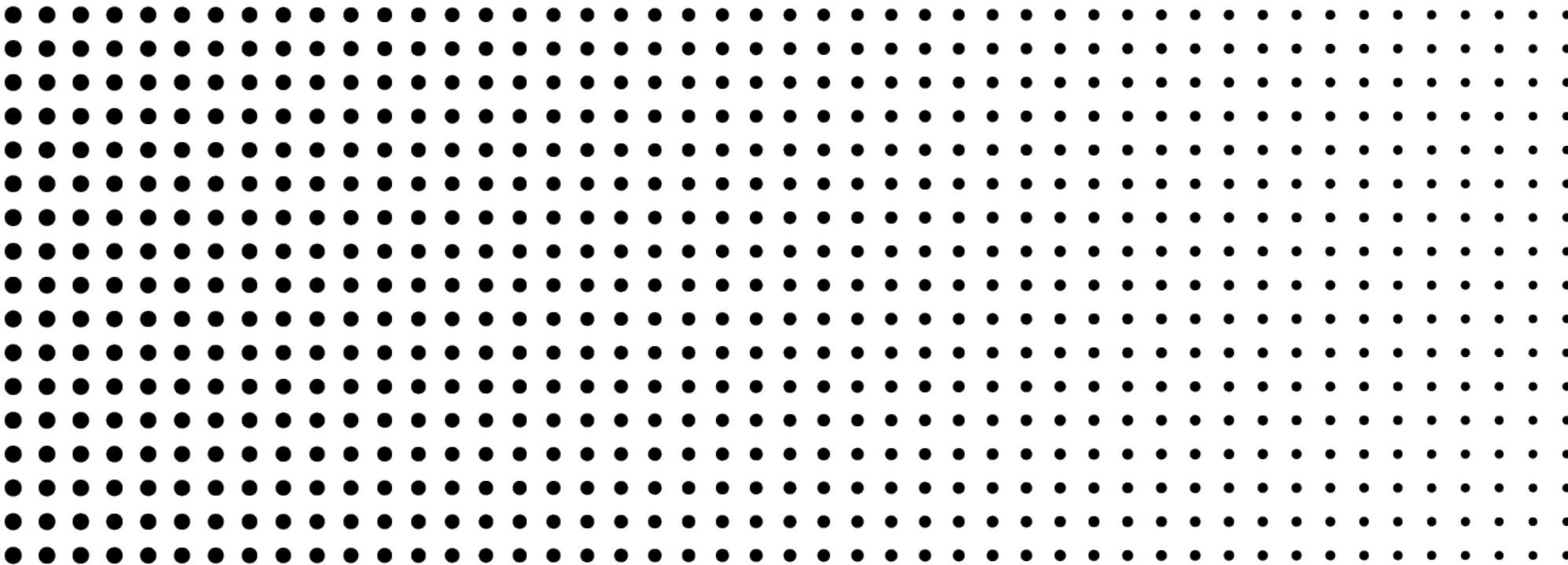


Kategorie	Nr.	Name
SB	01	senseBox MCU – Gesamtübersicht der Anschlüsse
SB	02	Die senseBox und JST-Adapterkabel
SB	03	Die Programmieroberfläche
SB	04	Umweltsensoren auslesen
SB	05	Feinstaubsensor auslesen
SB	06	Das Display
SB	07	Der Ultraschall-Distanzsensor
SB	08	Übertragen an die openSenseMap
SB	09	Speichern auf SD-Karte
GI	01	Variablen - Platzhalter
GI	02	Wenn-Dann-Was?
GI	03	Operatoren
A	01	Basis-Modul: Hello World
A	02	Modul 1: Zukunft der Stadt I
A	03	Modul 1: Zukunft der Stadt II
A	04	Modul 2: Zukunft der Energie I
A	05	Modul 2: Zukunft der Energie II
A	06	Modul 3: Zukunft der Gesundheit
L	01	Lösung: Hello World
L	02	Lösung Modul 1: Zukunft der Stadt I
L	03	Lösung Modul 2: Zukunft der Energie I
L	04	Lösung Modul 2: Zukunft der Energie II
L	05	Lösung Modul 3: Zukunft der Gesundheit

# FUTURIUM

senseBox 

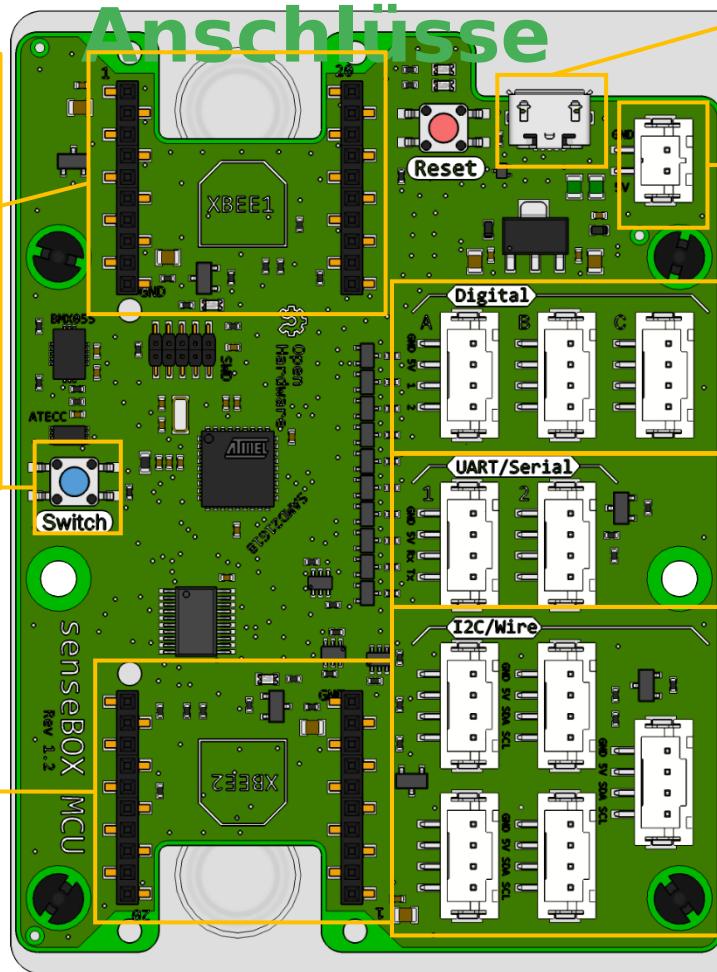


# senseBox MCU - Gesamtübersicht der

**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**

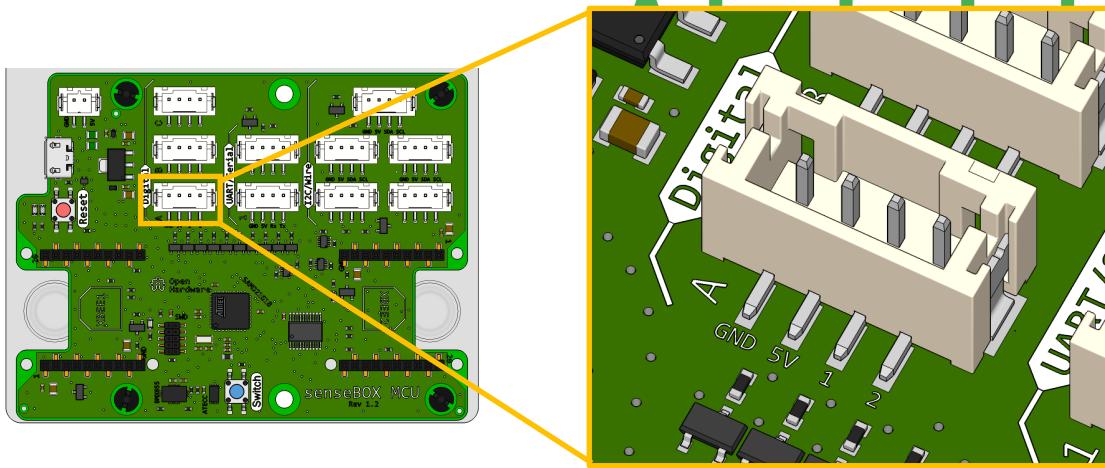
**Akku-Anschluss**

**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- Anschlusskabel



Jeder **Digital/Analog-Port** 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.

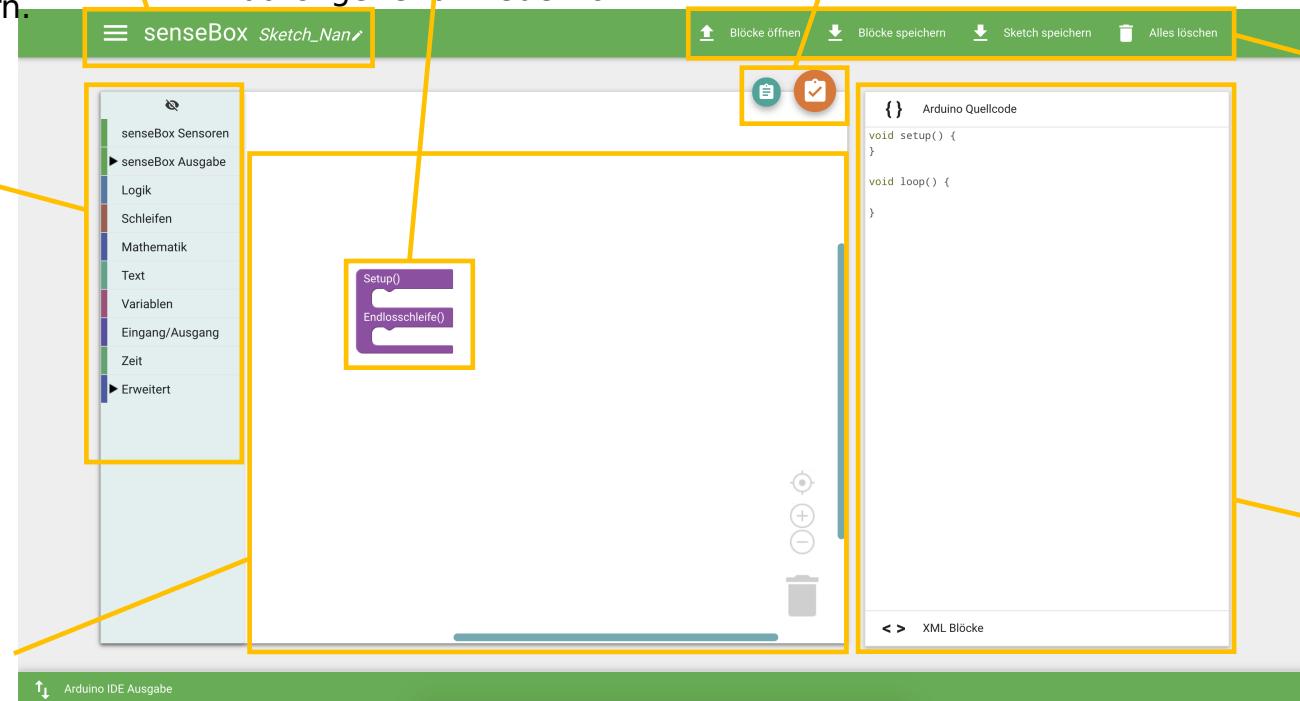
Für eine LED an Digital 1 würde der Block

LED an digital	PIN:	1	Status	Ein	Aus
		0			
		✓ 1			
		2			
		3			
		4			
		5			
		6			
		7			

# 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

Dieser Block wird automatisch beim Starten der Oberfläche geladen. Die Setup-Funktion wird einmalig zum Programmstart ausgeführt. Die Endlosschleife wird durchgehend wiederholt.

Mit dieser Schaltfläche kannst du dein Programm kompilieren (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

# 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 Dropdownmenü auswählen.



## Lichtsensor

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

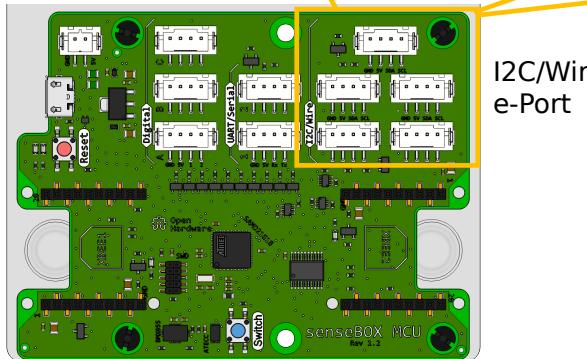
Dieser Block gibt dir den Messwert des Lichtsensors aus. Im Dropdown Menü kannst du den gewünschten Messwert auswählen. Der Sensor kann das sichtbare Licht (lux) und das UV-Licht



## Temperatur- & Luftfeuchtigkeitssenso

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.



**Anschließen der Sensoren**  
Alle Umweltsensoren der senseBox werden über ein JST-Kabel an die I2C/Wire-Port angeschlossen. Mehr zu den JST-Kabeln findest du auf der Karte **Die senseBox und JST-Kabel**

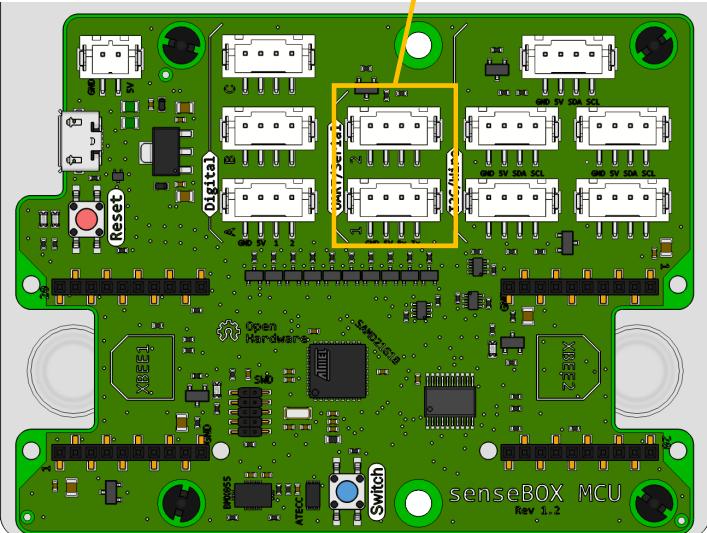
# Feinstaubsensor auslesen

## Anschluss des Sensors

Der Feinstaubsensor wird über das Feinstaubsensor - JST Kabel an einen der beiden UART/Serial Ports angeschlossen.



UART/Serial



## Feinstaubsensor auslesen

### Feinstaubsensor

Messwert: PM2.5 ▾ Serial1 ▾

Dieser Block gibt dir den Messwert des Feinstaubensors. Im Dropdown Menü kannst du den gewünschte Messwert auswählen.

Der Feinstaubsensor kann Feinstaub in zwei verschiedenen Partikelgrößen gemessen werden.

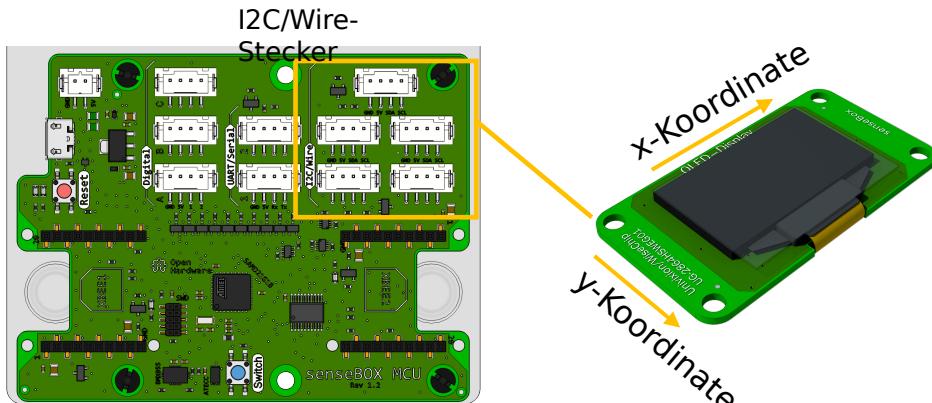
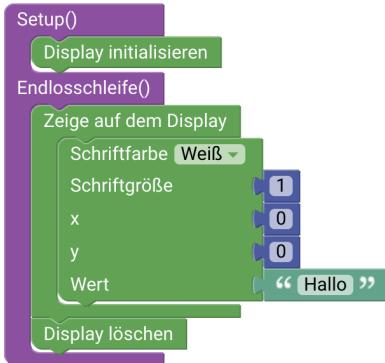
PM2.5: Gibt dir den Anzahl der Feinstaubpartikel  $<2,5 \mu\text{m}$  in  $\mu\text{g}/\text{m}^3$  an

PM10: Gibt dir den Anzahl der Feinstaubpartikel  $<10 \mu\text{m}$  in  $\mu\text{g}/\text{m}^3$  an

# Das Display

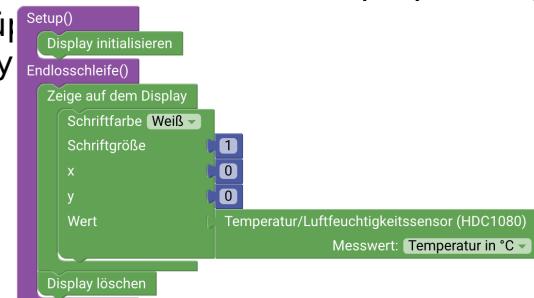
## Text anzeigen

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:

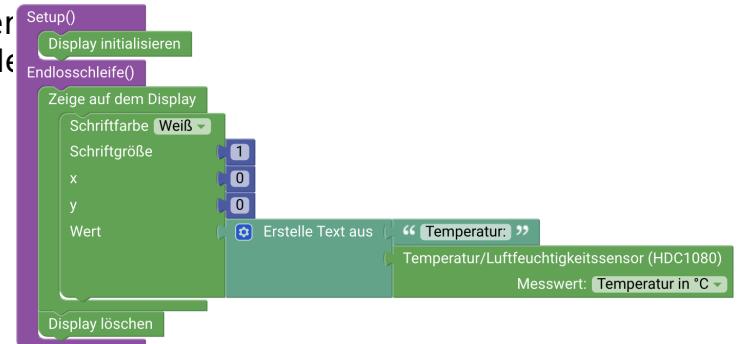


## Messwerte anzeigen und beschreiben

Um Messwerte auf dem Display anzeigen zu lassen verknüpft du die *Temperatur/Luftfeuchtigkeitssensor (HDC1080)* direkt mit den *Display* Blöcken.

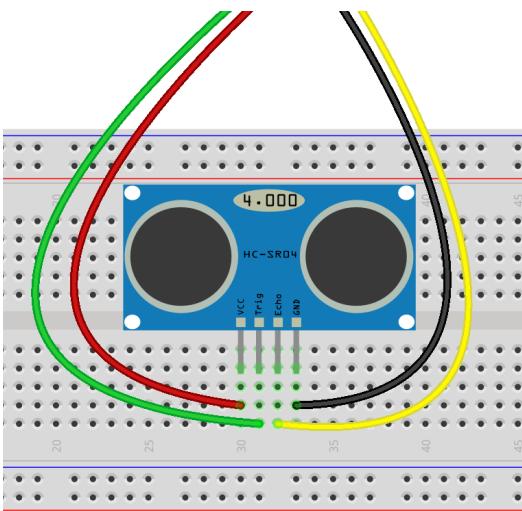
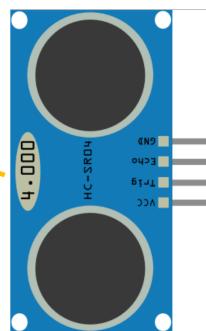
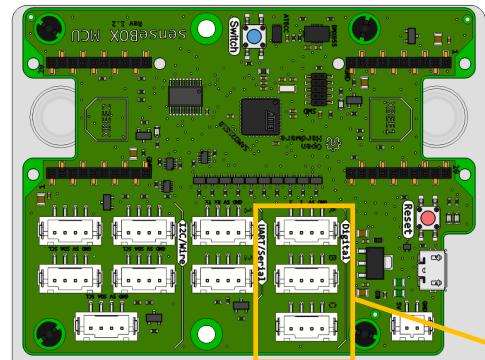


d  
oder erstelle Text aus dem Messwert



**INFO:** Das Display hat eine Auflösung von 128x64 Pixel.

# Der Ultraschall-Distanzsensor



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 **SB02 Die senseBox und JST-Adapterkabel**.

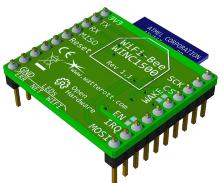
Der Ultraschall-Distanzsensor hat, genau wie alle anderen Sensoren, einen eigenen Block in dem definiert wird wo und wie der Sensor angeschlossen ist.

Ultraschall Abstandssensor an Port A  
Trigger 1 Echo 2

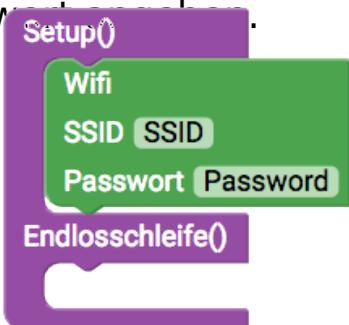
# Übertragen an die openSenseMap

## Wifi Verbindung herstellen

Verbinde das Wifi Bee mit dem Xbee Steckplatz 1

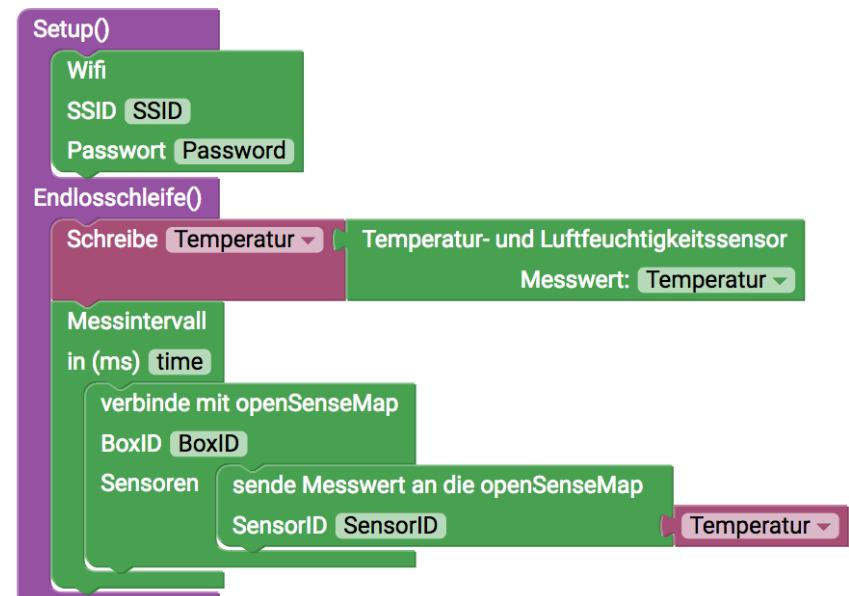


Anschließend musst du den Wifi Block ins Setup ziehen und deinen Netzwerknamen (SSID) und das Passwort eingeben.



## Senden an die openSenseMap

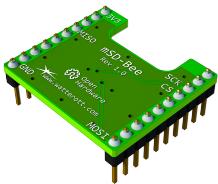
Nach der Registrierung deiner senseBox auf der openSenseMap erhältst du eine BoxID und für jeden Messwert eine SensorID. Trage die BoxID in den „verbine mit openSenseMap Block“ ein. Das Messintervall steuert die Zeit nach der ein Messwert übertragen wird



# Speichern auf SD-Karte

## Datei erstellen

Verbinde zuerst das SD-Bee mit dem Xbee Steckplatz **2**.



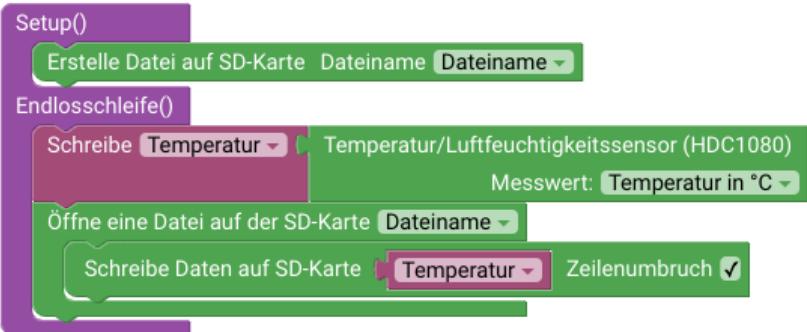
Anschließend wird in der Setup() Schleife eine Datei erstellt. Den Block hierzu findest du in der Kategorie senseBox Output - SD



Um Daten auf die SD-Karte zu speichern muss die Datei immer geöffnet werden. Nach dem schreiben der Messwerte kann die Datei wieder geschlossen werden.

## Messwerte in die Datei schreiben

Die Datei wird nun in der Endlosschleife() geöffnet und der Messwert kann geschrieben werden. Um mehrere Messwerte in einer Datei zu schreiben kannst du diese mithilfe von einem Komma oder Semikolon trennen.

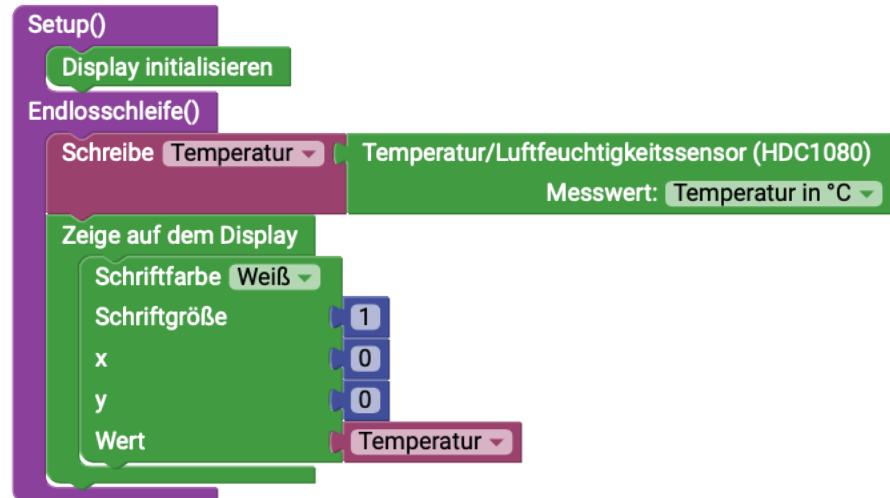


# Variablen - Platzhalter

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



## 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)

# 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.



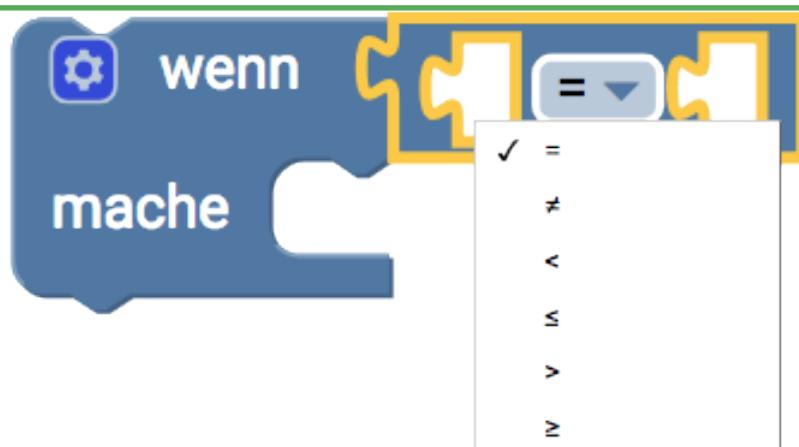
Mit diesem Block kannst du immer zwei Sachen miteinander vergleichen. Eine Erklärung für die verschiedenen Symbole findest du auf der Karte **Operatoren**.



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**:

Das **=-Zeichen**: Hiermit kannst du die senseBox einen Programmierbefehl ausführen lassen, wenn zwei Werte gleich groß sind.

Das **≠-Zeichen**: Hiermit kannst du die senseBox einen Programmierbefehl ausführen lassen, wenn zwei Werte unterschiedlich groß sind.

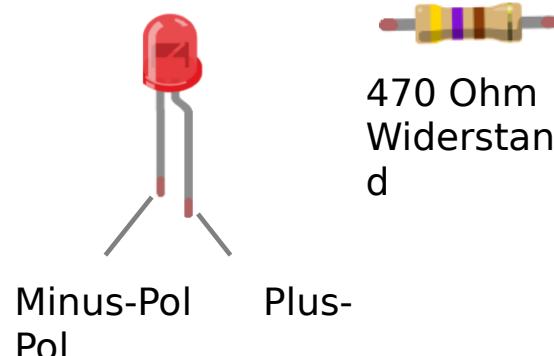
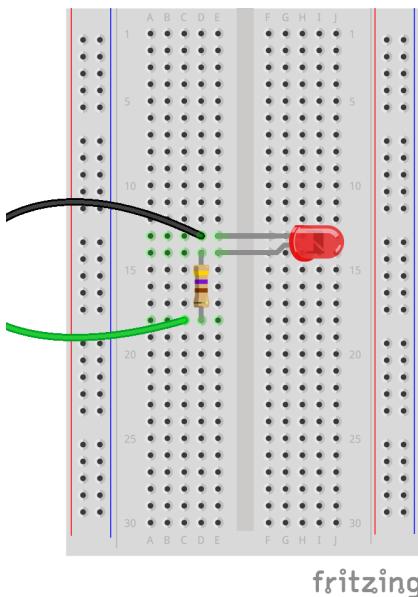
Das **<-Zeichen**: Mithilfe dieses Zeichens kannst du zwei Werte vergleichen lassen. Wenn die Spitze des Symbols auf den kleineren Wert zeigt, führt die senseBox den nächsten Befehl aus.

Das **≤-Zeichen**: Dieses Zeichen ist eine Erweiterung des »kleiner«-Zeichens ( $\leq$ ) und schließt auch Werte ein, die kleiner und gleich groß sind ein.

Das **>-Zeichen**: Mithilfe dieses Zeichens kannst du den Calliope zwei Werte vergleichen lassen. Die Öffnung des Symbols zeigt diesmal auf den größeren Wert.

Das **≥-Zeichen**: Dieses Zeichen ist eine Erweiterung des »größer«-Zeichens ( $\geq$ ) und schließt auch Werte ein, die größer oder gleich groß sind.

# Basis-Modul : Hello World



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

# Modul 1: Zukunft der Stadt I

## Verkehrszähler

**Problemstellung:** Es soll ein neuer Fußgängerüberweg gebaut werden. Dafür stehen zwei Straßen zur Auswahl. Es ist jedoch unklar, an welcher der Straßen sich ein Fußgängerüberweg mehr lohnen würde. Um eine Entscheidung zu fällen soll das Verkehrsaufkommen an beiden Straßen gemessen werden.

**Aufgabe:** Baue und Programmiere einen automatischen Verkehrszähler mit Hilfe des Ultraschall-Distanzsensors

**Schritte:**

1. Schaue dir zuerst die Karte **SB07 „Sder Ultraschall-Distanzsensor“** an. Dort findest du alle wichtigen Informationen um Distanzen mit dem Ultraschall-Distanzsensor zu messen.
2. Überlege dir eine Bedingung, wie du mit Hilfe der Distanzwerte erkennen kannst ob ein Auto vorbeigefahren ist. Schaue dir anschließend die Karten **GI02 -03** an.  
Tipp: Baue dir eine kleine Modellstraße auf dem Tisch und definiere passende Spurbreiten.
3. Lasse dir die Anzahl der gezählten Autos auf dem Display anzeigen, schaue dir dazu die Karte **SB06** an.



## Hilfe

Um zu verhindern, dass Autos doppelt gezählt werden brauchst du eine zweite Bedingung, welche überprüft ob die Spur wieder frei ist. So wird ein vor dem Sensor haltendes Auto nicht mehrfach gezählt.

Für weitere Hilfen kannst du dir die Hilfekarte **H02** zum Verkehrszähler holen.

# Modul 1: Zukunft der Stadt II

## Umweltmessstation mit Internetanbindung

**Problemstellung:** Umweltmessstationen sind oftmals sehr teuer, weshalb recht wenige und meist nur in Städten von den Behörden aufgestellt werden. In der Zukunft wird es normal sein Bürger in die Erfassung von Umweltdaten einzubinden und die Daten direkt z.B. vom Balkon ins Internet zu übertragen.

**Aufgabe:** Baue eine Umweltmessstation, die ihre Messwerte auf dem Display anzeigen und ins Internet auf die openSenseMap ([www.opensensemaps.org](http://www.opensensemaps.org)) übertragen kann.

### Schritte:

1. Schaue dir zuerst die Karte **SB04 „Umweltsensoren auslesen“** an. Dort findest du alle wichtigen Informationen um die Umweltsensoren auszulesen.
2. Lasse dir die Messwerte der Umweltsensoren auf dem Display anzeigen. Informationen dazu findest du auf Karte **SB06 „Das Display“**.
3. Die openSenseMap ist eine Web-Applikation für freie Umweltdaten. Registriere eine neue Umweltmessstation mit einer „Manuellen Konfiguration“ und lasse die Messwerte deiner Station über WLAN übertragen. Informationen dazu findest du auf der Karte **SB08 „Übertragen an die openSenseMap“**. Wenn kein offenes WLAN zur Verfügung steht kannst du vielleicht mit deinem Handy einen WLAN-Hotspot erstellen.
4. Suche Deine Station auf der openSenseMap und schaue, ob die Daten live übertragen werden.



# Modul 2: Zukunft der Energie I

## Intelligente Straßenbeleuchtung

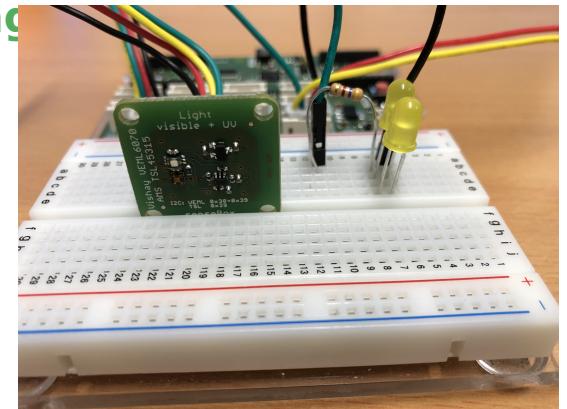
**Problemstellung:** In Deutschland werden jährlich etwa 750 Millionen Euro für die Beleuchtung von Straßen, Plätzen und Brücken ausgegeben. Dabei ist die Beleuchtung gar nicht notwendig, solange niemand die Straße oder den Gehweg benutzt.

**Aufgabe:** Baue und Programmiere eine intelligente Straßenbeleuchtung, die nur dann leuchtet wenn es dunkel ist.

### Schritte:

1. Schaue dir zuerst die Karte **SB04 „Umweltsensoren auslesen“** an. Dort findest du alle wichtigen Informationen um den Helligkeitssensor anzuschließen und zu programmieren.
2. Baue eine Modell-Straßenbeleuchtung aus LEDs auf.
3. Lege einen Helligkeitswert fest, ab dem die Beleuchtung eingeschaltet werden soll. Schaue dir anschließend die Karten **GI02 -03** an.

Tipp: Zum Festlegen eines Helligkeitswertes lässt du dir am besten die gemessenen Werte auf dem Display anzeigen und überprüfst durch abdecken und beleuchten den Wertebereich. Hilfe zum Display findest du auf der Karte **SB06 „Das Display“**.



# Modul 2: Zukunft der Energie II

## Intelligente Straßenbeleuchtung mit Bewegungsmelder

**Problemstellung:** In Deutschland werden jährlich etwa 750 Millionen Euro für die Beleuchtung von Straßen, Plätzen und Brücken ausgegeben. Dabei ist die Beleuchtung gar nicht notwendig, solange niemand die Straße oder den Gehweg benutzt.

**Aufgabe:** Baue und Programmiere eine intelligente Straßenbeleuchtung, die nur dann leuchtet wenn es dunkel ist und jemand vorbei geht oder fährt.

**Schritte:**

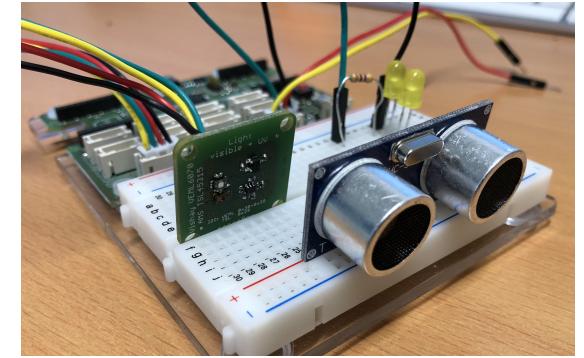
1. Schaue dir zuerst die Karte **SB04 „Umweltsensoren auslesen“** an. Dort findest du alle wichtigen Informationen um den Helligkeitssensor anzuschließen und zu programmieren.
2. Baue eine Modell-Straßenbeleuchtung aus LEDs auf.
3. Lege einen Helligkeitswert fest, ab dem die Beleuchtung eingeschaltet werden soll. Schaue dir anschließend die Karten **GI02** und **GI03** an.

Tipp: Zum Festlegen eines Helligkeitswertes lässt du dir am besten die gemessenen Werte auf dem Display anzeigen und überprüfst durch abdecken und beleuchten den Wertebereich. Hilfe zum Display findest du auf der Karte **SB06 „Das Display“**.

4. Überlege dir eine zusätzliche Bedingung, sodass die Beleuchtung nur dann angeschaltet wird wenn jemand vorbei geht oder fährt. Mit welchem Sensor lassen sich solche Bewegungen zuverlässig erkennen?

### Hilfe

Für weitere Hilfen kannst du dir die Hilfekarte **H01** zur intelligenten Straßenbeleuchtung holen.



# Modul 3: Zukunft der Gesundheit

## Kollektive Feinstaubmessung

**Problemstellung:** In Grundschulen sehr viele Kinder von ihren Eltern mit dem Auto gebracht. Dadurch geht nicht nur ein Stück Selbstständigkeit verloren, vor den Schulen ist morgens und nachmittags ein Verkehrschaos vorzufinden, das für die Kinder gefährlich sein kann. Ein Problem dabei ist unter anderem die erhöhte Feinstaubbelastung, die gerade für Kinder gesundheitsschädlich sein kann.



**Aufgabe:** Baue und Programmiere einen Feinstaubsensor der die Messwerte auf SD-Karte speichert.

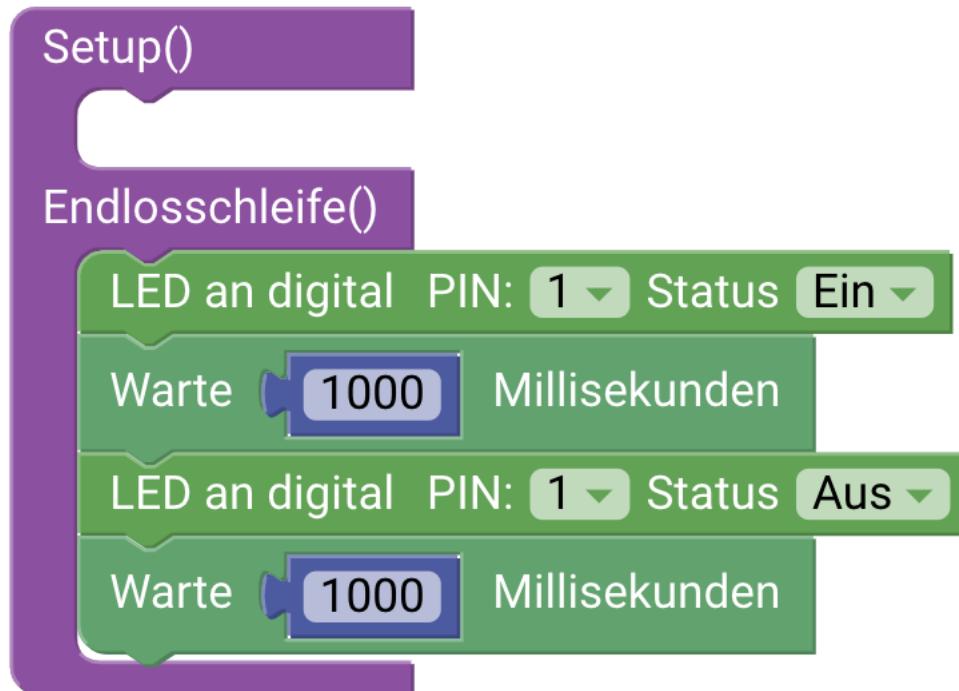
### Schritte:

1. Schaue dir zuerst die Karte **SB05 „Feinstaubsensor auslesen“** an. Dort findest du alle wichtigen Informationen, um Feinstaub mit dem Feinstaubsensor zu messen.
2. Lasse dir die aktuellen Feinstaubwerte auf dem Display anzeigen. Schaue dir das die Karte **SB06** an.
3. Die Feinstaubkonzentration ist über den Tagesverlauf nicht konstant sondern verändert sich. Um eine Zeitreihe der Messwerte zu erstellen, speichern wir diese im nächsten Schritt auf eine SD-Karte. Alle Informationen zum Speichern auf SD-Karte findest du auf der Karte **SB09**.

### Hilfe

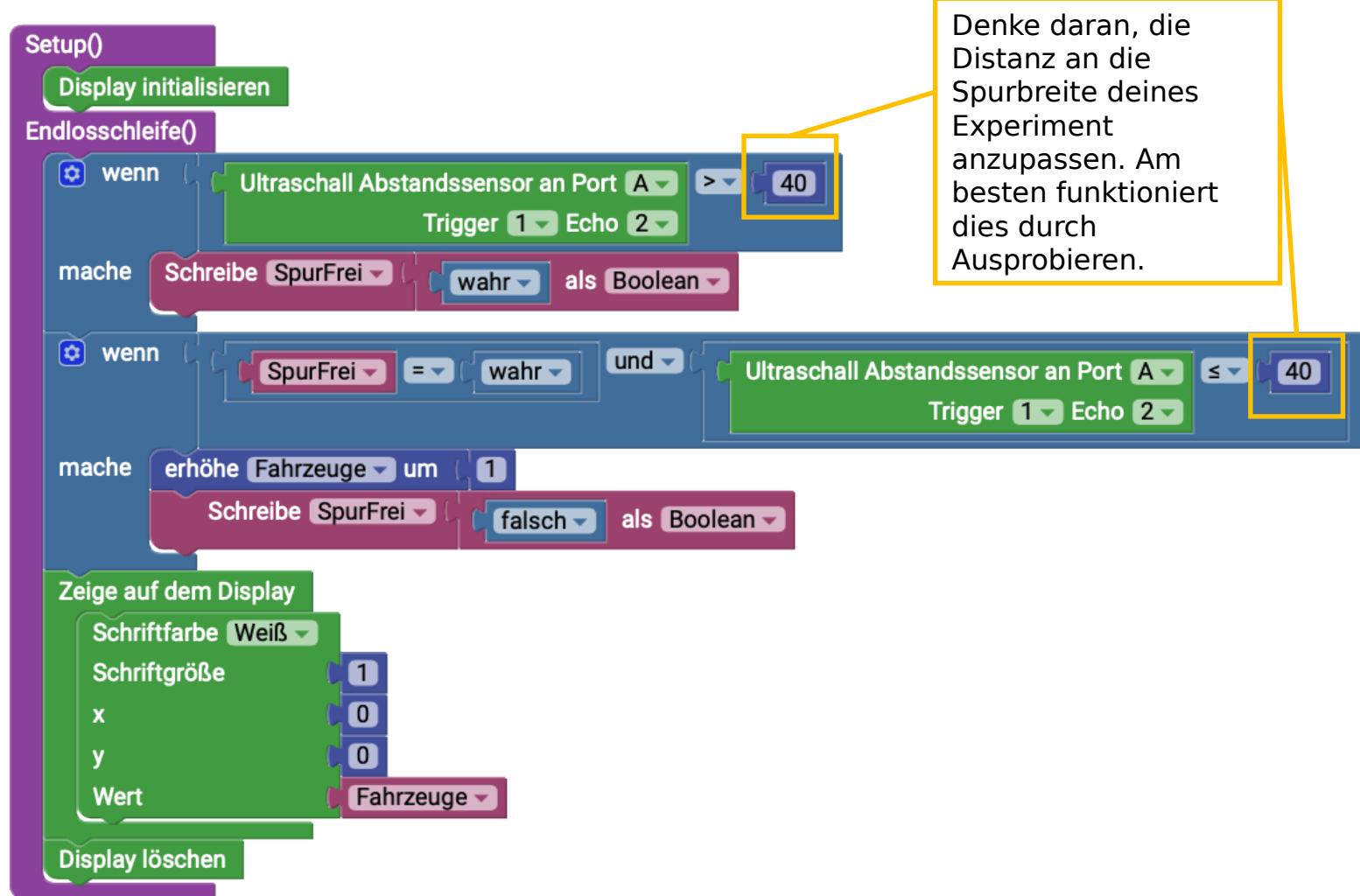
Lege dir zwei Variablen an, in die du jeweils die aktuellen Messwerte speicherst. Du kannst die Variablen dann verwenden um die Messwerte sowohl auf dem Display anzeigen zu lassen als auch zum Speichern auf die SD-Karte.

# Lösung: Basis-Modul

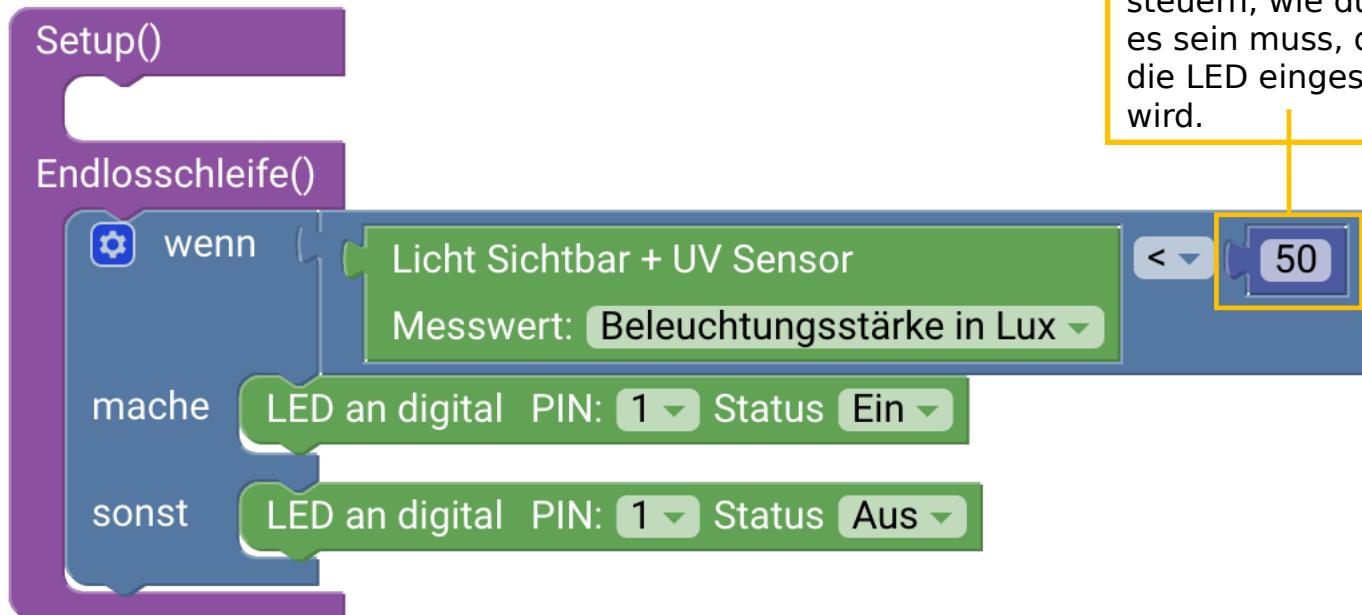


# Lösung: Modul I

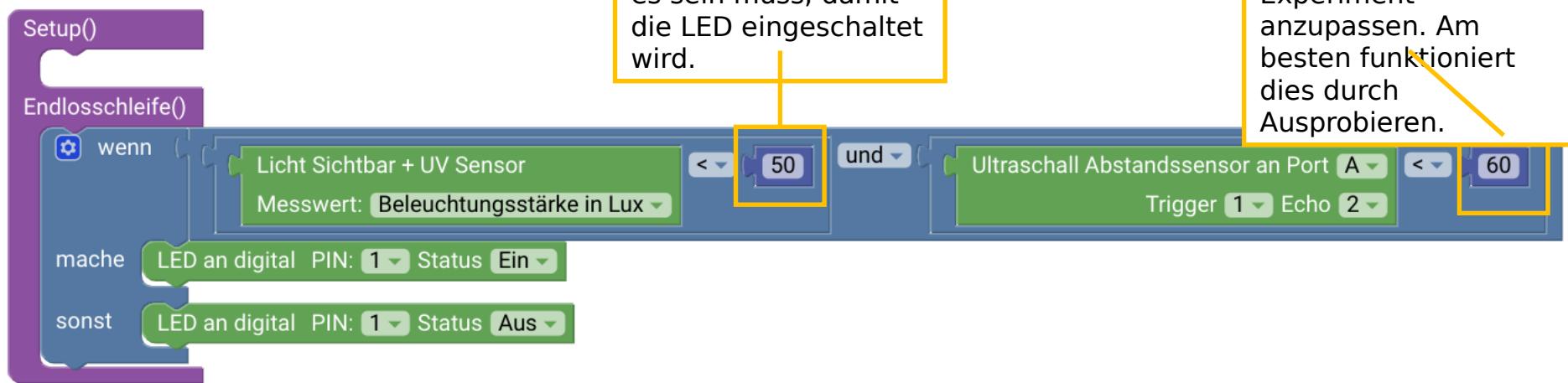
Denkt daran die Spurbreiten auf euer Experiment anzupassen!



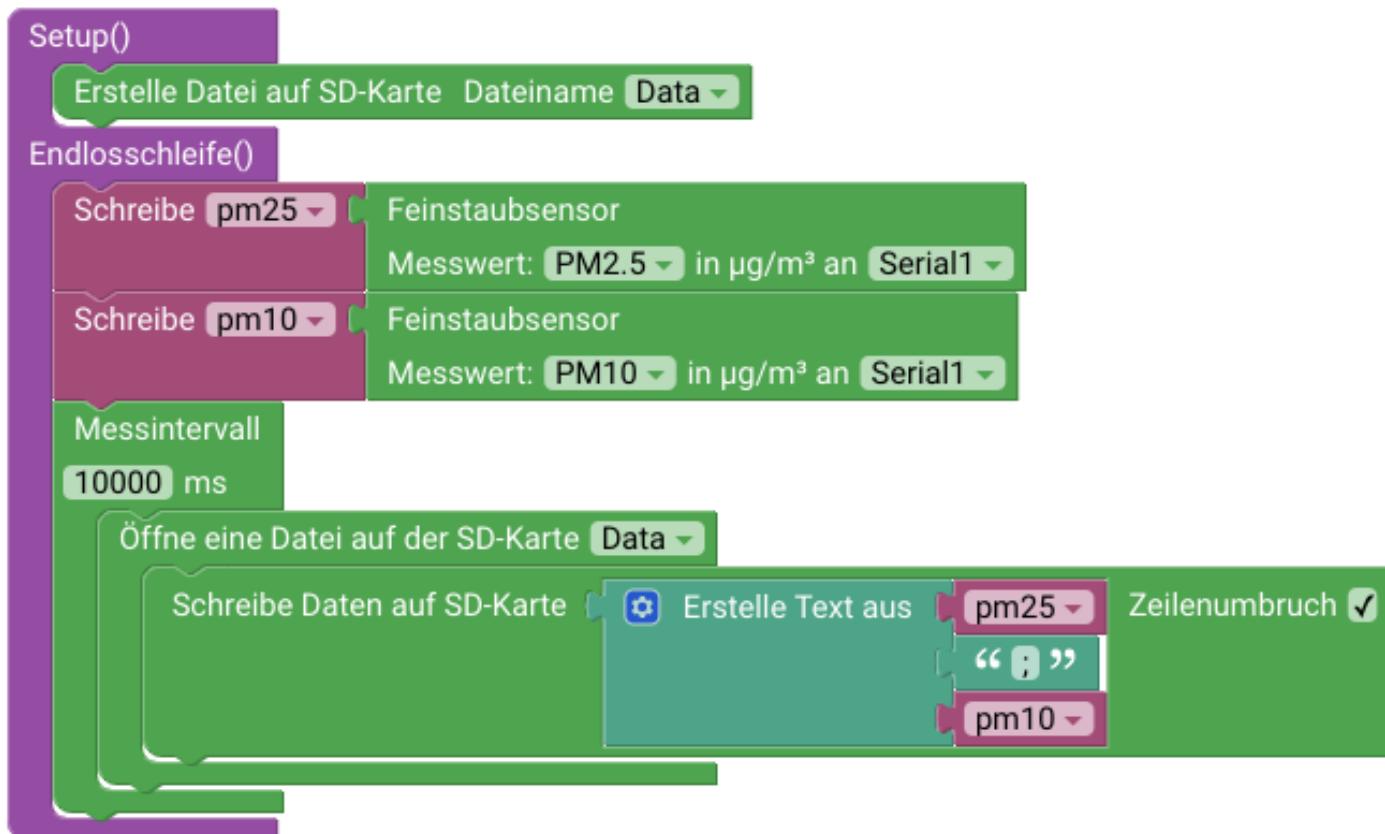
# Lösung Modul 2: Zukunft der Energie I



# Lösung Modul 2: Zukunft der Energie II



# Lösung: Modul 3: Zukunft der Gesundheit



# Hilfekarte: Verkehrszähler

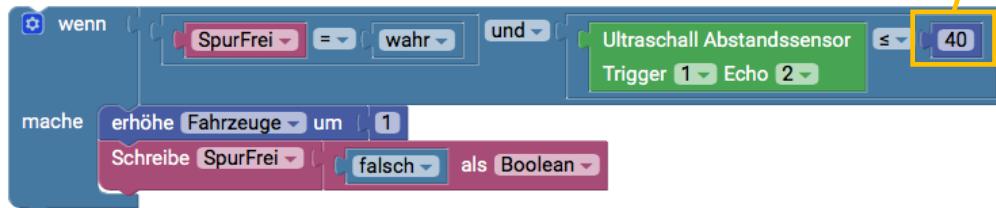
## Den Verkehr zählen

Um den Verkehr zu zählen muss ständig überprüft werden, ob eine bestimmte Distanz (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 Zählen auf dem Display anzeigen zu lassen!