabe, ein Einplatinencomputer, an den Sensoren und Aktoren angeschlossen werden können. Damit ein Mikrocontroller das tut, was er soll, muss man ihn programmieren.

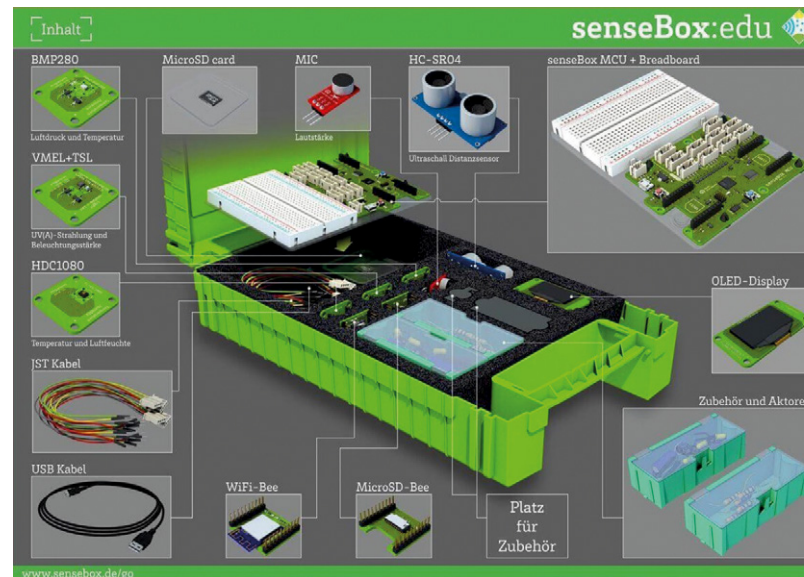
## Was sind Sensoren und Aktoren?

Sensoren sind Bauteile zur Messung der Umgebungsbedingungen. Sie sind somit für die Außenwahrnehmung zuständig. Wären wir

Menschen ein Computer, so wären beispielsweise unsere Augen ein Sensor. Die Aktoren sind das Gegenstück zu den Sensoren. Als Aktor bezeichnet man im Allgemeinen ein Antriebselement. Bei uns Menschen wäre zum Beispiel die Hand ein Aktor.

## Was beinhaltet die senseBox:edu?

Das Herzstück bildet der **Mikrocontroller**, der mit einem Breadboard fest verbunden ist. Dort können Bees (Erklärung folgt gleich), Sensoren und Zubehör angeschlossen werden.



Die **Bees** sind die Schnittstellen der senseBox. Mithilfe der Wifi-Bee können Daten übertragen und mithilfe der SD-Bee Daten auf der senseBox gespeichert werden.

Folgende Sensoren ermöglichen die Messung von Umweltphänomenen (manche Sensoren messen zeitgleich mehrere Phänomene): Temperatur und relative Luftfeuchte; Luftdruck und Temperatur; Beleuchtungsstärke und UV-Strahlung; Ultraschall-Sensor; Mikrofon.

Darüber hinaus liegt folgendes Zubehör bei: ein LED-Display zur Ausgabe von Sensorwerten und Nachrichten, eine Micro-SD-Karte (2 GB), ein USB-Kabel zum Verbinden der senseBox mit dem Computer, Leuchtdioden und dafür benötigte Widerstände, Knöpfe, Drehregler und ein Fotowiderstand.

## Wie werden die Bauteile richtig angeschlossen?

Siehe senseBox-Lernkarten ([https://sensebox.de/docs/senseBox\\_Lernkarten\\_v2.2\\_online.pdf](https://sensebox.de/docs/senseBox_Lernkarten_v2.2_online.pdf)). **Diese werden auch an die SuS zum Nachschlagen ausgeteilt.**

## Wie steuere ich die senseBox?

Damit die senseBox genau das macht, was man gerne möchte, muss man ihr dies mithilfe eines Programmes sagen. Um ein solches Programm zu schreiben und dieses dann auf die senseBox zu übertragen, benötigt man eine Programmierumgebung, die auch Editor genannt wird.

„Programmiert wird die senseBox mit Hilfe der grafischen Programmieroberfläche ‚Blockly für senseBox‘. Diese bietet Blöcke für alle Bauteile und Funktionen der senseBox an, aus denen sich ein Programm im Handumdrehen zusammenbauen lässt – komplett ohne Arbeit am Quellcode! Die Programmieroberfläche ist frei im Web verfügbar, benötigt keinerlei Software- oder Treiberinstallation und somit auch keinerlei Wartung.

Zusätzlich lässt sich die senseBox mit Hilfe von NEPO, der ebenfalls grafischen Programmieroberfläche des Fraunhofer Instituts IAIS, programmieren. Für fortgeschrittene Anwendungen kann die senseBox:edu auch direkt in der Arduino IDE programmiert werden.“<sup>1</sup>

## Wie funktioniert die Programmierumgebung Blockly?

Den in diesen Workshops verwendeten Editor findet man unter <https://blockly.sensebox.de/> › senseBox MCU.

**Tipp:** Sprache auf Deutsch stellen: Auf 3 Striche links oben › Settings › Language › Deutsch.

Der Aufbau dieses Editors ist wie folgt: Links befinden sich die Programmierblöcke, unterteilt in verschiedene Kategorien (1). Im großen weißen Bereich in der Mitte werden die Programmblöcke zu einem Programmcode zusammengefügt (2), der rechts automatisch in textuellen Code übersetzt wird (3).

## 5 Aktion – Reaktion

**Verzweigungen** sind Strukturen, die dazu dienen, alternative Abläufe zu beschreiben. Es handelt sich somit um Fallunterscheidungen, bei welchen ein Codeteil ausgeführt wird, wenn eine Bedingung erfüllt ist, und ein anderer, wenn die Bedingung nicht erfüllt ist.

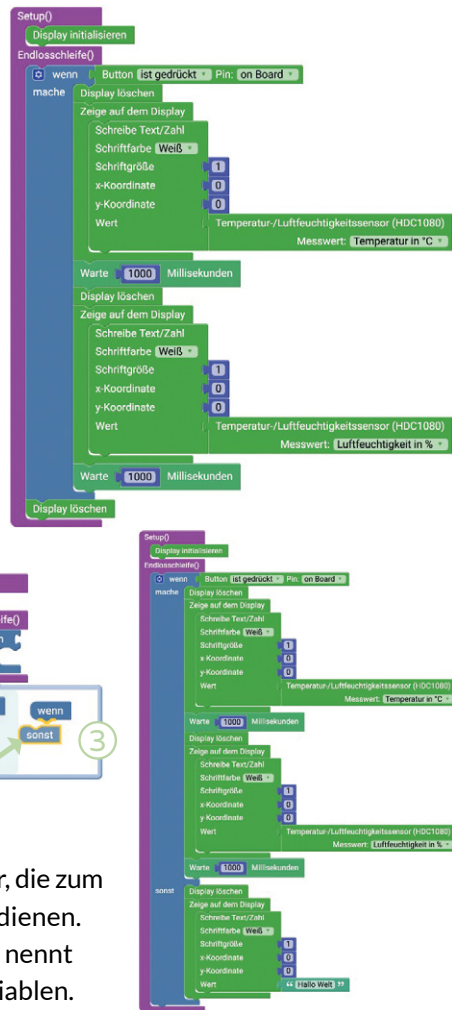
*Beispiel: Wenn es regnet, nehme ich einen Regenschirm mit. Sonst (= wenn es nicht regnet) nehme ich einen Volleyball mit.*

**5.1** Erweitere nun dein Programm von eben. Nur wenn der Button (Knopf, auf Board als „Switch“ bezeichnet) gedrückt wird, sollen die Sensorwerte angezeigt werden.

**5.2** Erweitere dein Programm von eben. Wenn der Knopf nicht gedrückt wird, soll auf dem Display „Hallo Welt“ stehen. Dazu musst du auf das Zahnrad vom „wenn“-Block klicken (1) und den „sonst“-Block (2) in den weißen Bereich unter den „wenn“-Block ziehen (3). Damit das Menü wieder zugeht, klicke erneut auf das Zahnrad.

## 6 Zählen

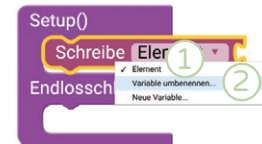
**Variablen**, oft auch Platzhalter genannt, sind Wertespeicher, die zum Merken und Verwalten von Daten dienen. Legt man eine neue Variable an, so nennt man das das Deklarieren einer Variablen. Eine Variable kann einen Wert zugewiesen bekommen, dann entspricht die Variable diesem Wert. Diesen Wert kann man jederzeit ändern. Bekommt eine Variable zu Beginn des Pro-



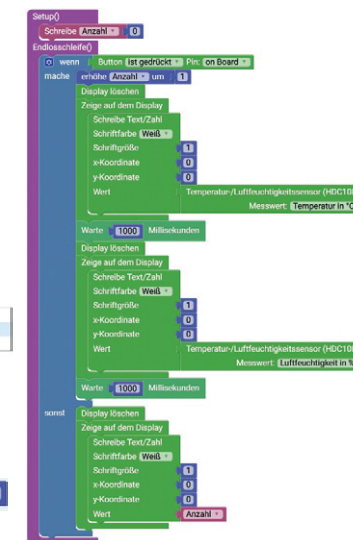
gramms einen Wert zugewiesen, so nennt man das das Initialisieren einer Variablen.

**6.1** Erweitere dein Programm von eben. Die Klicks auf den Button sollen gezählt werden. Die Anzahl der Klicks soll auf dem Display ausgegeben werden, wenn der Button nicht gedrückt wird (statt wie bisher „Hallo Welt“).

Unter der Kategorie „Variablen“ findest du den Block „Schreibe Element“. Diesen fügst du im Setup() ein und setzt den Startwert mithilfe des Blockes „Zahl“ aus der Kategorie „Mathematik“ auf 0. Du kannst die Variable durch einen Klick auf den kleinen Pfeil (1) umbenennen (2), z. B. in „Anzahl“.



Um die Variable zu erhöhen, verwende den Block „erhöhe ... um“ aus der Kategorie „Mathematik“. Jetzt musst du diesen Block nur noch an der richtigen Stelle in deinem Programm einfügen und die Variable auf dem Display anzeigen, damit die Buttonklicks gezählt werden.



Du kennst nun die Grundfunktionen der senseBox.

Du hast noch Zeit? Dann werde selbst kreativ. Hier ein paar Ideen:

- › Bau deine eigene Ampelschaltung. Nutze dafür zusätzlich die RGB-LED, s. Lernkarten SB08 ([https://sensebox.de/projects/de/2019-04-29-ampel\\_blockly.html](https://sensebox.de/projects/de/2019-04-29-ampel_blockly.html)).
- › Es ist zu heiß? Schlag Alarm! Nutze dafür zusätzlich die RGB-LED, s. Lernkarten SB08 (<https://sensebox.de/projects/de/2020-08-11-hitzefrei-alarm>).
- › Es ist zu laut? Schlag Alarm! Nutze dafür zusätzlich den roten Geräuschesensor und die RGB-LED, s. Lernkarten SB08 und SB09.



senseBox-Workshop

# EINSTEIGER

Lehrerversion (mit Musterlösung)

Sekundarstufe 1 & 2

Niveaustufe: Einsteiger

Dauer: 2–3 Stunden

Sachmittel:  
 ☑ senseBox  
 ☑ PC oder Laptop

Voraussetzungen / Kompetenzen:

- ☑ Den SuS ist die senseBox als solche bekannt (vgl. Inhalt von senseBox-Workshop – How to senseBox).
- ☑ Die SuS benötigen keine Erfahrungen in der Programmierung.



Ein **Programm** ist eine Arbeitsanweisung zur Lösung eines Problems, die in einer Programmiersprache so präzise geschrieben ist, dass sie von einer Maschine wie einem Computer, einem Handy oder deiner senseBox ausgeführt werden kann.

Nicht nur in der Informatik nennt man solche Arbeitsanweisungen einen **Algorithmus**. Auch in deinem Alltag kommen Algorithmen vor. Das Backen eines Kuchens ist hierfür das beste Beispiel. Um dieses Problem zu lösen, wird immer gleich vorgegangen und auch das Ergebnis ist immer das Gleiche – ein leckerer Kuchen. Selbst sehr schwierige Algorithmen werden meist in einfachen Teilschritten gelöst.

## 1 Das erste Programm: Blinken

Wenn die senseBox gestartet wird, soll die eingebaute LED „Builtin\_1“ angehen.

LED an digitalen Pin: BUILTIN\_1 Status Ein

Du siehst gar nicht, welche LED angegangen ist?

Dann lassen wir sie blinken! Das geht folgendermaßen:

LED an – 2 Sekunden warten – LED aus – 2 Sekunden warten – LED an – 2 Sekunden warten – ....  
Lass die LED nun 2x blinken.

Warte 1000 Millisekunden

```

Setup()
LED an digitalen Pin: BUILTIN_1 Status Ein
Warte 2000 Millisekunden
LED an digitalen Pin: BUILTIN_1 Status Aus
Warte 2000 Millisekunden
LED an digitalen Pin: BUILTIN_1 Status Ein
Warte 2000 Millisekunden
LED an digitalen Pin: BUILTIN_1 Status Aus
Endlosschleife()
  
```

Strukturen, die bestimmte Befehlsfolgen wiederholt ausführen lassen, nennt man **Schleifen**. Dabei kann die Anzahl der Wiederholungen fest vorgegeben (z. B. *sage 80 x „Hallo“* > *for-Schleife*) oder an eine Bedingung geknüpft sein (z. B. *solange du auf das Handy schaust, lasse den Bildschirm an* > *while-Schleife*). Auch endlose Wiederholungen sind möglich.

## 2 Automatisches Blinken

2.1 Nun soll die LED dauerhaft (endlos) blinken. Schaffst du es, die LED mit nur 4 Blöcken dauerhaft zum Blinken zu bringen?

```

Setup()
Endlosschleife()
  LED an digitalen Pin: BUILTIN_1 Status Ein
  Warte 2000 Millisekunden
  LED an digitalen Pin: BUILTIN_1 Status Aus
  Warte 2000 Millisekunden
  
```

2.2 Erweitere dein Programm, sodass die zweite LED genau dann angeht, wenn die erste LED ausgeht, und umgekehrt.

```

Setup()
Endlosschleife()
  LED an digitalen Pin: BUILTIN_1 Status Ein
  LED an digitalen Pin: BUILTIN_2 Status Ein
  Warte 2000 Millisekunden
  LED an digitalen Pin: BUILTIN_1 Status Aus
  LED an digitalen Pin: BUILTIN_2 Status Aus
  Warte 2000 Millisekunden
  
```

2.3 Verändere dein Programm so, dass die LEDs

- (a) schneller blinken,
- (b) langsamer blinken.

Jetzt weißt du, was der Unterschied zwischen „Setup()“ und „Endlosschleife()“ ist und wie man LEDs zum Blinken bringt. Nun wollen wir uns mit dem Display und Sensoren beschäftigen.

## 3 Das Display

3.1 Schließe nun das Display an einen beliebigen I2C/Wire-Anschluss an. Beim Start der senseBox muss das Display zunächst initialisiert werden. Gib anschließend darauf deinen Namen aus.

„Dein Name“

```

Setup()
  Display initialisieren
Endlosschleife()
  Zeige auf dem Display
    Schreibe Text/Zahl
    Schriftfarbe Weiß
    Schriftgröße 1
    x-Koordinate 0
    y-Koordinate 0
    Wert „Dein Name“
  
```

3.2 Die senseBox soll dich dauerhaft mit „Hallo“ – Pause – „Dein Name“ begrüßen.

Auf dem Display stehen noch andere Buchstaben als gewünscht? Verwende den „Display löschen“-Block, um das Display zu leeren.

3.3 Was ist das längste Wort mit Schriftgröße eins, was noch auf dem Display angezeigt werden kann?

3.4 Was passiert, wenn du die Zahl bei der

- (a) Schriftgröße,
- (b) x-Koordinate,
- (c) y-Koordinate

veränderst?

```

Setup()
  Display initialisieren
Endlosschleife()
  Display löschen
  Zeige auf dem Display
    Schreibe Text/Zahl
    Schriftfarbe Weiß
    Schriftgröße 1
    x-Koordinate 0
    y-Koordinate 0
    Wert „Hallo“
  Warte 1000 Millisekunden
  Display löschen
  Zeige auf dem Display
    Schreibe Text/Zahl
    Schriftfarbe Weiß
    Schriftgröße 1
    x-Koordinate 0
    y-Koordinate 0
    Wert „Dein Name“
  Warte 1000 Millisekunden
  
```

## 4 Sensoren

4.1 Schließe nun den Temperatur-/Feuchtigkeitssensor an einen beliebigen I2C/Wire-Anschluss an. Gib nun abwechselnd die Temperatur und die Luftfeuchtigkeit auf dem Display aus.

Temperatur-/Luftfeuchtigkeitssensor (HDC1080)  
Messwert: Temperatur in °C

4.2 Schließe nun mindestens 2 weitere Sensoren an und gib die entsprechenden Werte auf dem Display aus.

```

Setup()
  Display initialisieren
Endlosschleife()
  Display löschen
  Zeige auf dem Display
    Schreibe Text/Zahl
    Schriftfarbe Weiß
    Schriftgröße 1
    x-Koordinate 0
    y-Koordinate 0
    Wert Temperatur-/Luftfeuchtigkeitssensor (HDC1080) Messwert: Temperatur in °C
  Warte 1000 Millisekunden
  Display löschen
  Zeige auf dem Display
    Schreibe Text/Zahl
    Schriftfarbe Weiß
    Schriftgröße 1
    x-Koordinate 0
    y-Koordinate 0
    Wert Temperatur-/Luftfeuchtigkeitssensor (HDC1080) Messwert: Luftfeuchtigkeit in %
  Warte 1000 Millisekunden
  
```

## 4 SenseBox mit der Wifi-Bee aufrüsten und Programm übertragen

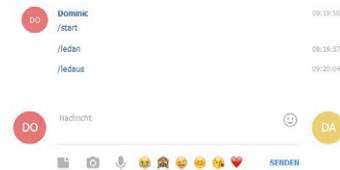
**4.1** Verbinde die Wifi-Bee mit dem XBee-Steckplatz 1. Achte darauf, dass die Richtung des Symbols auf der Wifi-Bee mit der Richtung des Symbols auf dem Steckplatz übereinstimmt.

**4.2** Übertrage dein Programm auf die senseBox.

## 5 Chatbot verwenden

**5.1** Klick nun in Telegram auf den Link, den dir der BotFather geschickt hat. Es sollte sich ein neuer leerer Chat mit deinem Bot öffnen. Klick ggfs. auf „Starten“.

**5.2** Gib /ledan bzw. /ledaus an. Wenn du alles richtig gemacht hast, sollte die LED an- bzw. ausgehen.



Wenn es nicht funktioniert, überprüfe die WLAN-Daten, den Access Token und die Steckrichtung der Wifi-Bee.

## 6 Zeit, selbst kreativ zu werden!

Dir fehlen Ideen? Hier sind ein paar:

- › Schließe Umweltsensoren an und lass dir die Werte vom Chatbot auf dein Handy schicken.
- › Bau dein eigenes „Smart Home“: Schickst du die Nachricht /heizungkalt bzw. /heizungheiß, wird die RGB-LED blau bzw. rot (s. Lernkarte SB08).
- › Finde heraus, welche Nachrichten an deine senseBox geschickt wurden: Zeige auf dem Display alle eingegangenen (und ausgehende) Nachrichten an.
- › Wenn der Button (s. Lernkarte SB10) gedrückt wird, soll eine Warnmeldung verschickt werden.
- › Programmiere deinen eigenen Chatbot: Dein Chatbot soll Witze erzählen und das Wetter ausgeben können.



senseBox-  
Workshop

# FORTGESCHRITTEN TELEGRAM BOT

Lehrerversion (mit Musterlösung)



Sekundarstufe 1 & 2



Niveaustufe:  
Fortgeschritten



Dauer: 2–3 Stunden



Sachmittel:  
☑ senseBox  
☑ PC oder Laptop



Voraussetzungen /  
Kompetenzen:

- ☑ Den SuS ist die senseBox als solche bekannt (vgl. Inhalt von senseBox-Workshop – How to senseBox).
- ☑ Die SuS sind mit den Konzepten der Schleifen und Verzweigungen vertraut (vgl. Inhalt senseBox-Workshop – Einsteiger).

# FORTGESCHRITTEN TELEGRAM BOT



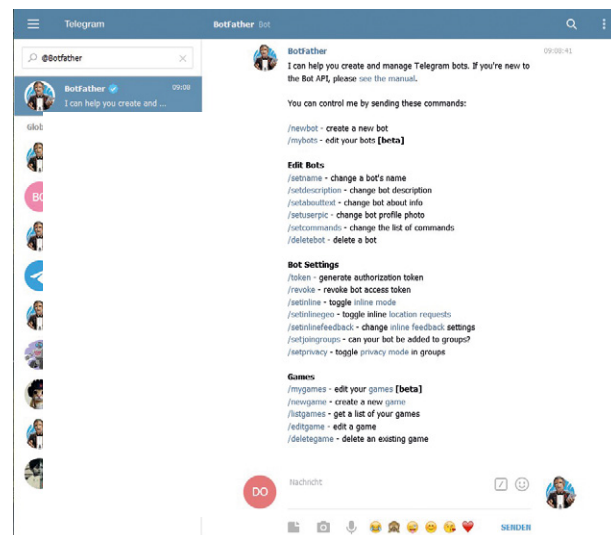
Ein eigener Chatbot ist ein cooles Feature. Man kann zum Beispiel von der ganzen Welt aus erfragen, wie warm es gerade in deinem Zimmer ist, und eine Lampe (bzw. die Heizung) einschalten, damit es gemütlich warm ist, wenn du wieder nach Hause kommst. Oder wenn dir langweilig ist, schickst du ihm die Nachricht „Witz“ und kriegst einen Witz zurückgeschickt. So kannst du fast alles machen, was Siri, Alexa und Co auch können, aber diesmal entscheidest du, was sie sagen!

## 1 Telegram installieren und Chatbot erzeugen

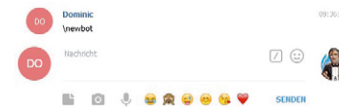
Telegram ist ein Messenger wie WhatsApp, kann aber noch mehr!

**1.1** Falls du den Telegram Messenger noch nicht verwendest, installiere diesen auf deinem Smartphone oder Computer (<https://telegram.org/apps>).

**1.2** Starte Telegram und suche über das Eingabefeld nach **@BotFather** und starte den Chat mit ihm. **BotFather** ist selbst ein Bot und wird genutzt, um eigene Bots zu erstellen oder zu bearbeiten.



**1.3** Erstelle nun mit dem **/newbot**-Befehl im Chat einen neuen Bot.



**1.4** Zunächst brauchen wir einen beliebigen Namen für den Bot, z. B. **Dani** (1).

**1.5** Als Nächstes braucht der Bot noch einen eindeutigen Nutzernamen (username), der auf „bot“ endet, z. B. **senseBoxChatbot** (2).

Wähle einen eigenen Nutzernamen!

Wenn der Nutzernamen noch nicht vergeben war, bekommst du eine Bestätigung, das alles funktioniert hat (3).



In der Bestätigung finden sich der Chatlink zu deinem Bot sowie der rot gedruckte Access Token. Letzteren benötigst du später als Eingabe in der senseBox.

Ein **Access Token** (deutsch: Zugangszeichen) funktioniert wie dein Haustürschlüssel: Wer den Schlüssel hat, kann deine Tür öffnen bzw. den Chatbot steuern.

## 2 senseBox – Setup() programmieren

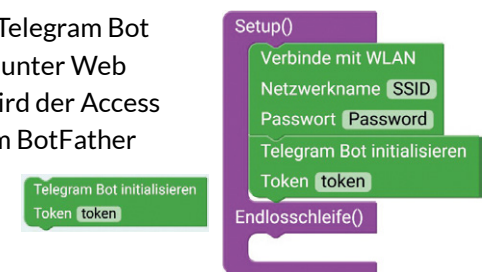
**2.1** Zuerst muss die senseBox über WLAN mit Telegram kommunizieren können. Wähle dafür den „Verbinde mit WLAN“-Block (zu finden unter Web > Wifi) und gebe deine WLAN-Zugangsdaten ein.



Die Abkürzung **SSID** steht für „Service Set Identifier“ und meint den Namen eines WLAN-Netzwerkes.

**WLAN** steht für „Wireless Local Area Network“ (deutsch: Drahtloses Lokales Netzwerk).

**2.2** Anschließend muss der Telegram Bot initialisiert werden (zu finden unter Web > Telegram). Im Feld „Token“ wird der Access Token eingegeben, den du vom BotFather in Telegram erhalten hast.

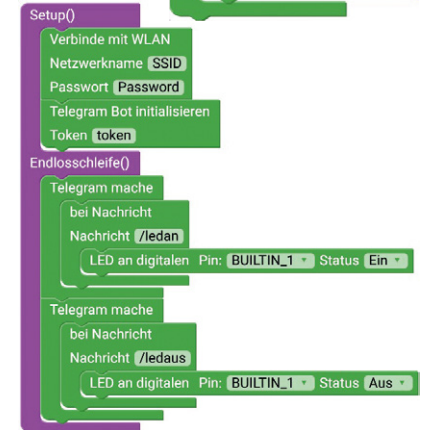


## 3 senseBox – Endlosschleife() programmieren

**3.1** Nun kann man einstellen, was der Bot beim Erhalt einer Nachricht machen soll. Das geschieht im „Telegram mache“-Block. Hier muss der „bei Nachricht“-Block verwendet werden, in dem man die Nachricht angeben kann, bei der etwas passieren soll. Achte darauf, dass die Nachricht mit / beginnt.

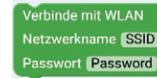


**3.2** Programmiere die senseBox so, dass bei der Nachricht /ledan die interne LED 1 angeschaltet und mit /ledaus die LED ausgeschaltet wird.



### 3 senseBox – Setup() programmieren

**3.1** Zuerst muss die senseBox über WLAN mit dem Internet kommunizieren können. Wähle dafür den „Verbinde mit WLAN“-Block (zu finden unter Web > Wifi) und gib deine WLAN-Zugangsdaten ein.



Die Abkürzung **SSID** steht für „Service Set Identifier“ und meint den Namen eines WLAN-Netzwerkes.

**WLAN** steht für „Wireless Local Area Network“ (deutsch: Drahtloses Lokales Netzwerk).

### 4 senseBox – Endlosschleife() programmieren

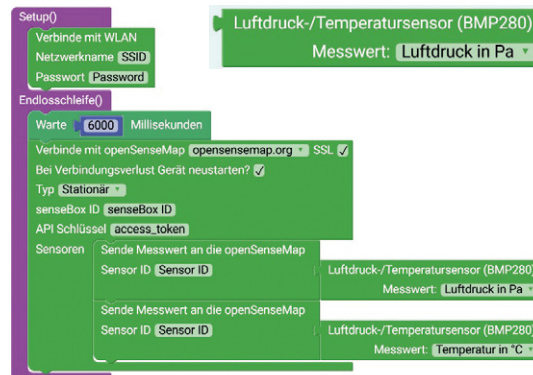
**4.1** Anschließend soll die senseBox in regelmäßigen Abständen (z. B. alle 60 Sekunden) die Sensorwerte an die openSenseMap schicken. Verwende dazu die Blöcke „Verbinde mit openSenseMap“ und für jeden Sensorwert den Block „Sende Messwerte an die openSenseMap“ aus der Kategorie Web > openSenseMap.



Ein **Access Token** (deutsch: Zugangszeichen) funktioniert wie dein Haustürschlüssel: Wer den Schlüssel hat, kann deine Tür öffnen bzw. den Chatbot steuern.

**4.2** Trage an den weiß markierten Stellen die entsprechenden Werte aus der openSenseMap ein und füge die passenden Sensoren hinzu.

Unter dem Punkt „Typ“ sollte „Stationär“ stehen. Ansonsten werden aktuelle GPS-Daten benötigt.



**4.3** Übertrage das Programm auf deine senseBox.

### 5 Überprüfen, ob Daten korrekt übertragen werden

**5.1** Gehe auf <https://opensensemap.org/> und gib oben in der Suchleiste deine Box ein oder such händisch nach dem von dir eingestellten Ort.

Wird deine senseBox nicht gefunden, schaue nach, ob die Wifi-Bee richtig sitzt und ob du die WLAN- und openSenseMap-Daten korrekt eingegeben hast.

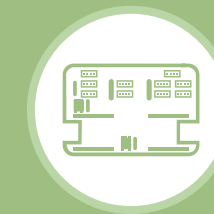
**5.2** Glückwunsch! Mit einem Klick auf deine senseBox kannst du die von dir übertragenen Daten ansehen.

**5.3** Füge ggfs. weitere Sensoren hinzu, indem du rechts oben auf deinen Benutzer > Dashboard klickst. Im nächsten Fenster unter deiner senseBox auf Edit > links: Sensors > rechts oben: + klicken.

### 6 Zeit, selbst kreativ zu werden!

Dir fehlen Ideen? Hier sind ein paar:

- › Schließ eine Powerbank an deine Wetterstation an, damit du mobil wirst, mache einen Spaziergang und schau dir im Nachhinein die Werte an.
- › Gib die Werte zusätzlich auf dem Display aus.
- › Lasse die RGB-LED blinken, wenn die Werte übertragen werden (s. Lernkarte SB08).
- › Starte die Übertragung mit einem Klick auf den Button (s. Lernkarte SB10).
- › Lasse dir die Werte direkt auf dein Handy schicken. Schau dir dazu den Workshop zum Thema „Telegram Bot“ an.



senseBox-  
Workshop

# FORTGESCHRITTEN OPENSENSEMAP 1

Lehrerversion (mit Musterlösung)



Sekundarstufe 1 & 2



Niveaustufe:  
Fortgeschritten



Dauer: 2–3 Stunden



Sachmittel:  
☑ senseBox  
☑ PC oder Laptop  
☑ ggfs. Powerbank



Voraussetzungen /  
Kompetenzen:

- ☑ Den SuS ist die senseBox als solche bekannt (vgl. Inhalt von senseBox-Workshop – How to senseBox).
- ☑ Die SuS sind mit den Konzepten der Schleifen und Verzweigungen vertraut (vgl. Inhalt senseBox-Workshop – Einsteiger).



**HOPP FOUNDATION**  
for computer literacy & informatics



Deine Messdaten im Internet? Für jeden – und damit auch für dich – von überall verfügbar? Das geht mit der senseBox und der openSenseMap! Gleich wirst du eine Umweltstation aufbauen, die verschiedene Umweltphänomene wie Temperatur, Luftfeuchte, Helligkeit und Luftdruck erfasst und im Internet auf der Plattform openSenseMap teilt.



Dabei werden Daten wie z. B. die senseBox ID und die Temperatur von der senseBox zur openSenseMap übertragen.



Menschen mit Internetzugang können sich die Daten anschauen.

Als **Open Data** (deutsch: offene Daten) werden Daten bezeichnet, die von jeder und jedem zu jedem Zweck eingesehen, verwendet und weiterverbreitet werden dürfen.

*Beispiel: Feinstaub-Sensoren in Stuttgart messen die Luftverschmutzung. So wird die Datenlage verbessert und die Diskussion über das Umwelt- und Gesundheitsproblem erleichtert (s. <https://www.codefor.de/blog/stadtgeschichte-feinstaub-stuttgart/>).*

## 1 Sensoren und Wifi-Bee auswählen und anschließen

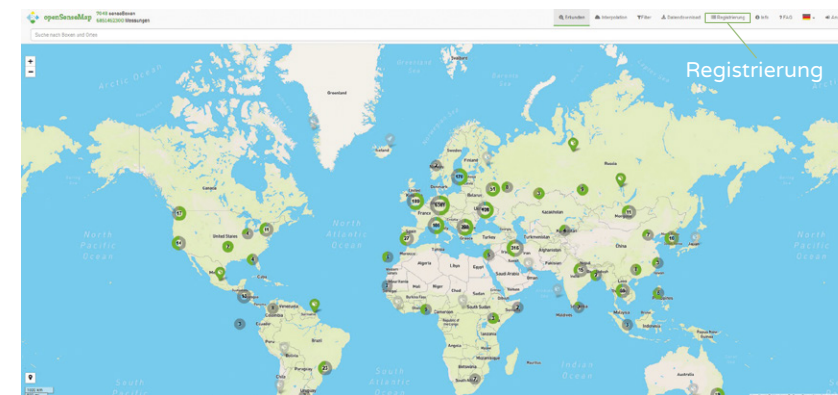
**1.1** Wähle mindestens zwei Umweltsensoren aus, mit denen du Umweltphänomene messen willst.

**1.2** Schließe die Umweltsensoren mithilfe der JST-Kabel an den I2C-Anschlüssen deiner senseBox an.

**1.3** Verbinde die Wifi-Bee mit dem XBee-Steckplatz 1. Achte darauf, dass die Richtung des Symbols auf der Wifi-Bee mit der Richtung des Symbols auf dem Steckplatz übereinstimmt.

## 2 Registrierung auf der openSenseMap

**2.1** Gehe auf <https://opensensemap.org/>.



**2.2** Klicke auf „Registrierung“.

**2.3** Fülle die Felder aus. Merke dir deine Mailadresse und dein Passwort gut! Falls du nicht deine eigene Mailadresse, sondern eine temporäre Mailadresse nutzen möchtest, gehe auf <https://10minemail.com/de/>. Nach dem Klick auf „Registrieren“ musst du noch der Erklärung über die Verwendung deiner Daten zustimmen („I agree“ ankreuzen + „Next“ drücken).

**2.4** Nachdem dein Benutzeraccount angelegt wurde, musst du nun deine senseBox anlegen.

Wähle einen Namen für deine senseBox (1), gib an, ob deine senseBox drinnen oder draußen steht oder mobil ist (2). Den „Group identifier“ kannst du unverändert bei „edu“ lassen (3). Gib auf der Karte an, wo deine senseBox steht (4).

**2.5** Wähle nun die senseBox:edu aus (5) und füge die von dir angeschlossenen Sensoren hinzu. Klicke dazu auf „Add Sensor“ (6). Den Typ des Sensors (z. B. BMP280) findest du auf der Rückseite des jeweiligen Sensors. Beachte, dass viele Sensoren zwei Umweltphänomene messen. Füge in dem Fall z. B. Airpressure und Temperature hinzu. Du kannst auch im Nachhinein weitere Sensoren hinzufügen.

**2.6** Kontrolliere unten, ob alle Umweltsensoren korrekt hinzugefügt wurden. Falls ja, klicke auf „Next“ und bestätige auf der nächsten Seite deine Eingaben mit „Finish“.

**2.7** Die senseBox ID, den Access Token und die Sensoren-IDs (grün markiert) wirst du später bei der Programmierung benötigen. Lasse die Webseite deshalb offen. Die Sensoren-IDs werden benutzt, um die Sensoren zu unterscheiden. Jede senseBox und jeder Sensor, der auf der openSenseMap registriert ist, besitzt eine einzigartige ID. Beim Hochladen der Messwerte werden diese IDs benötigt, damit die openSenseMap weiß, zu welchen Sensoren die Messwerte gehören.

**General**

Please submit additional information about your senseBox. This information can be changed after registration as well.

Name of your station: 1

Exposure: 2

Group identifier: 3

Location: 4

Hardware: 5

6

Summary

Phenomenon	Unit	Type
Luftdruck	hPa	BMP280
Temperatur	°C	BMP280

Access Token: [grün markiert]

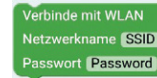
Sensors & IDs

Phenomenon	Unit	Type
Luftdruck (hPa) - BMP280	hPa	BMP280
Temperatur (°C) - BMP280	°C	BMP280



### 3 senseBox – Setup() programmieren

**3.1** Zuerst muss die senseBox über WLAN mit dem Internet kommunizieren können. Wähle dafür den „Verbinde mit WLAN“-Block (zu finden unter Web > Wifi) und gib deine WLAN-Zugangsdaten ein.



Die Abkürzung **SSID** steht für „Service Set Identifier“ und meint den Namen eines WLAN-Netzwerkes.

**WLAN** steht für „Wireless Local Area Network“ (deutsch: Drahtloses Lokales Netzwerk).

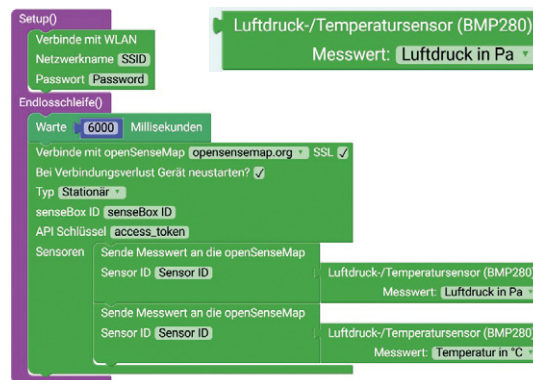
### 4 senseBox – Endlosschleife() programmieren

**4.1** Anschließend soll die senseBox in regelmäßigen Abständen (z. B. alle 60 Sekunden) die Sensorwerte an die openSenseMap schicken. Verwende dazu die Blöcke „Verbinde mit openSenseMap“ und für jeden Sensorwert den Block „Sende Messwerte an die openSenseMap“ aus der Kategorie Web > openSenseMap.



Ein **Access Token** (deutsch: Zugangszeichen) funktioniert wie dein Haustürschlüssel: Wer den Schlüssel hat, kann deine Tür öffnen bzw. den Chatbot steuern.

**4.2** Trage an den weiß markierten Stellen die entsprechenden Werte aus der openSenseMap ein und füge die passenden Sensoren hinzu.



Unter dem Punkt „Typ“ sollte „Stationär“ stehen. Ansonsten werden aktuelle GPS-Daten benötigt.

**4.3** Übertrage das Programm auf deine senseBox.

### 5 Überprüfen, ob Daten korrekt übertragen werden

**5.1** Gehe auf <https://opensensemap.org/> und gib oben in der Suchleiste deine Box ein oder such händisch nach dem von dir eingestellten Ort.

Wird deine senseBox nicht gefunden, schaue nach, ob die Wifi-Bee richtig sitzt und ob du die WLAN- und openSenseMap-Daten korrekt eingegeben hast.

**5.2** Glückwunsch! Mit einem Klick auf deine senseBox kannst du die von dir übertragenen Daten ansehen.

**5.3** Füge ggfs. weitere Sensoren hinzu, indem du rechts oben auf deinen Benutzer > Dashboard klickst. Im nächsten Fenster unter deiner senseBox auf Edit > links: Sensors > rechts oben: + klicken.

### 6 Zeit, selbst kreativ zu werden!

Dir fehlen Ideen? Hier sind ein paar:

- › Schließ eine Powerbank an deine Wetterstation an, damit du mobil wirst, mache einen Spaziergang und schau dir im Nachhinein die Werte an.
- › Gib die Werte zusätzlich auf dem Display aus.
- › Lasse die RGB-LED blinken, wenn die Werte übertragen werden (s. Lernkarte SB08).
- › Starte die Übertragung mit einem Klick auf den Button (s. Lernkarte SB10).
- › Lasse dir die Werte direkt auf dein Handy schicken. Schau dir dazu den Workshop zum Thema „Telegram Bot“ an.
- › Dein eigenes Spiel: Fang den Punkt (<https://sensebox.de/projects/de/2019-12-08-punktefang>).
- › Heimserver (<https://sensebox.de/projects/de/2019-11-25-bme680-webserver>).



senseBox-  
Workshop

# FORTGESCHRITTEN OPENSENSEMAP 2

Lehrerversion (mit Musterlösung)



Sekundarstufe 1 & 2



Niveaustufe:  
Fortgeschritten



Dauer: 2–3 Stunden



Sachmittel:  
☑ senseBox  
☑ PC oder Laptop  
☑ ggfs. Powerbank



Voraussetzungen /  
Kompetenzen:

- ☑ Den SuS ist die senseBox als solche bekannt (vgl. Inhalt von senseBox-Workshop – How to senseBox).
- ☑ Die SuS sind mit den Konzepten der Schleifen und Verzweigungen vertraut (vgl. Inhalt senseBox-Workshop – Einsteiger).



**HOPP FOUNDATION**  
for computer literacy & informatics

Deine Messdaten im Internet? Für jeden – und damit auch für dich – von überall verfügbar? Das geht mit der senseBox und der openSenseMap! Gleich wirst du eine Umweltstation aufbauen, die verschiedene Umweltphänomene wie Temperatur, Luftfeuchte, Helligkeit und Luftdruck erfasst und im Internet auf der Plattform openSenseMap teilt.



Dabei werden Daten wie z. B. die senseBox ID und die Temperatur von der senseBox zur openSenseMap übertragen.



Als **Open Data** (deutsch: offene Daten) werden Daten bezeichnet, die von jeder und jedem zu jedem Zweck eingesehen, verwendet und weiterverbreitet werden dürfen.

*Beispiel: Feinstaub-Sensoren in Stuttgart messen die Luftverschmutzung. So wird die Datenlage verbessert und die Diskussion über das Umwelt- und Gesundheitsproblem erleichtert (s. <https://www.codefor.de/blog/stadtgeschichte-feinstaub-stuttgart/>).*

## 1 Sensoren und Wifi-Bee auswählen und anschließen

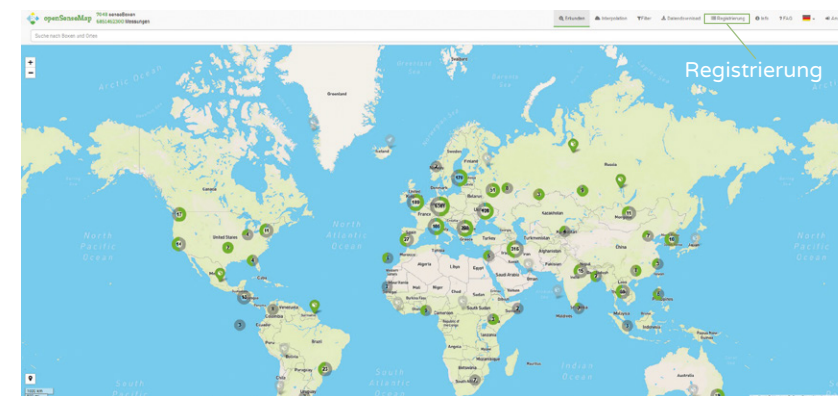
**1.1** Wähle mindestens zwei Umweltsensoren aus, mit denen du Umweltphänomene messen willst.

**1.2** Schließe die Umweltsensoren mithilfe der JST-Kabel an den I2C-Anschlüssen deiner senseBox an.

**1.3** Verbinde die Wifi-Bee mit dem XBee-Steckplatz 1. Achte darauf, dass die Richtung des Symbols auf der Wifi-Bee mit der Richtung des Symbols auf dem Steckplatz übereinstimmt.

## 2 Registrierung auf der openSenseMap

**2.1** Gehe auf <https://opensensemap.org/>.



**2.2** Klicke auf „Registrierung“.

**2.3** Fülle die Felder aus. Merke dir deine Mailadresse und dein Passwort gut!

Falls du nicht deine eigene Mailadresse, sondern eine temporäre Mailadresse nutzen möchtest, gehe auf <https://10minemail.com/de/>. Nach dem Klick auf „Registrieren“ musst du noch der Erklärung über die Verwendung deiner Daten zustimmen („I agree“ ankreuzen + „Next“ drücken).

**2.4** Nachdem dein Benutzeraccount angelegt wurde, musst du nun deine senseBox anlegen.

Wähle einen Namen für deine senseBox (1), gib an, ob deine senseBox drinnen oder draußen steht oder mobil ist (2). Den „Group identifier“ kannst du unverändert bei „edu“ lassen (3). Gib auf der Karte an, wo deine senseBox steht (4).

**2.5** Wähle nun die senseBox:edu aus (5) und füge die von dir angeschlossenen Sensoren hinzu. Klicke dazu auf „Add Sensor“ (6). Den Typ des Sensors (z. B. BMP280) findest du auf der Rückseite des jeweiligen Sensors. Beachte, dass viele Sensoren zwei Umweltphänomene messen. Füge in dem Fall z. B. Airpressure und Temperature hinzu. Du kannst auch im Nachhinein weitere Sensoren hinzufügen.

**2.6** Kontrolliere unten, ob alle Umweltsensoren korrekt hinzugefügt wurden. Falls ja, klicke auf „Next“ und bestätige auf der nächsten Seite deine Eingaben mit „Finish“.

**2.7** Die senseBox ID, den Access Token und die Sensoren-IDs (grün markiert) wirst du später bei der Programmierung benötigen. Lasse die Webseite deshalb offen. Die Sensoren-IDs werden benutzt, um die Sensoren zu unterscheiden. Jede senseBox und jeder Sensor, der auf der openSenseMap registriert ist, besitzt eine einzigartige ID. Beim Hochladen der Messwerte werden diese IDs benötigt, damit die openSenseMap weiß, zu welchen Sensoren die Messwerte gehören.