

</> Programmierung

Grundlagen: senseBox

Speziell für den Einsatz im Bildungsbereich wurde die senseBox:edu entwickelt. In einer praktischen grünen Kunststoffkiste sind alle Sensoren, Datenübertragungsmodule und weitere elektronische Bauteile gut sortiert. Spannende Projekte und viele weitere kostenlose Lernmaterialien ermöglichen einen einfachen und unkomplizierten Einstieg.

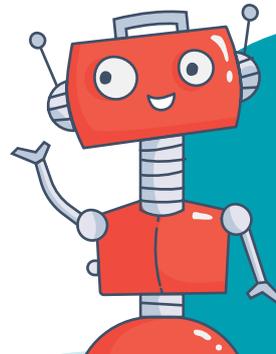


Level:

Für Fortgeschrittene



Wir freuen uns über euer Feedback zu unserem Material!
post@junge-tueftler.de / Mehr Materialien unter: tueftellab.de





Art:
Lernkarten



Alter:
Ab 12 Jahren



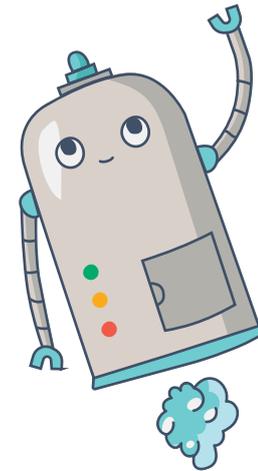
Kategorie:
Programmieren



Level:
Für Fortgeschrittene

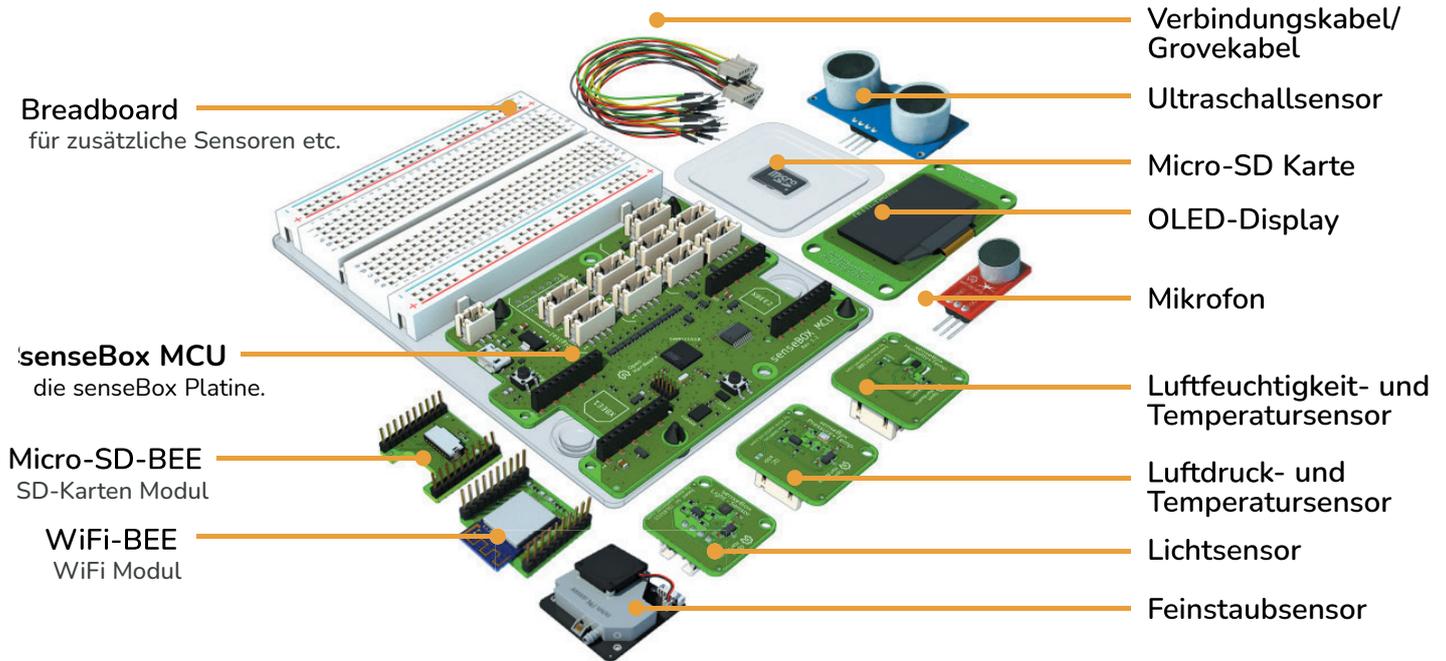
Lernziele

- Schulung des Umgangs mit Computer-Hardware
- Aufbau und Nutzung von visuellen Programmiersprachen verstehen
- Verständnis über Verwendung und Möglichkeiten von Variablen
- Chancen und Schwierigkeiten bei der Programmierung und Nutzung von Platinen und Zubehör wie Sensoren oder Speichermedien



Die Bestandteile

senseBox Grundlagen



Die Bestandteile

senseBox Grundlagen

Reset und senseBox

Lernmodus Button

für Lernmodus 2x drücken

XBEE1-Anschluss

für XBEE WLAN-Modul

Button

Eingabe, Interaktion

XBEE2-Anschluss

für XBEE SD-Karten Modul

Mirko-USB-Anschluss

Übertragung von Programme,
Stromversorgung

Akku-Anschluss

für Batteriehalter oder
LiPo Akku

Digital/Analog Ports

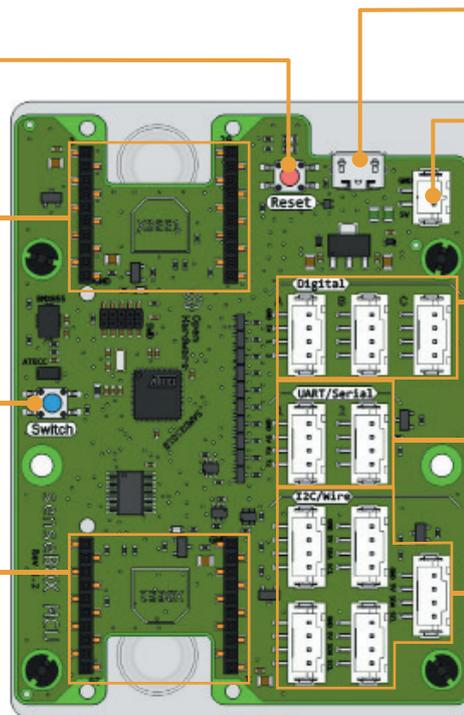
zum Anschließen von
Sensoren und Aktoren über
das Breadboard

UART/Serial Ports

für Feinstaubsensoren

I2C/Wire-Ports

für Umweltsensoren und
Display, I2C/ Wire-Ports

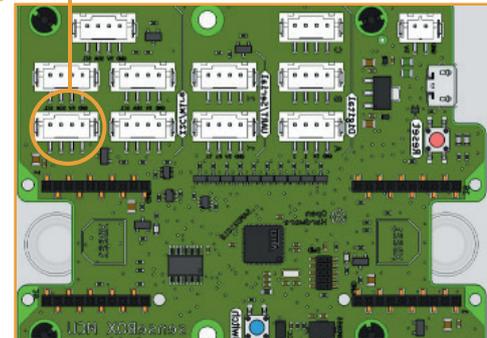
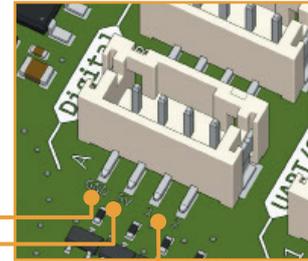


Steckverbindung

Sensoren oder Aktoren an Digital/Analog Ports anschließen

Jeder Digital/Analog-Port der senseBox MCU hat vier verschiedene Pins:

1. Der GND-Pin ist der Minuspol und muss immer mit dem schwarzen Kabel verbunden werden
2. Der 5V-Pin dient zur Stromversorgung der Sensoren und wird mit dem roten Kabel verbunden
3. Die mit 1 und 2 beschrifteten Pins sind die digitalen bzw. analogen Pins 1 und 2. Diese Nummerierung läuft fort bis zum Pin 6 an Port Digital C

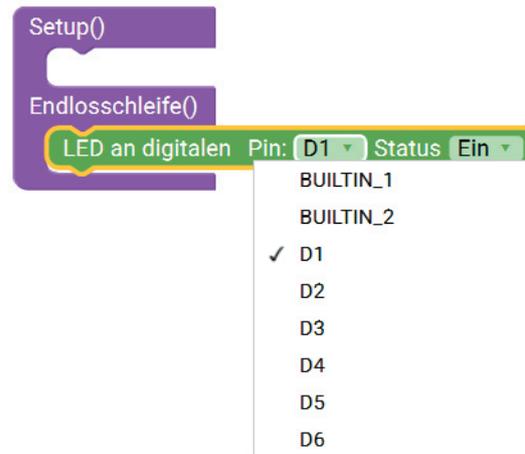


Steckverbindung

Sensoren oder Aktoren an Digital/Analog Ports anschließen

Damit die senseBox weiß, welchen **Verbraucher** oder welche **Ausgabe** sie nutzen soll, müsst ihr dies in euren Programmen festlegen.

Habt ihr ein LED an Pin D1 (Digital 1) angeschlossen, würde der Block also wie folgt aussehen:



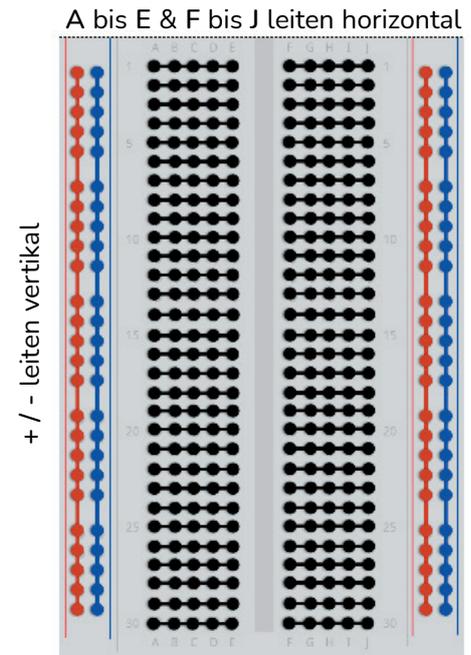
Das Breadboard

Mit dem Steckbrett Schaltungen verbinden

Mit dem Breadboard (Steckbrett), lassen sich Schaltungen ohne Löten ganz einfach verbinden. Die elektronischen Bauteile oder Stecker werden einfach in die Federkontakte gesteckt.

Das Breadboard besteht aus jeweils zwei langen Reihen für die Plus- und Minus-Anschlüsse, sowie zweimal 30 Reihen mit je fünf Kontakten.

Die Kontakte sind, wie in der Grafik dargestellt, leitend miteinander verbunden.

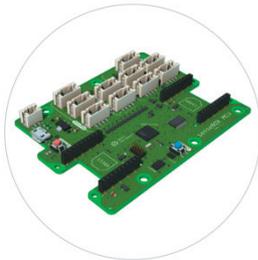


Die Programmieroberfläche

Programmieren mit Blockly für senseBox

Mit Hilfe der Lernkarten lernt ihr, wie die senseBox programmiert wird.
Dazu nutzt ihr den Browser eures Computers:

1. Geht zu »<https://blockly.senseBox.de/>«.
2. Wähle euer senseBox Modell.



senseBox MCU



senseBox MCU-S2



senseBox MCU-S2

Die Programmieroberfläche

Programmieren mit Blockly für senseBox

Programmieren: Zieht die Blöcke aus dem Menü auf die Programmieroberfläche und fügt sie aneinander (Drag-and-Drop).

The image shows the senseBox Blockly programming environment. It features a top navigation bar with a hamburger menu, the text 'senseBox Blockly', and a language dropdown set to 'Deutsch'. Below the bar is a toolbar with icons for saving, undo, redo, and other functions. The main workspace is a grid where green Blockly blocks are being assembled. A 'Display' block is being dragged from the left-hand 'Programmierblock-Bibliothek' (block library) into the workspace. The workspace contains several 'Zeige auf dem Display' blocks. On the right, a code editor shows the corresponding Arduino C++ source code, including an include statement for 'senseBox10.h' and a 'void setup()' function. A hand cursor is pointing at a block in the workspace.

Einstellungen
z.B. Sprache einstellen und senseBox Modell auswählen

Programmstart-Block
Die Setup-Funktion wird einmalig zum Programmstart ausgeführt, die Endlosschleife durchgehend wiederholt

Programmierblock-Bibliothek

Programm-Namen ändern

Programm herunterladen

Programmcode kopieren

Projekt herunterladen

Projekt öffnen

Workspace zurücksetzen

Programmier-Bereich

Arduino Quellcode
Übersetzung der Blöcke

Ein Programm auf die senseBox laden

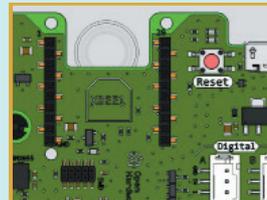
Programm herunterladen und auf der senseBox speichern

1. Verbindet eure senseBox mit Hilfe des USB-Kabels mit dem Computer
2. Klickt auf das Programm herunterladen Icon
3. Kopiert die heruntergeladene Datei aus dem Download-Ordner auf das senseBox Laufwerk
4. Nach dem Speichern startet das Programm auf der senseBox automatisch.



TIPP

Drückt doppelt auf den roten Knopf auf der senseBox, um die senseBox als Austauschlaufwerk zu sehen.
(Die rote LED sollte leuchten)



senseBox Lernkarten

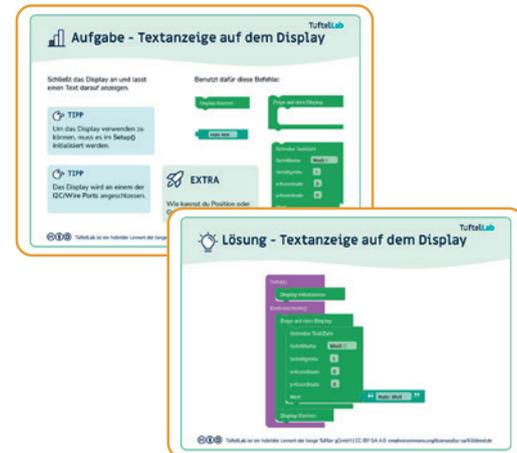
Jetzt kann es losgehen!

1. Die Lernkarten haben eine Vorder- und Rückseite. Auf der Vorderseite befinden sich die Aufgaben, auf der Rückseite ein Vorschlag, der euch beim Lösen der Aufgabe hilft.

2. Die Balken geben den Schwierigkeitsgrad an. Je mehr Balken eingefärbt sind, desto anspruchsvoller ist die Aufgabe.



3. Die Lernkarten können als Set an die Schülerinnen und Schüler verteilt werden und dienen dem eigenständigen Lernen.





Aufgabe - Textanzeige auf dem Display

Schließt das Display an und lasst einen Text darauf anzeigen.



TIPP

Um das Display verwenden zu können, muss es im **Setup()-Block** initialisiert werden.



TIPP

Das Display wird an einem der **I2C/Wire Ports** angeschlossen.

Benutzt dafür diese Befehle:

Display löschen

Hallo Welt

Zeige auf dem Display

Schreibe Text/Zahl

Schriftfarbe

Schriftgröße

x-Koordinate

y-Koordinate

Wert

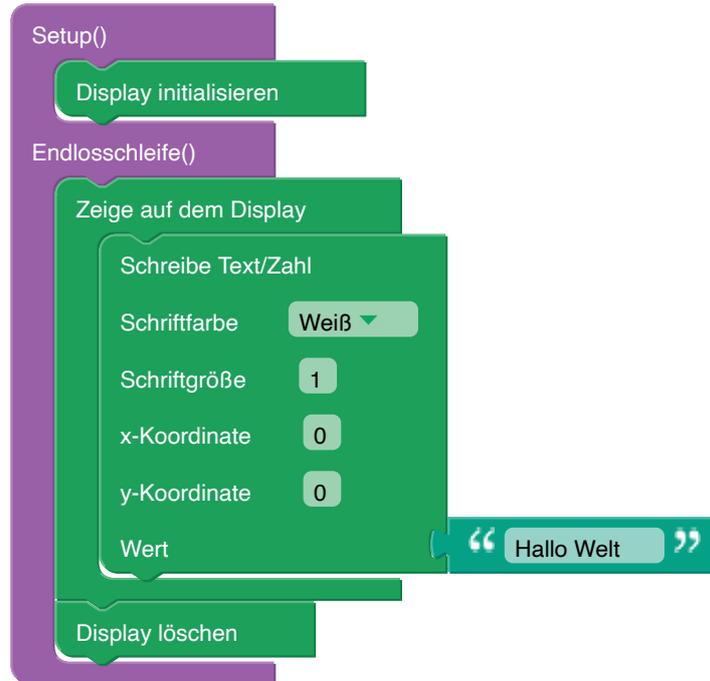


EXTRA

Wie könnt ihr Position oder Größe des Textes verändern?



Lösung - Textanzeige auf dem Display





Aufgabe - Messwert anzeigen

Schließt den Luftdrucksensor an und lasst den Messwert auf dem Display anzeigen.



TIPP

Die Umweltsensoren werden an einem der I2C/Wire Ports angeschlossen.



EXTRA

Schließt nacheinander die anderen Umweltsensoren an und lasst euch Messwerte anzeigen.

Benutzt dafür diese Befehle:

Display löschen

Luftdruck-/Temperatursensor (BMP280)

Messwert:

Zeige auf dem Display

Schreibe Text/Zahl

Schriftfarbe

Schriftgröße

x-Koordinate

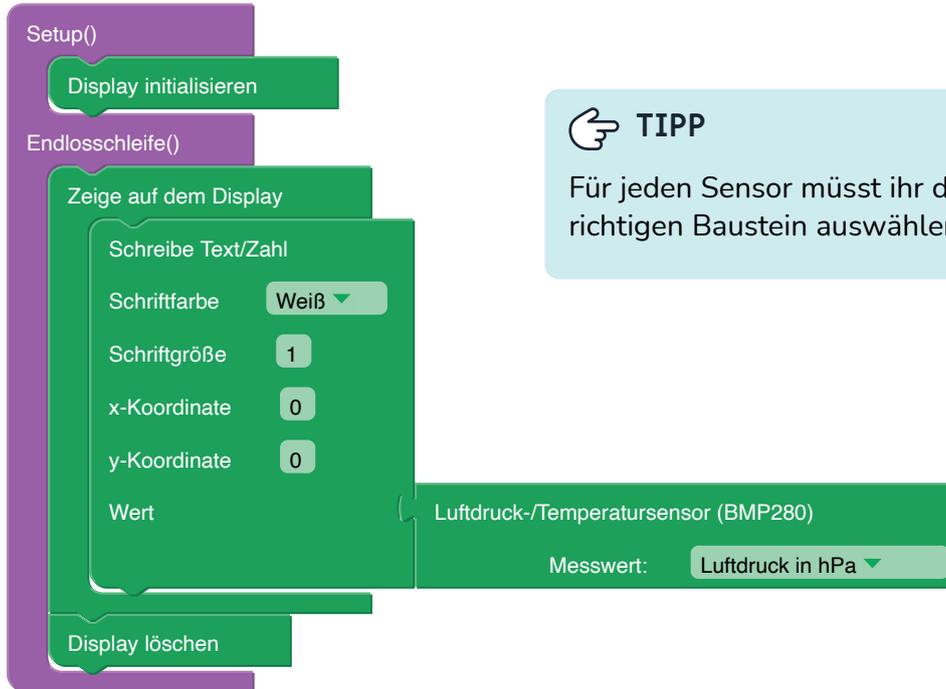
y-Koordinate

Wert

Display initialisieren



Lösung - Messwert anzeigen



TIPP

Für jeden Sensor müsst ihr den richtigen Baustein auswählen.



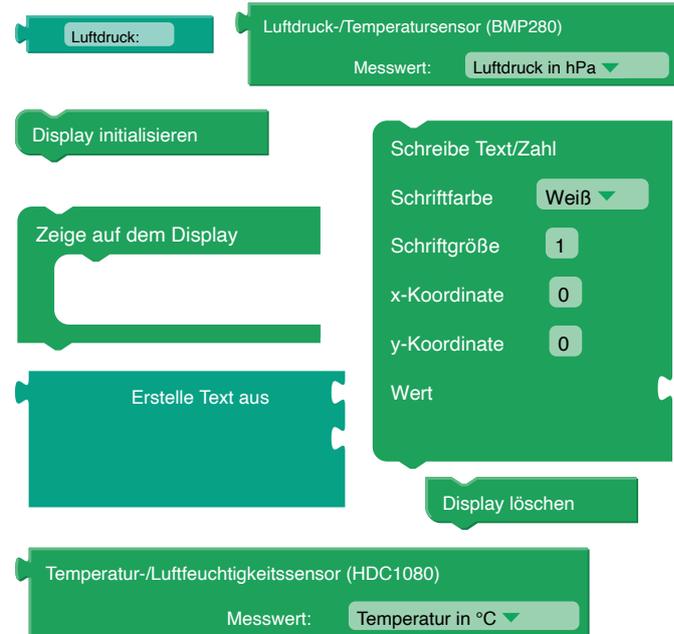
Aufgabe - Messwert beschriften

Schließt den Luftdruck- und Temperatursensor an und lasst beide Werte jeweils zusammen mit einer eigenen Beschriftung anzeigen.

TIPP

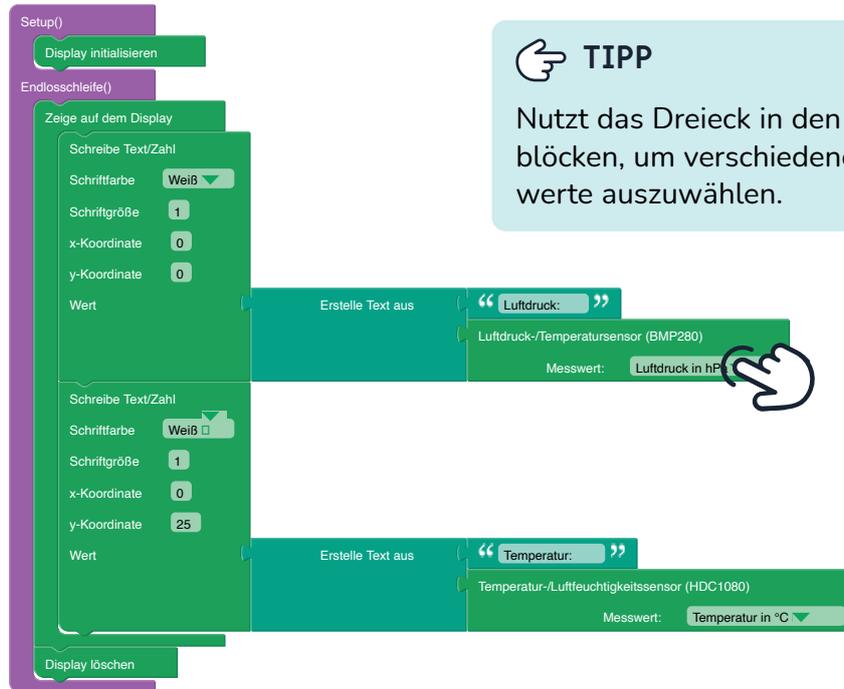
Das Display hat eine Auflösung von 128 x 64 Pixel.

Benutzt dafür diese Befehle:





Lösung - Messwert beschriften

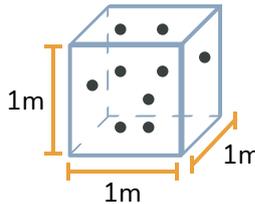


Feinstaub messen

Feinstaubsensor

Mit dem Feinstaubsensor könnt ihr die Feinstaub-Belastung in eurer Umgebung messen. Der Sensor misst den Feinstaub in Mikrogramm pro Kubikmeter ($\mu\text{g}/\text{m}^3$).

Das gibt die Menge Feinstaub an, die sich in einem "Würfel" von $1\text{m} \times 1\text{m} \times 1\text{m}$ Luft befindet.



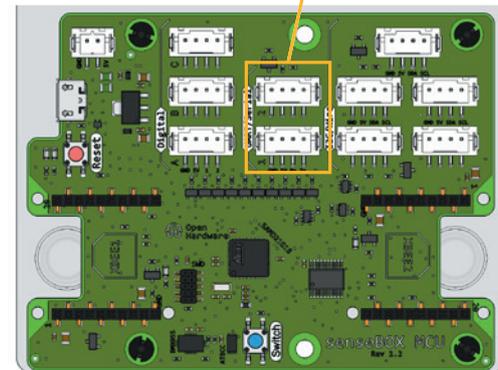
Es können zwei verschiedenen Partikelgrößen gemessen werden. Dies legt ihr im Programmblock des Feinstaubsensors fest.

PM10: Lässt Feinstaubpartikel, die kleiner als 10 Mikrometer (μm) sind, messen.

PM2.5: Lässt Feinstaubpartikel, die kleiner als 2,5 Mikrometer (μm) sind, messen.



UART/Serial





Aufgabe - Feinstaub messen

Misst die Feinstaubbelastung in eurer Umgebung und zeigt sie auf dem Display an.

TIPP

Der Feinstaubsensor wird an einem der UART/Serial Ports angeschlossen. Merkt euch, welchen Port (1 oder 2) ihr benutzt habt.

Benutzt dafür diese Befehle:

Display löschen

Display initialisieren

Zeige auf dem Display

Schreibe Text/Zahl

Schriftfarbe

Schriftgröße

x-Koordinate

y-Koordinate

Wert

Feinstaubsensor SDS011

Messwert: in $\mu\text{g}/\text{m}^3$ an



Lösung - Feinstaub messen

Setup()

Display initialisieren

Endlosschleife()

Zeige auf dem Display

Schreibe Text/Zahl

Schriftfarbe

Schriftgröße

x-Koordinate

y-Koordinate

Wert

Feinstaubsensor SDS011

Messwert: in $\mu\text{g}/\text{m}^3$ an

Display löschen

TIPP

Wählt Serial 1, wenn ihr den Sensor an Port 1 angeschlossen habt, oder Serial 2, wenn ihr Port 2 nutzt.



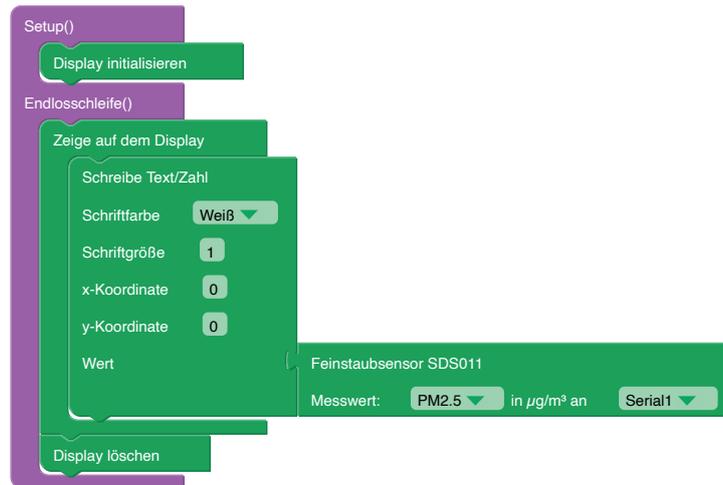
Aufgabe - Messeinheit Luft bestimmen

Findet heraus, wie fein der Staub ist, den der Sensor messen kann.



TIPP

Vergleicht dazu die Größe eines menschlichen Haars.
(Dafür könnt ihr auch online recherchieren.)





Lösung - Messeinheit Luft bestimmen

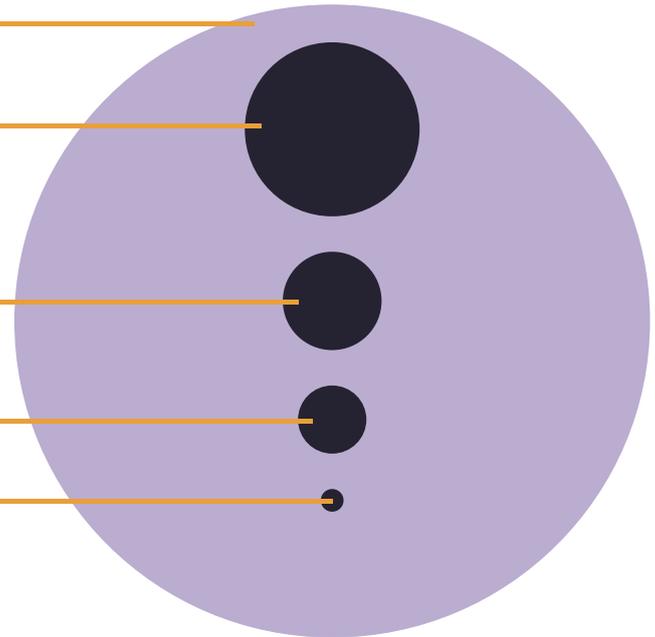
menschliches Haar — 150 μm

sichtbarer Staub — 30 μm

Mehl — 10 μm

Ölnebel — 3 μm

Ölrauch — 0.5 μm





Aufgabe - Lichtintensität

Schließt den Helligkeitssensor an und lasst euch den Messwert auf dem Display anzeigen.



TIPP

Die Umweltsensoren werden an einem der I2C/Wire Ports angeschlossen.

Benutzt dafür diese Befehle:

Display löschen

Display initialisieren

Zeige auf dem Display

Schreibe Text/Zahl

Schriftfarbe

Schriftgröße

x-Koordinate

y-Koordinate

Wert

Ultraschall-Abstandssensor an Port

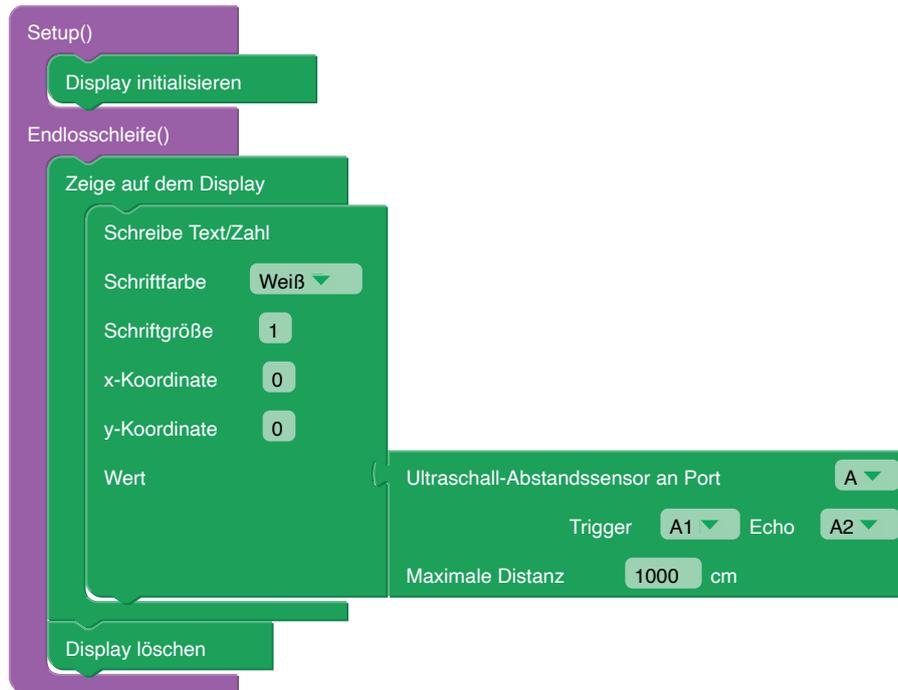
Trigger Echo

Maximale Distanz cm





Lösung - Lichtintensität





Aufgabe - Messeinheit Licht bestimmen

Wie hell leuchtet bzw. wieviel Lux erzeugt eine Taschenlampe?



EXTRA

Findet heraus, wieviel Lux an einem sonnigen oder bewölkten Tag erreicht werden.





Lösung - Messeinheit Licht bestimmen

Die Beleuchtungsstärke einer Handytaschenlampe liegt zwischen 2000 – 3000 Lux.

Ein wolkenfreier Sommertag erreicht bis zu 100.000 Lux. Ein bewölkter Himmel bei gleichem Sonnenstand mitunter nur noch 20.000 Lux.

Im Winter liegt die Beleuchtungsstärke an einem bewölkten Tag dagegen nur noch bei etwa 3.500 Lux.



Abstand messen

Ultraschall-Distanzsensor

Mit dem **Ultraschall-Distanzsensor** könnt ihr Entfernungen zu Objekten messen. Schließt den Sensor über das Breadboard an die senseBox Platine an. Nutzt dazu eines der Verbindungskabel mit vier Pins am Ende.

Die Pins müssen folgendermaßen verbunden werden:

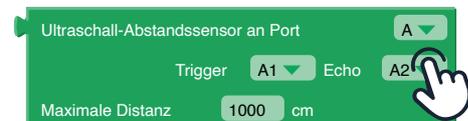
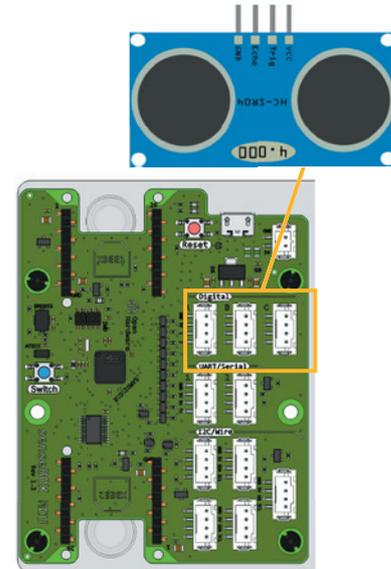
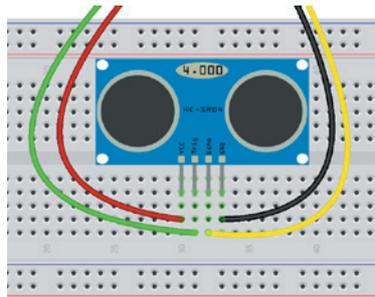
GND – GND (schwarzes Kabel)

Echo – 2 (gelbes Kabel)

Trig – 1 (grünes Kabel)

VCC – 5V (rotes Kabel)

Beachtet: Wenn ihr den Sensor an einen anderen Port anschließt, ändert sich auch die Belegung für Trigger und Echo (z.B. Pin D3 und D4 anstatt D1 und D2).





Aufgabe - Abstand messen

Schließt den Distanz-Sensor an und lasst den Messwert auf dem Display anzeigen.

Benutzt dafür diese Befehle:

Display löschen

Display initialisieren

Zeige auf dem Display

Schreibe Text/Zahl

Schriftfarbe

Schriftgröße

x-Koordinate

y-Koordinate

Wert

Ultraschall-Abstandssensor an Port

Trigger

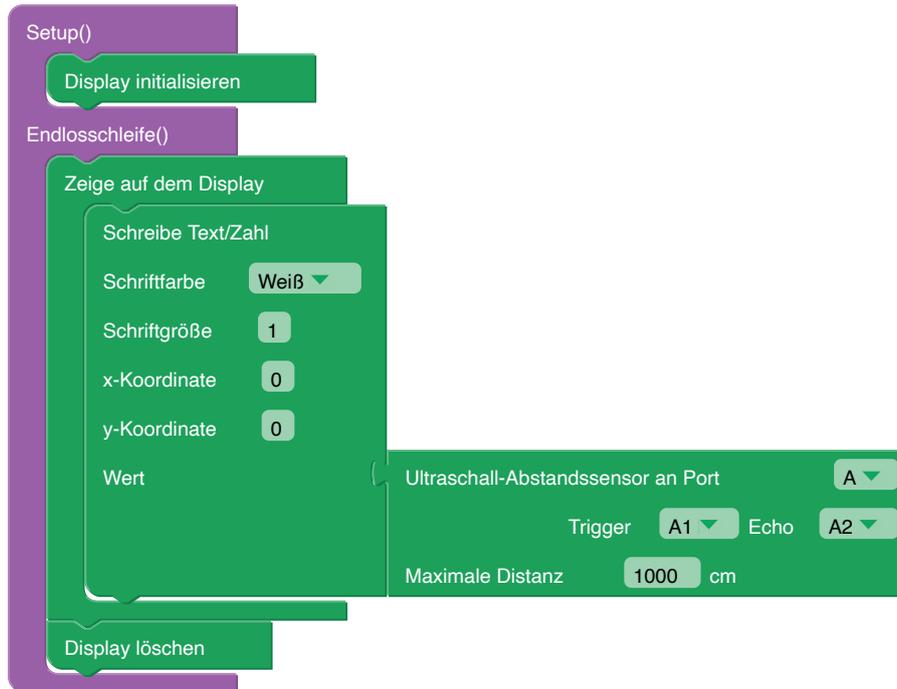
Echo

Maximale Distanz cm





Lösung - Abstand messen



Variablen

Mit Platzhaltern arbeiten

Variablen sind eine Art Container, denen man einen Namen gibt und denen man verschiedene Daten (zum Beispiel: Zahlen / Buchstaben) zuweist. Diese Platzhalter und ihre Daten kann man dann immer wieder abrufen, wenn man sie braucht.

int Element ▾ set int Element ▾ to

Variablen können ihren Wert im Verlauf eines Programms verändern. So könnt ihr zum Beispiel der Variable „Temperatur“ immer die aktuell gemessene Temperatur zuweisen.

set int Temperatur ▾ to Temperatur-/Luftfeuchtigkeitssensor (HDC1080)
Messwert: Temperatur in °C ▾

In der Informatik werden Variablen für viele verschiedene Dinge (Punktezähler / Zeitangaben / Berechnungen) genutzt.



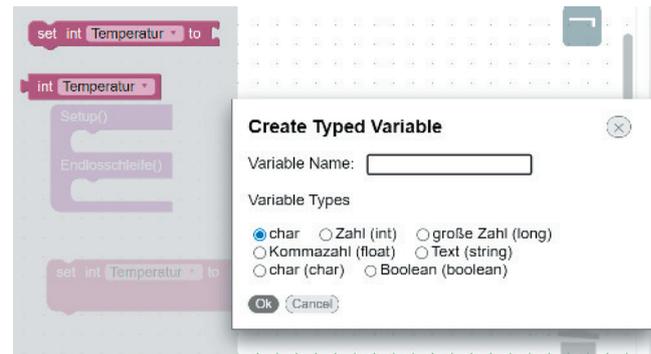
Variablen

Mit Platzhaltern arbeiten

Datentypen:

Je nachdem, was ihr in einer Variable speichern möchtet, müsst ihr den richtigen Datentyp auswählen.

- **char**: Für einzelne Textzeichen (Buchstaben).
- **string**: Für komplette Wörter oder Sätze.
- **int**: Für Zahlen zwischen - 32768 und 32768.
- **long**: Für große Zahlen zwischen - 2147483648 und 2147483647.
- **float**: Für Kommazahlen (zum Beispiel 25,56).
- **boolean**: Speichert nur zwei Zustände: Wahr oder falsch.

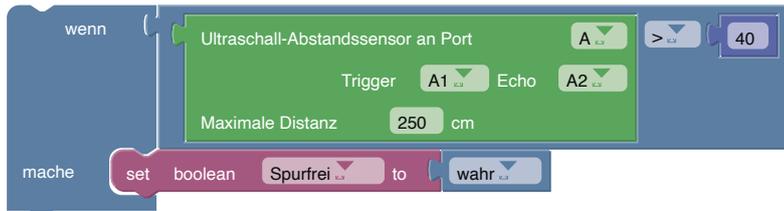


Wenn - Dann Bedingungen aufstellen

Übersicht Befehlsblöcke

Die „Wenn-Dann Bedingung“ ist beim Programmieren eine der wichtigsten Kontrollstrukturen, die ihr kennenlernen werdet. Mithilfe der “Wenn-Dann Bedingung“ kann die senseBox bestimmte Aktionen ausführen, wenn etwas bestimmtes passiert ist.

Wenn die gemessene Temperatur über 20 Grad beträgt, dann soll die eingebaute LED aufleuchten.

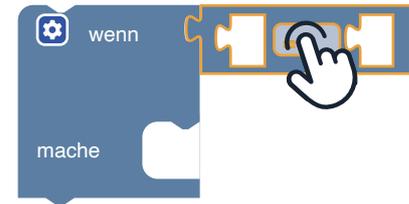


Mit dem Block hinter „wenn“ könnt ihr zwei Sachen miteinander vergleichen. Dafür braucht ihr **Operatoren**. Was das ist, erfahrt ihr auf der Rückseite.

Wenn - Dann Bedingungen aufstellen

Übersicht Operatoren

Operatoren werden beim Programmieren häufig verwendet. Damit können Bedingungen überprüft oder auch Werte verglichen werden. Ihr könnt zwischen folgenden Operatoren wählen.



$$A = B$$

Bedingung: Zwei Werte sind gleich groß.

$$A \neq B$$

Bedingung: Zwei Werte sind NICHT gleich groß.

$$A < B$$

Bedingung: Ein Wert "A" ist kleiner als der andere "B".

$$A \leq B$$

Bedingung: Ein Wert "A" ist kleiner als der andere "B" oder gleich groß.

$$A > B$$

Bedingung: Ein Wert "A" ist größer als der andere "B".

$$A \geq B$$

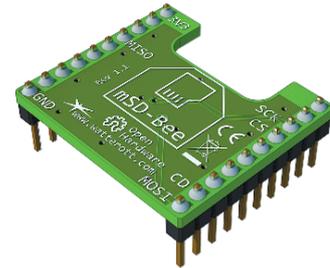
Bedingung: Ein Wert "A" ist größer als der andere "B" oder gleich groß.



Aufgabe - Daten speichern (SD-Karte)

1. Verbindet das SD-Bee Modul mit dem Xbee-Steckplatz 2 auf der senseBox Platine.
2. Erstellt in der `Setup()`-Schleife eine Datei. Den Block hierzu findet ihr in der Kategorie senseBox Output – SD
3. Um Daten auf die SD-Karte zu speichern muss jedes Mal die Datei mit dem „Öffne-Datei auf SD- Karte“-Block geöffnet werden
4. Danach kann mit dem „Schreibe Daten auf SD-Karte“-Block eine Variable in die Datei geschrieben werden.

Der „Öffne-Datei auf SD-Karte“-Block schließt nach dem Schreiben die Datei automatisch.





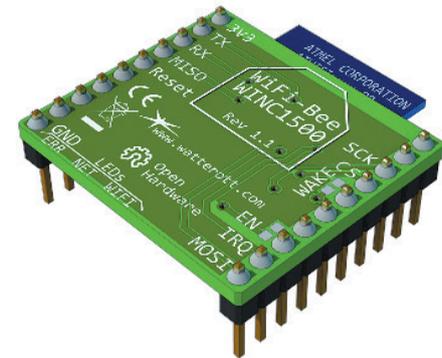
Lösung - Daten speichern (SD-Karte)

```
Setup()
  Erstelle Datei auf SD-Karte Daten Werte . txt
Endlosschleife()
  set int Temperatur to Temperatur-/Luftfeuchtigkeitssensor (HDC1080)
  Messwert: Temperatur in °C
  Öffne eine Datei auf der SD-Karte Data . txt
  Schreibe Daten auf SD-Karte int Temperatur Zeilenumbruch ✓
```



Aufgabe - Daten übertragen (Openseensemap)

1. Verbindet das WiFi-Bee Modul mit dem XBEE1-Steckplatz auf der senseBox Platine.
2. Zieht den Wifi-Block ins Setup und gebt euren Netzwerknamen (SSID) und das Passwort ein.
3. Registriert eure senseBox unter: <https://opensensemap.org/register> auf der openSenseMap. Ihr erhaltet dann eine BoxID und für jeden Sensor eine SensorID.
4. Tragt nun die BoxID in den „verbinde mit openSenseMap Block“ und die SensorID in den „sende Messwert an die openSenseMap“- Block ein.
5. Unter „Messintervall“ legt ihr fest, wie häufig gemessen werden soll.





Lösung - Daten übertragen (Openseensemap)

```
Setup()
  Erstelle Datei auf SD-Karte Daten Werte . txt
Endlosschleife()
  set int Temperatur to Temperatur-/Luftfeuchtigkeitssensor (HDC1080)
  Messwert: Temperatur in °C
  Intervall: Interval 60000 ms
  Öffne eine Datei auf der SD-Karte Data . txt
  Schreibe Daten auf SD-Karte int Temperatur Zeilenbruch ✓
```



Aufgabe - Langzeitmessung

Messt die Temperatur in eurer Umgebung über einen längeren Zeitraum und speichert die Daten auf der SD Karte.



TIPP

Nutzt geeignete Messintervalle, um nicht von Daten überhäuft zu werden.

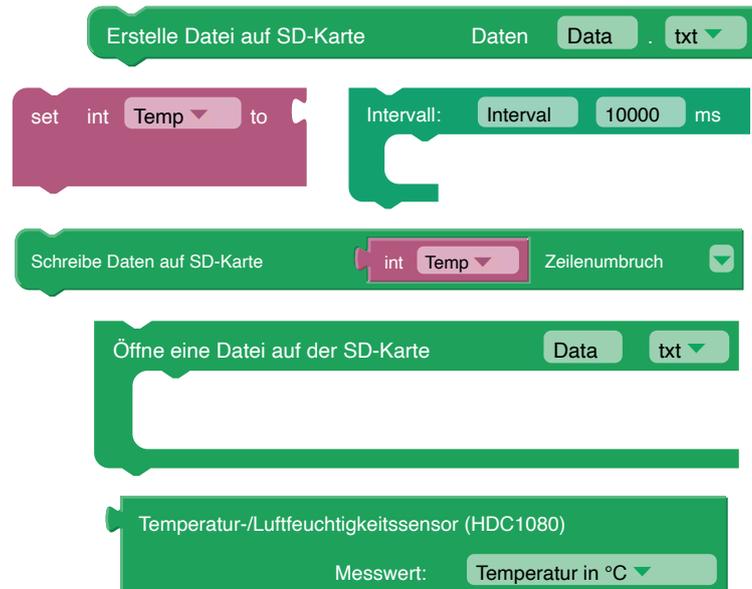
1 Minute = 60000 Millisekunden.



EXTRA

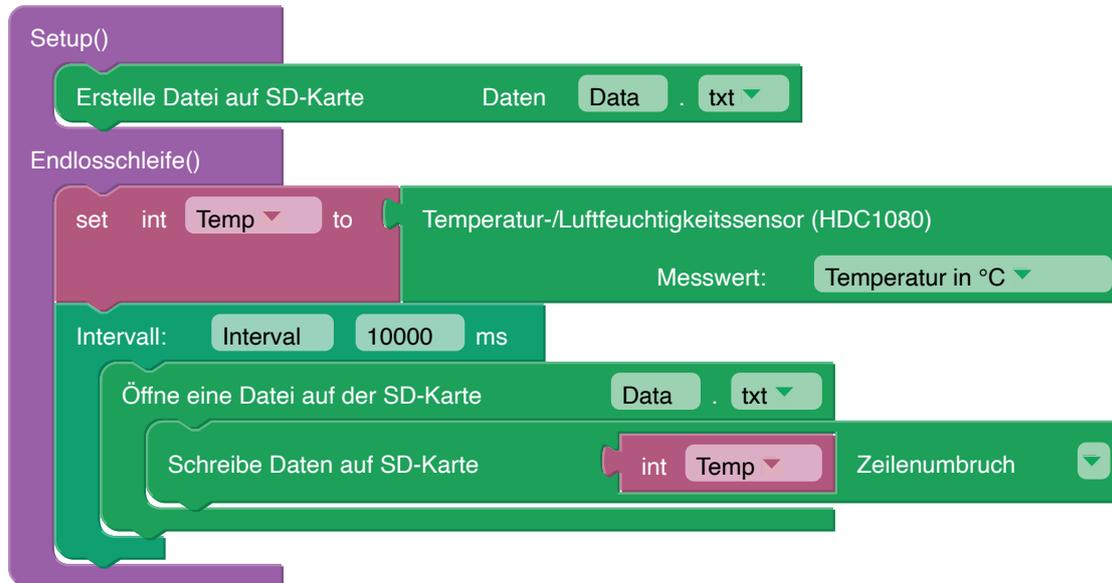
Schaut euch die Messwerte auf der SD Karte auf eurem Computer an. Wie haben sich eure Werte über die Zeit verändert?

Benutzt dafür diese Befehle:





Lösung - Langzeitmessung





Aufgabe - Vernetzte Umweltstation

Baut eure eigene vernetzte Umweltstation, die ihre Messwerte auf dem Display anzeigt und über WLAN ins Internet auf die openSenseMap überträgt.

Misst zum Beispiel Temperatur, Luftdruck, Luftfeuchtigkeit und Feinstaubbelastung auf eurem Balkon oder am Fenster.



TIPP

Gibt es kein offenes WLAN und ihr nutzt euer Handy als Hotspot, achtet auf euren Datenverbrauch.

Benutzt dafür diese Befehle:

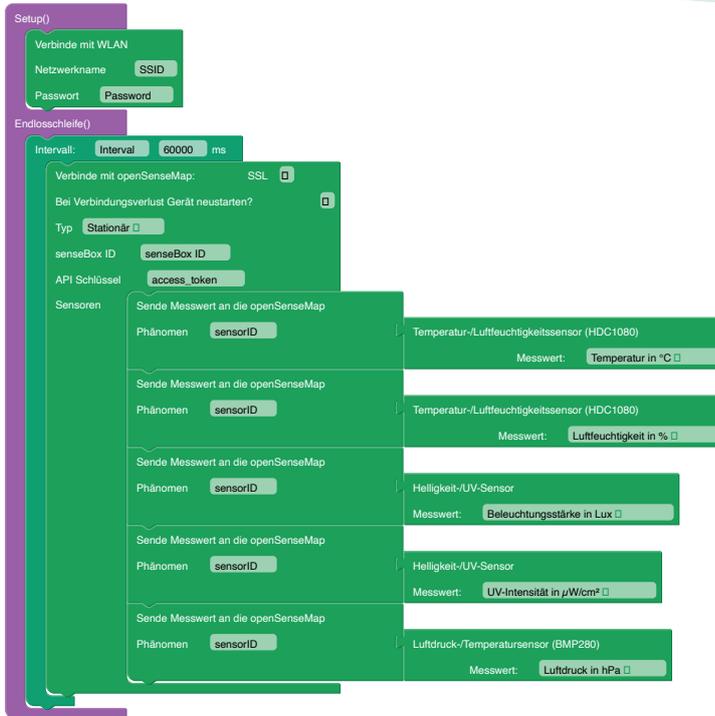
The image shows several code blocks for configuring a networked environmental station:

- Temperatur-/Luftfeuchtigkeitssensor (HDC1080)**: Messwert: Temperatur in °C
- Luftdruck-/Temperatursensor (BMP280)**: Messwert: Luftdruck in hPa
- Helligkeit-/UV-Sensor**: Messwert: Beleuchtungsstärke in Lux
- Sende Messwert an die openSenseMap**: Phänomen: sensorID
- Intervall:** Interval: 10000 ms
- Verbinde mit WLAN**: Netzwerkname: SSID, Passwort: Password
- Verbinde mit openSenseMap:** SSL (dropdown), Bei Verbindungsverlust Gerät neustarten? (dropdown), Typ: Stationär (dropdown), senseBox ID: senseBox ID, API Schlüssel: access_token, Sensoren: (empty list)





Lösung - Vernetzte Umweltstation



TIPP

Jeder Sensor braucht eine `sensorID`. Tragt sie ein, um sie bei der Registrierung eurer `senseBox` anzeigen zu lassen.



Aufgabe - Verkehrszähler

Für den Bau eines Fußgängerüberweges soll das Verkehrsaufkommen gemessen werden. Baut und programmiert einen automatischen Verkehrszähler mit Hilfe des Ultraschall-Distanzsensors.

Ihr braucht eine kleine Modellstraße, mit der ihr die Spurbreite festlegen könnt.

Wie werden vorbeifahrende Autos erkannt?



TIPP

Um zu verhindern, dass Autos doppelt gezählt werden, braucht ihr eine zweite Bedingung, die überprüft ob die Spur wieder frei ist.

Benutze dafür diese Befehl:

The image shows a sequence of Scratch code blocks:

- Display initialisieren** (green)
- set boolean** block with **Spurfrei** selected in the dropdown and **to** on the right.
- wenn** block (blue) containing:
 - wahr** dropdown
 - =** comparison operator
 - Ultraschall-Abstandssensor an Port** block with **A1** selected in the **Trigger** dropdown, **A2** selected in the **Echo** dropdown, and **250** cm in the **Maximale Distanz** field.
- mache** block (blue)
- Zeige auf dem Display** block (green)
- int** dropdown block with **Fahrzeuge** selected.
- Schreibe Text/Zahl** block (green) with:
 - Schriftfarbe** dropdown set to **Weiß**
 - Schriftgröße** set to **1**
 - x-Koordinate** set to **0**
 - y-Koordinate** set to **0**
 - Wert** field (empty)





Lösung - Verkehrszähler

Setup()

- Display initialisieren

Endlosschleife()

- wenn
 - Ultraschall-Abstandssensor an Port
 - Trigger: A1, Echo: A2
 - Maximale Distanz: 250 cm
 - Werte: A > 40
- mache
 - set boolean Spurfrei to wahr
- wenn
 - boolean Spurfrei = wahr und
 - Ultraschall-Abstandssensor an Port
 - Trigger: A1, Echo: A2
 - Maximale Distanz: 250 cm
 - Werte: A > 40
- mache
 - erhöhe Fahrzeuge um 1
 - set boolean Spurfrei to falsch

Zeige auf dem Display

- Schreibe Text/Zahl
 - Schriftfarbe: Weiß
 - Schriftgröße: 1
 - x-Koordinate: 0
 - y-Koordinate: 0
 - Wert: int Fahrzeuge



TIPP

Denkt daran, die Distanz an die Spurbreite eures Experimentes anzupassen. Am besten funktioniert das durch Ausprobieren.



</> Programmierung

Geschafft!

Ihr habt nun die senseBox kennengelernt und seid Sensor-Grundlagenexpert*innen. Es gibt zahlreiche Möglichkeiten mit senseBox kreative Projekte umzusetzen.

Für mehr Inspiration empfehlen wir diese Seiten:

- <https://senseBox.de/>
- tueftellab.de

Und nicht vergessen, es wird immer mal wieder neue Herausforderungen geben. Manchmal funktioniert nicht immer alles auf Anhieb und das ist vollkommen normal.



Wir freuen uns über euer Feedback zu unserem Material!
post@junge-tueftler.de / Mehr Materialien unter: tueftellab.de

