

# senseBox MCU – Gesamtübersicht der Anschlüsse

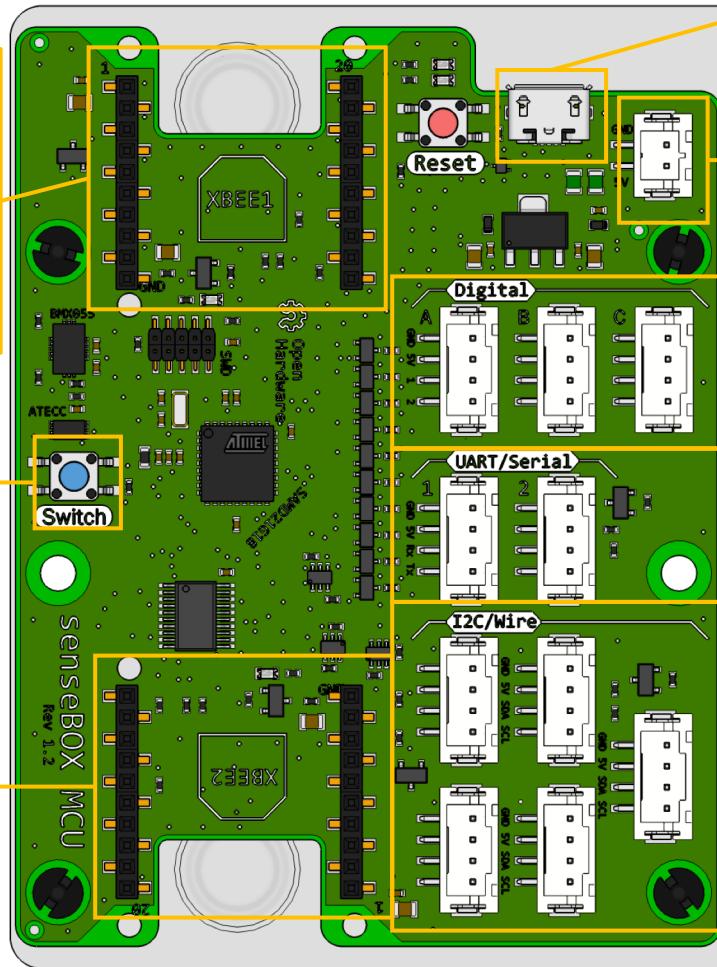
## XBEE-Steckplatz 1

XBEEs sind kleine Zusatzmodule um die senseBox um Funktionen wie WiFi oder eine SD-Karte zu erweitern. Schließe hier das WLAN-Modul an

## Button

## XBEE-Steckplatz 2

XBEEs sind kleine Zusatzmodule um die senseBox um Funktionen wie WiFi oder eine SD-Karte zu erweitern. Schließe hier das SD-Modul an



## USB-Anschluss

## Akku-Anschluss

## Digital/Analog-Anschlüsse

Hier werden Sensoren und Aktoren über das Breadboard angeschlossen.

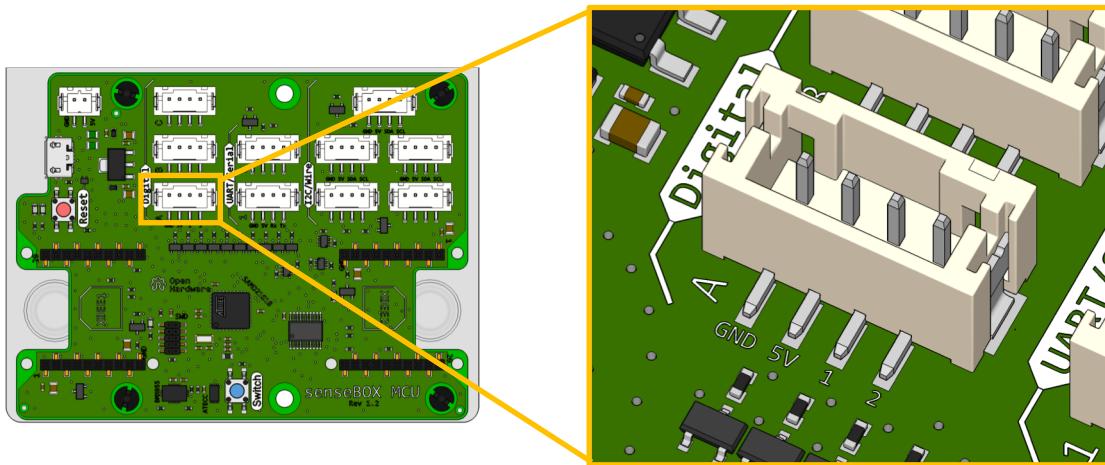
## UART/Serial-Anschlüsse

Hier wird der Feinstaubsensor angeschlossen

## I2C/Wire-Stecker

Hier werden alle Umweltsensoren sowie das Display angeschlossen.

# Die senseBox und JST-Adapterkabel



Jeder **Digital/Analog-Stecker** auf der senseBox MCU verfügt über vier verschiedene Pins.

Der **GND-Pin** ist der Minuspol und ist immer mit dem schwarzen Kabel verbunden.

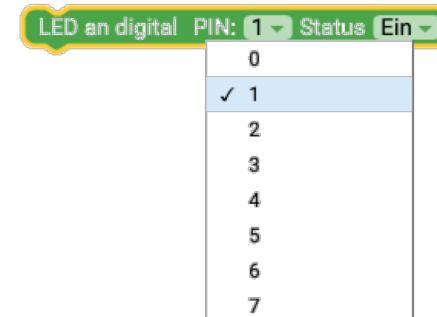
Der **5V-Pin** dient zur Stromversorgung der Sensoren und hat immer das rote Kabel.

Die mit **1** und **2** beschrifteten Pins sind die digitalen bzw. analogen Pins 1 und 2. Du wirst sehen, dass diese Nummerierung fortlaufend bis zum Pin 6 an Port Digital C zählt.

## Pins in den Blöcken einstellen

Damit deine eigenen Programme richtig funktionieren können, musst du in einigen Blöcken den Pin auswählen an dem dein Verbraucher (also z.B. eine LED) angeschlossen ist.

Für eine LED an Digital 1 würde der Block also wie folgt aussehen:

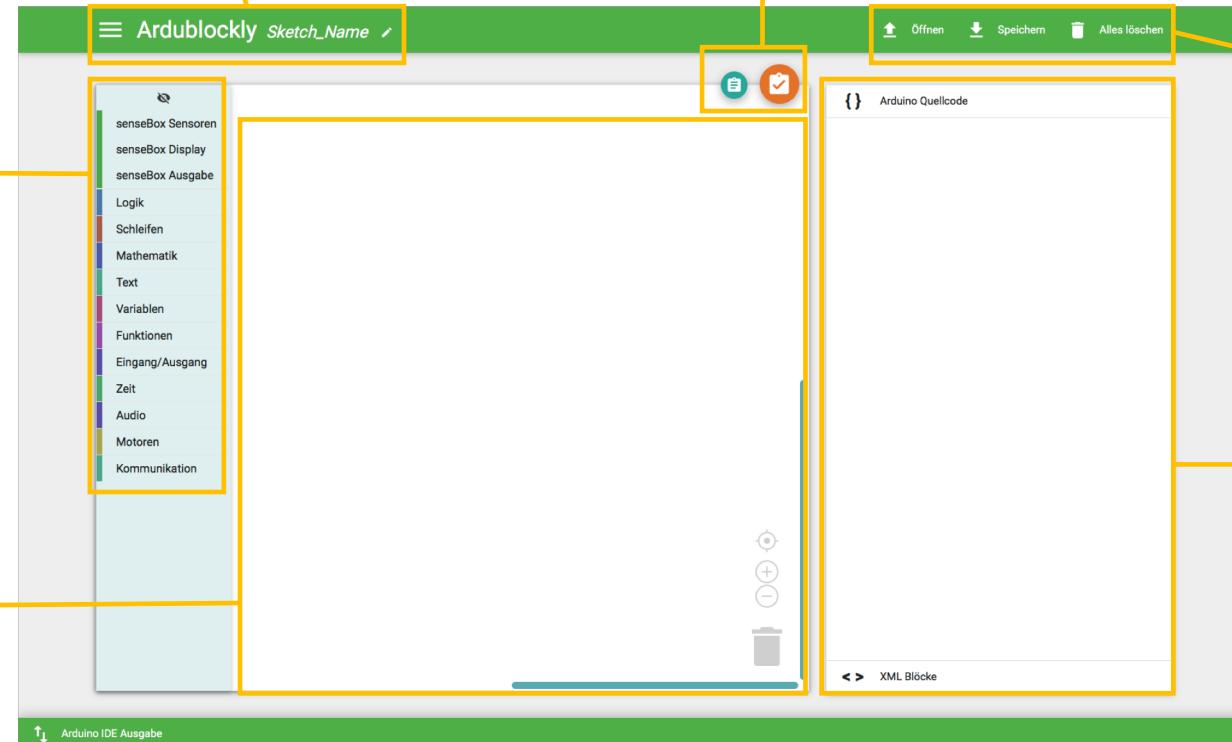






# Die Programmieroberfläche

Hier kannst du zum einen auf das Einstellungsmenü von Blockly für senseBox zugreifen sowie den Namen deines Programmes (Sketches) ändern.



Hinter diesen Schaltflächen verbergen sich alle Blöcke die du zum Programmieren der senseBox benötigst.

Dies ist dein Arbeitsbereich. Hier setzt du dein Programm aus Blöcken zusammen



Mit dieser Schaltfläche kannst du dein Programm komplizieren (in Maschinensprache umwandeln) sowie herunterladen um es auf deine senseBox-MCU zu übertragen.



Diese Schaltfläche kopiert deinen Programmcode in die Zwischenablage.



Mit diesen Schaltflächen kannst du Projekte öffnen, speichern sowie dein aktuelles Projekt löschen.

In diesem Fenster wird dir dein Programmquellcode.

# Variablen – Platzhalter

**Variablen** oder auch Platzhalter genannt werden in der Informatik für ganz viele verschiedene Dinge genutzt. Sie sind eine Art Box, die mit einem Name versehen ist, in diese Box kannst du nun verschiedene Dinge hinterlegen, z.B. Zahlen oder auch Texte, und diese später erneut abrufen.



## Variablen – Datentypen

Je nachdem, was du in einer Variable speichern möchtest musst du den richtigen Datentyp auswählen.

**Zeichen:** Für einzelne Textzeichen

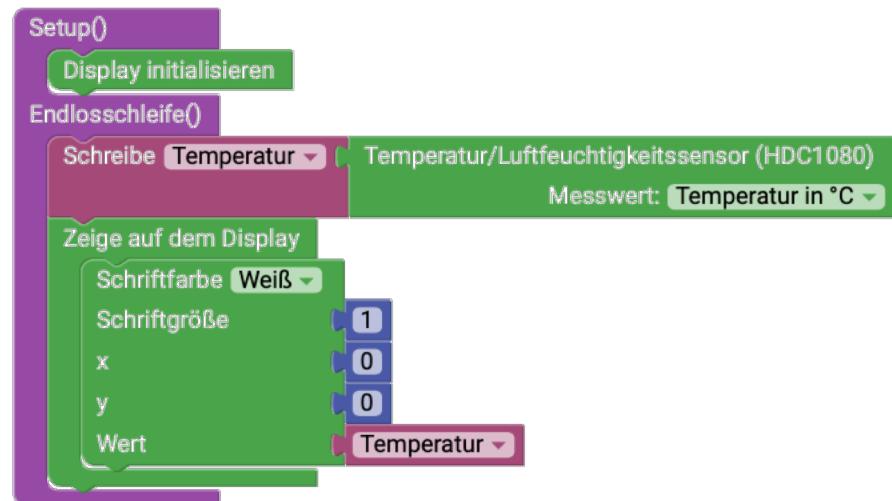
**Text:** Für komplette Sätze oder Wörter

**Zahl:** Für Zahlen zwischen -32768 und 32768

**Große Zahl:** Für große Zahlen zwischen -2147483648 und 2147483647

**Dezimalzahl:** Für Kommazahlen (z.B. 25,56)

**Variablen** können ihren Wert im Laufe des Programms verändern, so dass du zum Beispiel der Variable „Temperatur“ immer die aktuell gemessene Temperatur zuweist.







# Wenn-Dann-was?

Die Wenn-Dann Bedingung ist beim Programmieren eine der wichtigsten Kontrollstrukturen, die du kennenzulernen wirst.  
Mithilfe des Wenn-Dann Befehl kann die senseBox bestimmte Aktionen ausführen, wenn etwas bestimmtes (z.B. ein Knopf gedrückt wurde) passiert ist.



Hier siehst du den Programmierbefehl zur Wenn-Dann-Bedingung. Damit kannst du programmieren, was genau gemacht werden soll wenn eine Bedingung erfüllt ist.



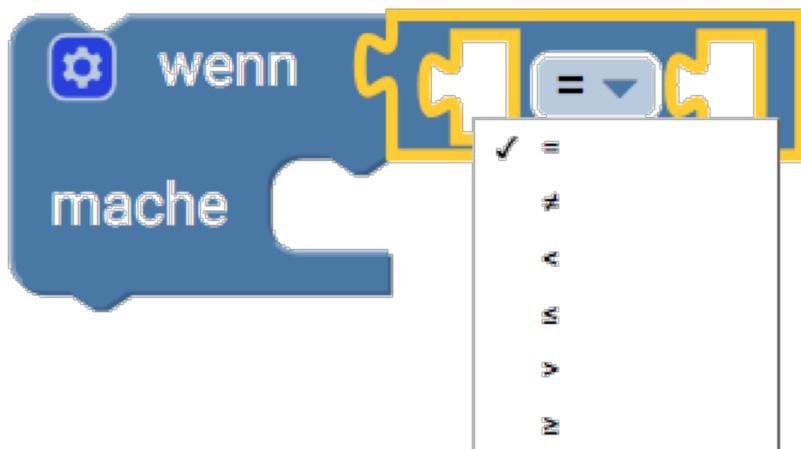
Mit diesem Block kannst du immer zwei Sachen miteinander vergleichen. Eine Erklärung für die verschiedenen Symbole findest du auf der Karte **Operatoren**.



Beispiel: **Wenn** die Temperatur größer als 20°C ist, **dann** wird die eingebaute LED angeschaltet.

# Operatoren

Operatoren werden in vielen Situationen beim Programmieren benötigt. Mithilfe der Operatoren können Bedingungen überprüft oder auch Werte verglichen werden.



Die folgenden Operatoren finden sich unter [Logik](#):

- =-Zeichen:** Hiermit kannst du die senseBox einen Programmierbefehl ausführen lassen, wenn zwei Werte gleich groß sind.
- Zeichen:** Hiermit kannst du die senseBox einen Programmierbefehl ausführen lassen, wenn zwei Werte unterschiedlich groß sind.
- <-Zeichen:** Mithilfe dieses Zeichens kannst du zwei Werte vergleichen lassen. Wenn die Spitze des Symbols auf den kleineren Wert zeigt, führt die senseBox den nächsten Befehl aus.
- <= -Zeichen:** Dieses Zeichen ist eine Erweiterung des »kleiner«-Zeichens ( $\leq$ ) und schließt auch Werte ein, die kleiner und gleich groß sind ein.
- >-Zeichen:** Mithilfe dieses Zeichens kannst du den Calliope zwei Werte vergleichen lassen. Die Öffnung des Symbols zeigt diesmal auf den größeren Wert.
- >= -Zeichen:** Dieses Zeichen ist eine Erweiterung des »größer«-Zeichens ( $\geq$ ) und schließt auch Werte ein, die größer oder gleich groß sind.





# Umweltsensoren auslesen

## Luftdrucksensor

Licht Sichtbar + UV Sensor  
Messwert: Beleuchtungsstärke in Lux ▾

Dieser Block gibt dir den Messwert des Luftdrucksensors aus. Der Sensor kann neben dem Luftdruck auch die Temperatur messen. Den gewünschten Messwert kannst du im Dropdown Menü auswählen.



## Lichtsensor

Luftdruck/Temperatur Sensor (BMP280)  
Messwert: Luftdruck in Pa ▾

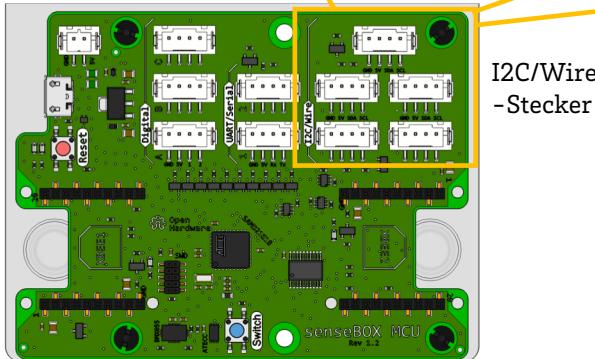
Dieser Block gibt dir den Messwert des Lichtsensors aus. Im Dropdown Menü kannst du den gewünschten Messwert auswählen. Der Sensor kann das sichtbare Licht (lux) und das UV-Licht messen.



## Temperatur- & Luftfeuchtigkeitssensor

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

Dieser Block gibt dir den Messwert des Temperatur- & Luftfeuchtigkeitssensors aus. Im Dropdown Menü kannst du den gewünschten Messwert auswählen.



I2C/Wire-Stecker

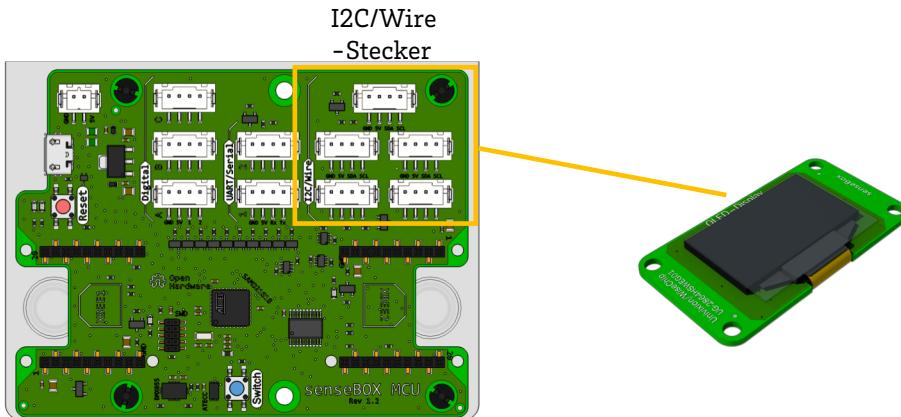
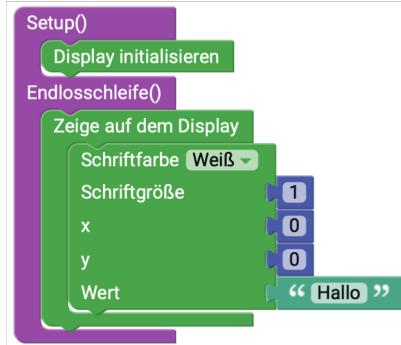
## Anschließen der Sensoren

Alle Umweltsensoren der senseBox werden über ein JST-Kabel an die I2C/Wire-Stecker angeschlossen. Mehr zu den JST-Kabeln findest du auf der Karte [Die senseBox und JST-Kabel](#)

# Das Display

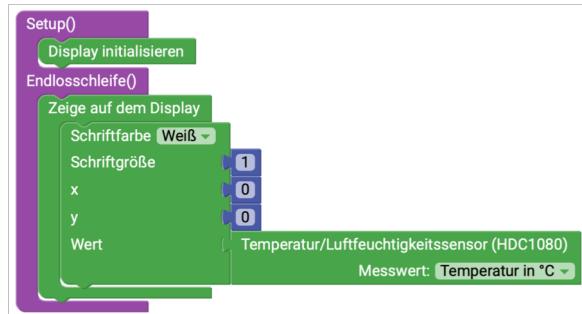
## Text anzeigen

Um das Display verwenden zu können muss es im *Setup()* initialisiert werden. Anschließend kannst du mit ein paar Blöcken in der *Endlosschleife()* Text oder auch Messwerte auf dem Display anzeigen lassen. Hier ein Beispiel für einen kurzen Text:

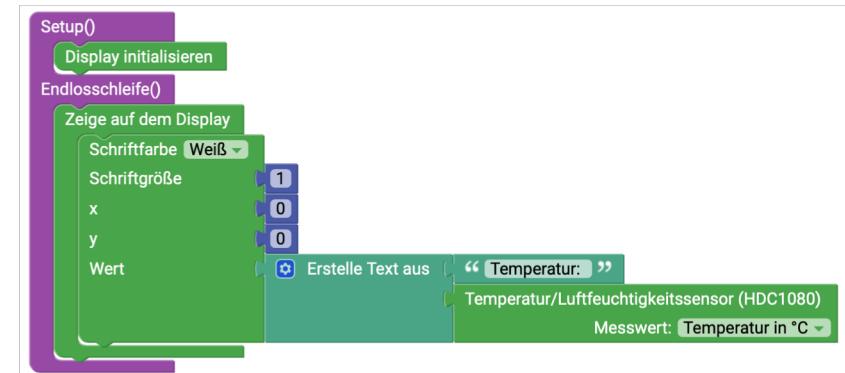


## Messwerte anzeigen und beschreiben

Um Messwerte auf dem Display anzeigen zu lassen verknüpfe entweder den Sensorblock direkt mit den Displayblöcken,



oder erstelle einen Text mit einer Beschreibung und dem Messwert.

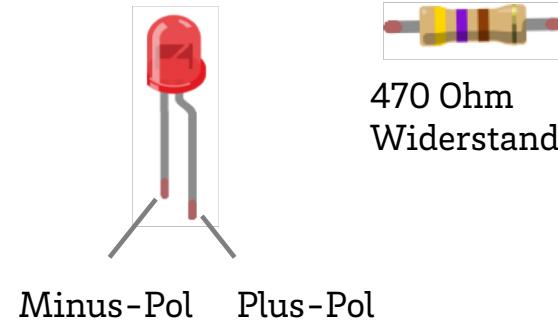
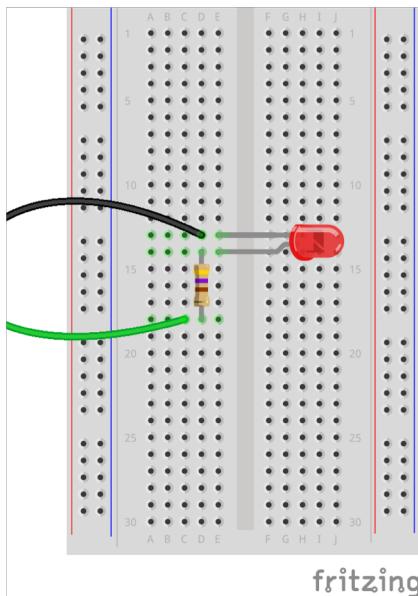








# LED – Die einfachste Ausgabemöglichkeit



Zum Anschließen einer LED benötigst du einen **Widerstand** (470 Ohm) und ein Adapterkabel. Das Adapterkabel steckst du in den **Digital A-** Stecker auf der senseBox-MCU. Danach verbindest du das **schwarze** und das **grüne** Kabel wie oben abgebildet.

Um die LED zum leuchten zu bringen benötigst du nur diesen einen Block:

LED an digital PIN: 1 Status Ein

**Aufgabe:** Baue die abgebildete Schaltung und bringe die LED zum leuchten.

LED an digital PIN: 1 Status Ein

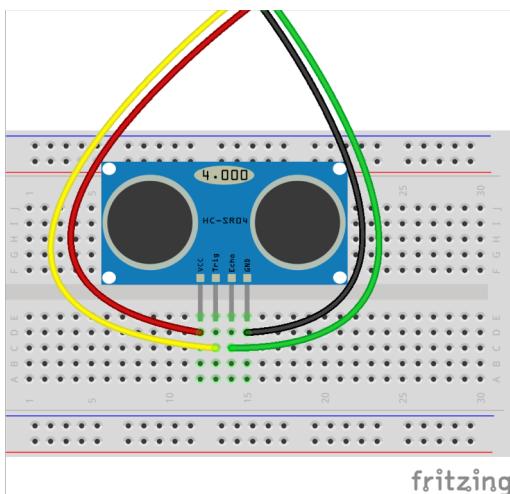
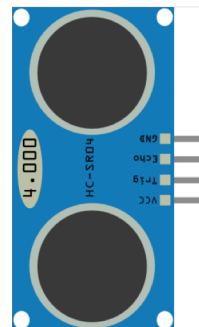
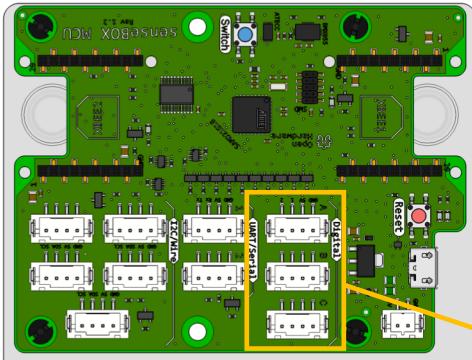
Warte 1000 Millisekunden

Warte 1000 Millisekunden

LED an digital PIN: 1 Status Aus



# Der digitale Zollstock



Zum Anschließen des Ultraschall-Distanzsensors benötigst du ein JST-Adapterkabel. Der Sensor selbst hat vier verschiedene Anschlüsse (Pins): VCC, Trig, Echo und GND. Diese vier Stecker müssen mit den vier Kabeln des JST-Adapterkabels verbunden werden.

GND	GND	(schwarzes Kabel)
Echo	2	(gelbes Kabel)
Trig	1	(grünes Kabel)
VCC	5V	(rotes Kabel)

**Beachte:** Wenn du den Sensor an einen anderen Port anschließt ändert sich auch die Belegung für Trigger und Echo.  
Mehr zur Funktionsweise der JST-Adapterkabel findest du auf der Karte [Die senseBox und JST-Adapterkabel](#).

Der Ultraschall-Distanzsensor hat, genau wie alle anderen Sensoren, einen eigenen Block in dem definiert wird wo und wie der Sensor angeschlossen ist und die gemessene Distanz in cm ausgibt.

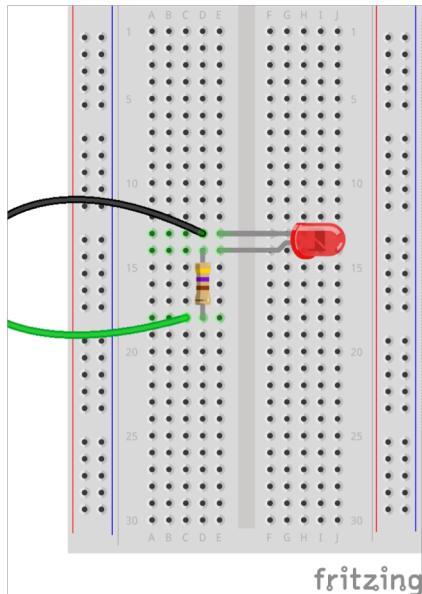
Ultraschall Abstandssensor an Port A  
Trigger 1 Echo 2

Wenn du dich nicht erinnern kannst wie die Sensoren ausgelesen und ihre Messwerte angezeigt werden schaue auf den Karten [Umweltsensoren auslesen](#) und [Das Display nach](#).



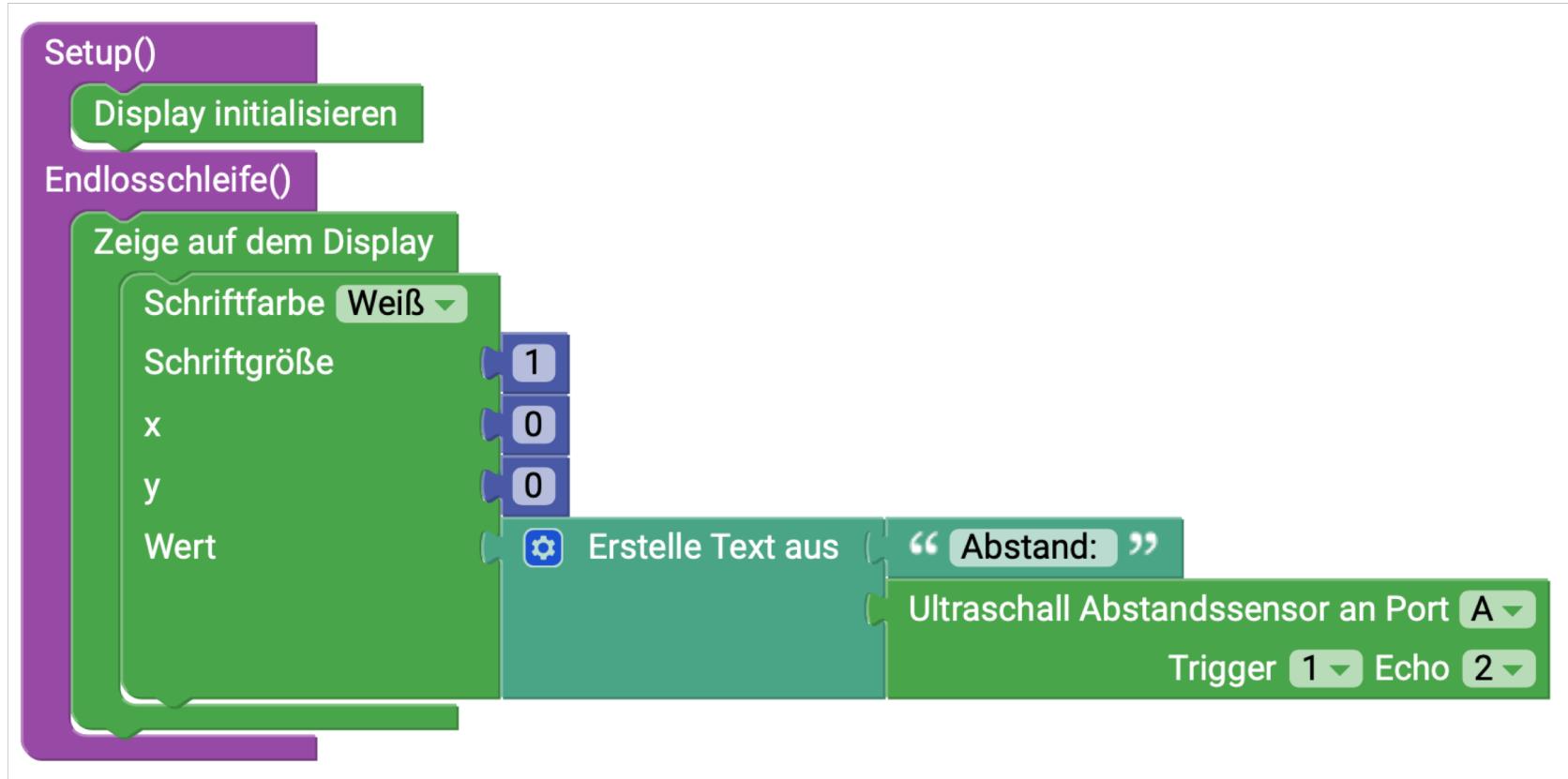
# LED – Die einfachste Ausgabemöglichkeit

## Lösungsvorschlag





# Der digitale Zollstock - Lösungsvorschlag



The Scratch script consists of two main sections: Setup() and Endlosschleife().

- Setup()**: Contains a single block: **Display initialisieren**.
- Endlosschleife()**:
  - Contains a **Zeige auf dem Display** block.
  - Contains a **Schriftfarbe Weiß** block.
  - Contains a **Schriftgröße 1** block.
  - Contains a **x 0** block.
  - Contains a **y 0** block.
  - Contains a **Wert** block.
  - Contains an **Erstelle Text aus** block with the value **“ Abstand: ”**.
  - Contains a **Ultraschall Abstandssensor an Port A** block.
  - Contains a **Trigger 1 Echo 2** block.



# Verkehrszähler

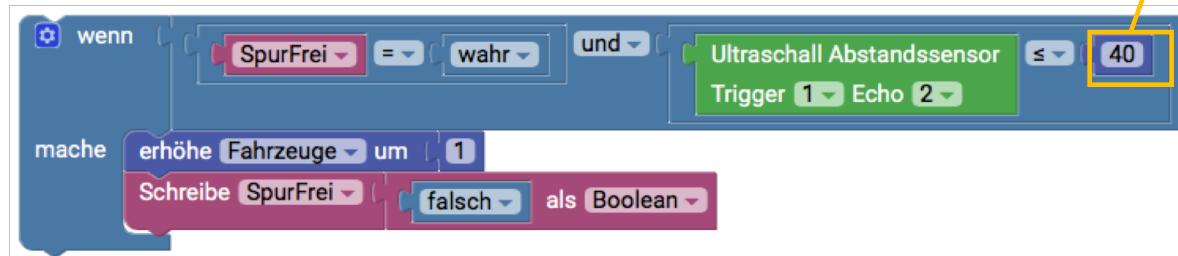
## Den Verkehr zählen

Um den Verkehr zu zählen muss ständig überprüft werden, ob eine bestimmte Distanz in cm (die Breite der Fahrspur) unterschritten wird. Außerdem muss verhindert werden, dass ein Auto mehrfach gezählt wird.



Denke daran, die Spurbreite an dein Experiment anzupassen. Am besten funktioniert dies durch Ausprobieren.

Mit den Blöcken oben wird überprüft ob die eingestellte Fahrspur frei ist. Dazu muss eine Maximaldistanz (das Ende der Fahrspur) festgelegt werden. Wenn diese überschritten wird wird die Variable „SpurFrei“ auf „wahr“ gesetzt.



Mit diesen Blöcken wird überprüft ob die Variable „SpurFrei“ wahr ist und ob die gemessene Distanz kleiner als die Maximaldistanz (Spurbreite) ist. Nur wenn beide Bedingungen wahr sind wird ein Fahrzeug gezählt und die Variable „SpurFrei“ auf „falsch“ gesetzt. Diese bleibt so lang auf „falsch“ bis in der ersten Bedingung erneut eine Distanz gemessen wurde die größer ist als die Maximaldistanz.

Vergiss nicht dir deinen Zähler auf dem Display anzeigen zu lassen!



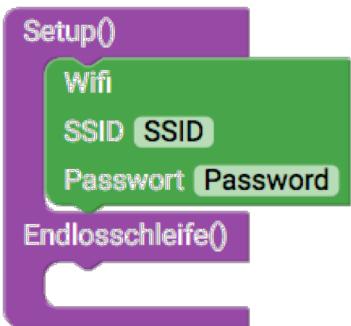
# Übertragen an die openSenseMap

## Wifi Verbindung herstellen

Verbinde das Wifi Bee mit dem XBee Steckplatz 1

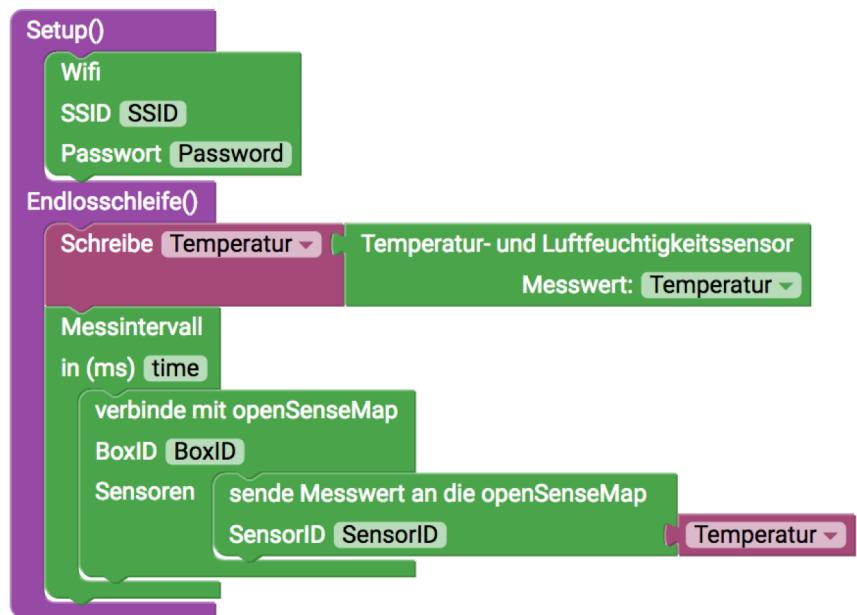


Anschließend musst du den Wifi-Block ins Setup ziehen und deinen Netzwerknamen (SSID) und das Passwort angeben.



## Senden an die openSenseMap

Nach der Registrierung deiner senseBox auf der openSenseMap erhältst du eine BoxID und für jeden Sensor eine SensorID. Trage die BoxID in den „verbinde mit openSenseMap“-Block ein und die SensorID in den „sende Messwert an die openSenseMap“-Block. Das Messintervall steuert die Zeit nach der ein Messwert übertragen wird

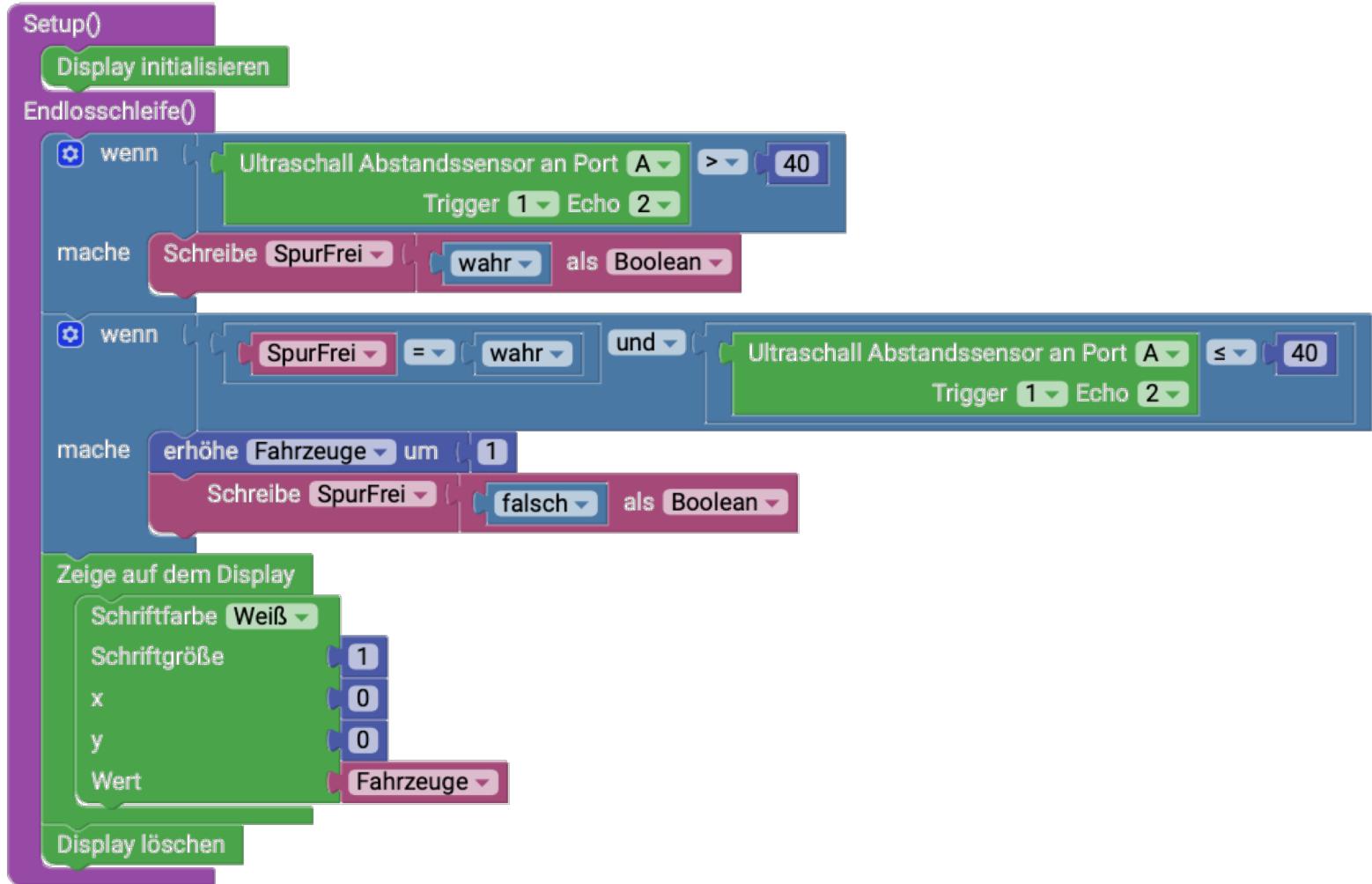


Auf der openSenseMap kannst du prüfen, ob Messwerte ins Internet übertragen werden.



# Verkehrszähler - Lösung

Denkt daran die Spurbreiten auf euer Experiment anzupassen!



The Scratch script consists of the following blocks:

- Setup()**:
  - Display initialisieren
- Endlosschleife()**:
  - wenn** Ultraschall Abstandssensor an Port A > 40  
Trigger 1 Echo 2
    - mache Schreibe SpurFrei wahr als Boolean
  - wenn** SpurFrei = wahr und Ultraschall Abstandssensor an Port A ≤ 40  
Trigger 1 Echo 2
    - mache erhöhe Fahrzeuge um 1
    - Schreibe SpurFrei falsch als Boolean
- Zeige auf dem Display**:
  - Schriftfarbe Weiß
  - Schriftgröße 1
  - x 0
  - y 0
  - Wert Fahrzeuge
- Display löschen**

