senseBox

Learning cards

SB
01

senseBox MCU – General Overview

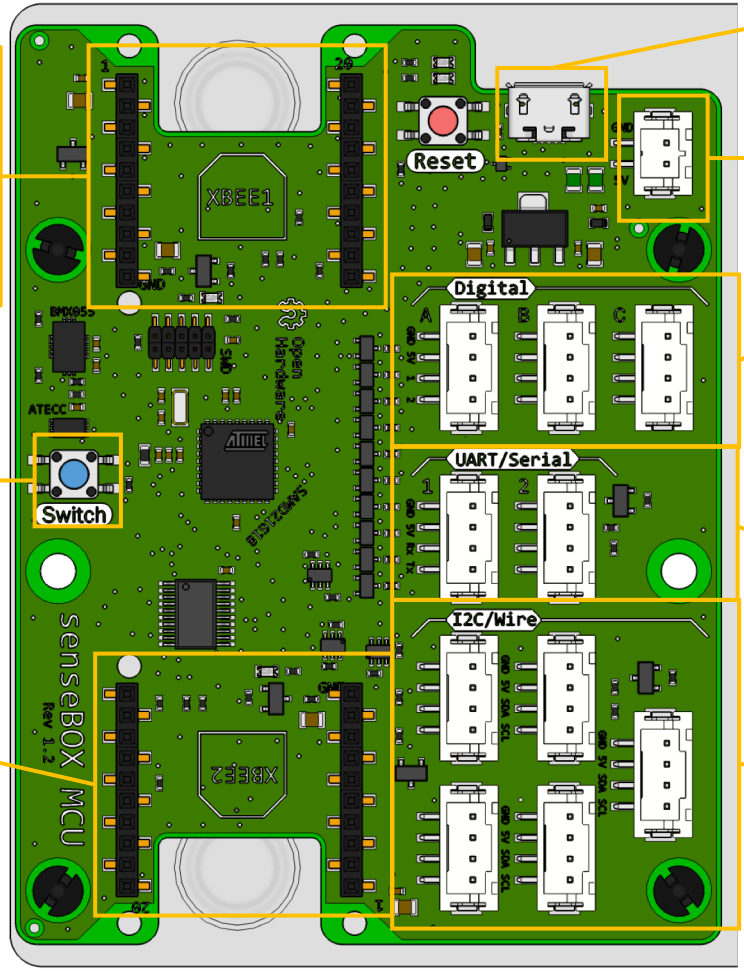
XBEE-Slot 1

BEEs are small add-on modules to add functions to the senseBox such as WiFi or an SD card slot. Connect the **WLAN module** here.

Button

XBEE-Steckplatz 2

BEEs are small add-on modules to add functions to the senseBox such as WiFi or an SD card. Connect the **SD module** here



USB-Connection

Battery-Connector

Digital/Analog-Ports

Sensors and actuators are connected here via the breadboard.

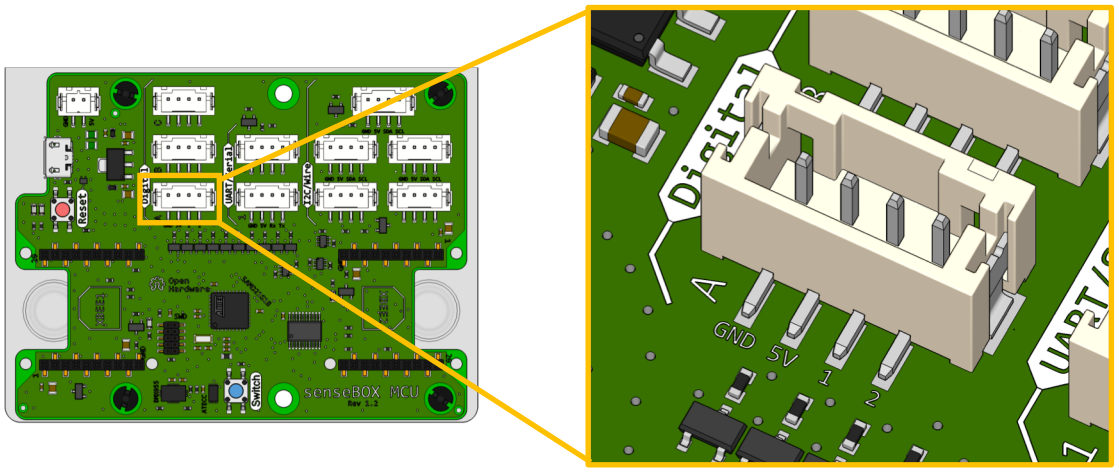
UART/Serial-Ports

This is where the fine dust sensor plugged

I2C/Wire-Ports

All environmental sensors and the display are connected here..

The senseBox and JST adapter cable



Each **Digital/Analog-Port** on the senseBox MCU has four different pins.

The **GND-Pin** is the negative terminal and is always connected with the black cable.

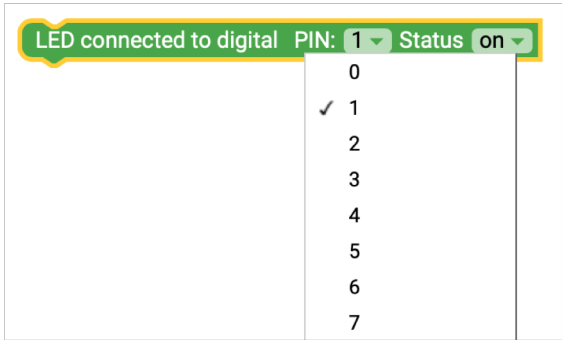
The **5V-Pin** is used to power the sensors with the red cable.

The pins labeled with **1** und **2** are the digital and analog pins 1 and 2 respectively. You will see that his numbering counts sequentially up to pin 6 at port Digital C.

Setup Pins in the Blocks

In order for your own programs to work properly, you must select the pin to which your consumer (e.g. an LED) is connected.

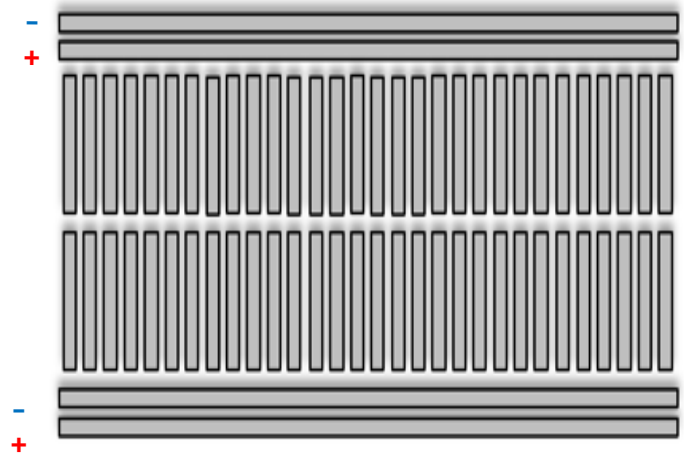
For an LED on Digital 1, the block would look like this:

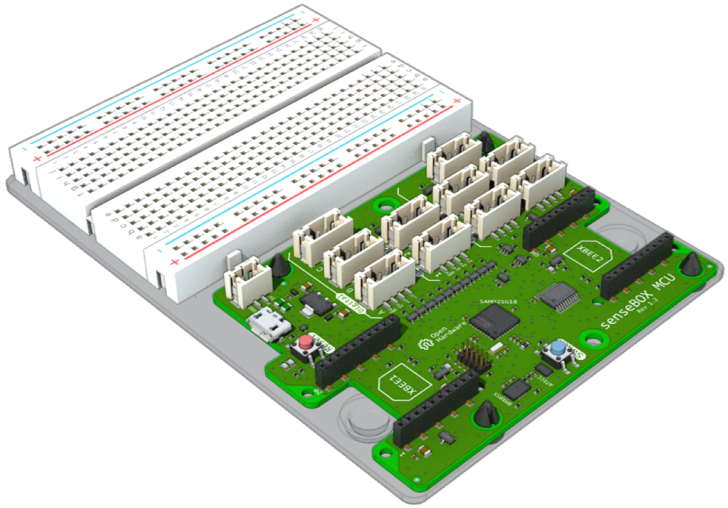


The Breadboard

The breadboard helps you to connect circuits without soldering. The electronic components are simply plugged into the spring contacts, so a circuit can be quickly changed by replugging.

The breadboard consists of two mirrored, non-conductively connected sides.





Each of these consists of two long rows for the positive and negative terminals and 30 rows with five spring contacts each, which are labeled with a to e or f to j.

The positive and negative terminals as well as the five spring contacts of a row are conductively connected as shown above

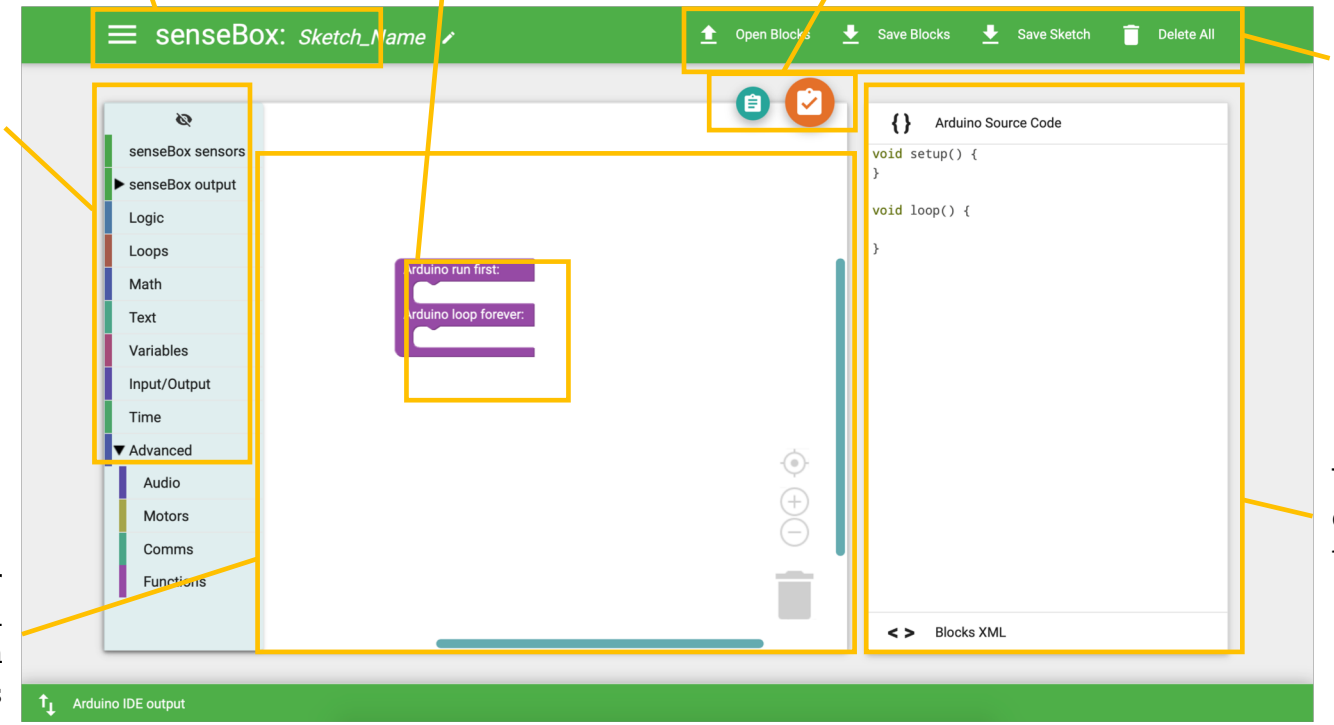
The programming interface

Here you can access the settings menu of Blockly for senseBox and change the name of your program (sketches).

This block is loaded automatically when the user interface is started. The setup function is executed once when the program is started. The endless loop is repeated continuously.

With this button you can compile (convert to machine language) and download your program to transfer it to your senseBox MCU.

This button copies your program code to the clipboard.



These buttons hide all the blocks you need to program the senseBox.

With these buttons you can open projects, save them and delete your current project.

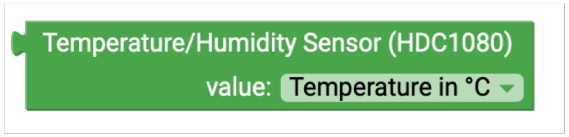
This is your workspace. Here you compose your program from blocks

Your program source code is displayed in this window.

SB 05

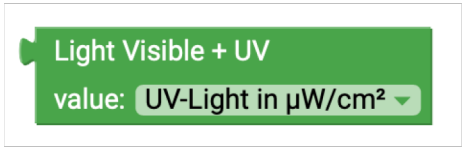
Reading out environmental sensors

Airpressure

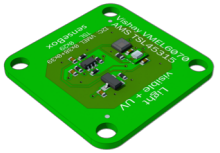


This block gives you the reading of the air pressure sensor. The sensor can measure the air pressure as well as the temperature. You can select the desired measured value in the dropdown menu.

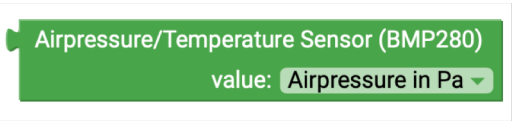
Light-/UV-Light



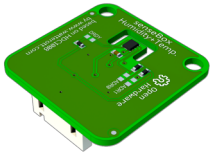
This block gives you the measured value of the light sensor. In the dropdown menu you can select the desired measured value. The sensor can measure the visible light (lux) and the UV light.

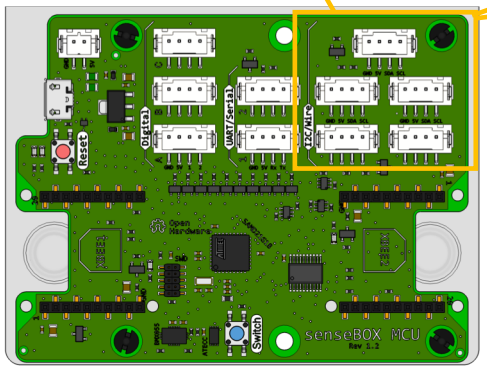


Temperature & Humidity



This block gives you the measured value of the temperature & humidity sensor. In the dropdown menu you can select the desired measured value.





I2C/Wire -Port

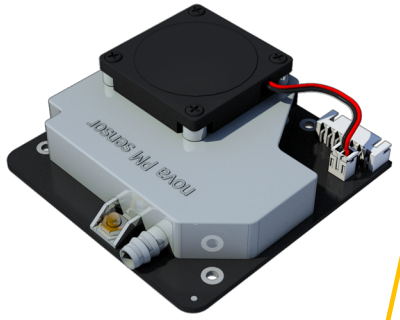
Connecting the sensors

All environmental sensors of the senseBox are connected to the I2C/Wire port via a JST cable. More about the JST cables you can find on the card **The senseBox and JST cables**

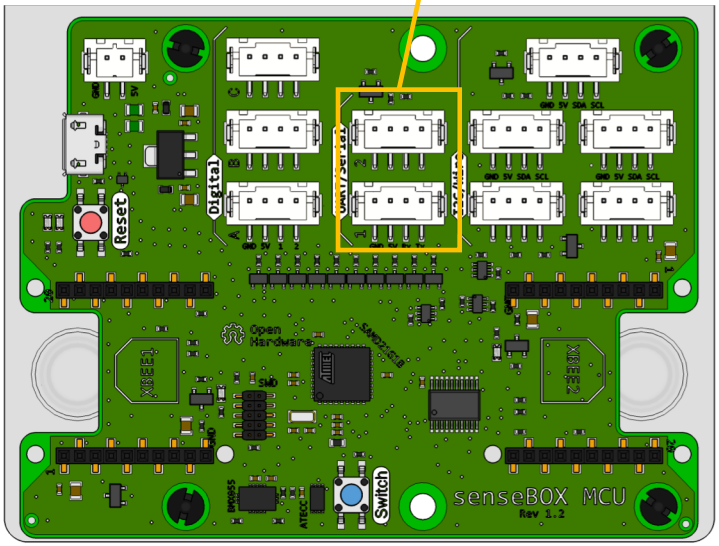
Read particulate matter concentrations

Connecting the sensors

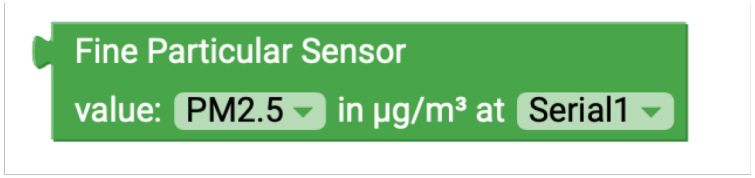
The fine dust sensor is connected via the fine dust sensor JST cable to one of the two UART/Serial ports..



UART/Serial



Read particulate matter sensor values



This block gives you the reading of the particulate matter sensor. In the dropdown menu you can select the desired value.

The fine dust sensor can measure fine dust in two different particle sizes.

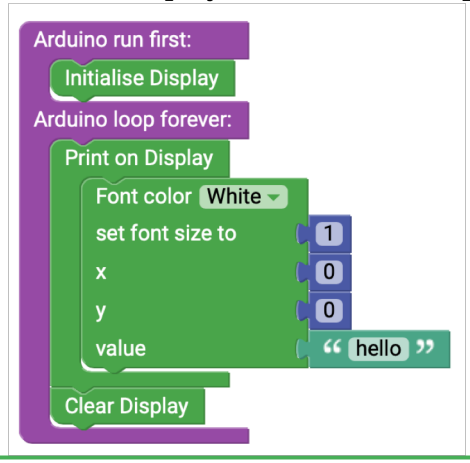
PM2.5: Indicates the number of fine dust particles <2.5 µm in µg/m^3.

PM10: Indicates the number of fine dust particles <10 µm in µg/m^3

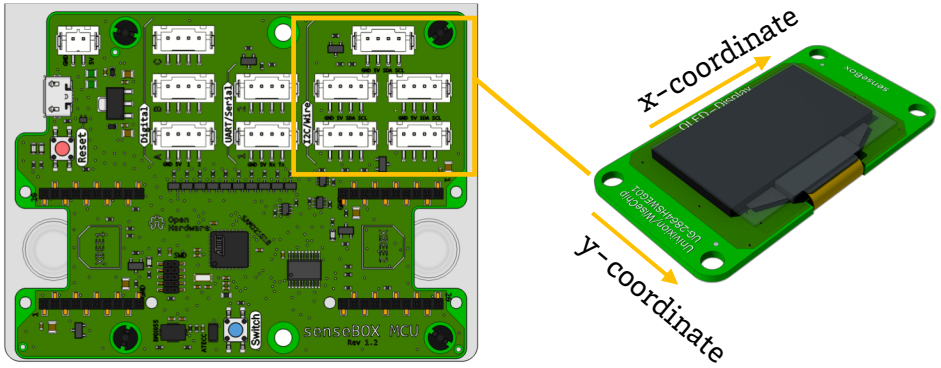
The OLED-Display

Text anzeigen

To be able to use the display, it must be initialized in *Setup( )*. Then you can use a few blocks in the *Loop( )* to display text or measured values on the display. Here is an example for a short text:

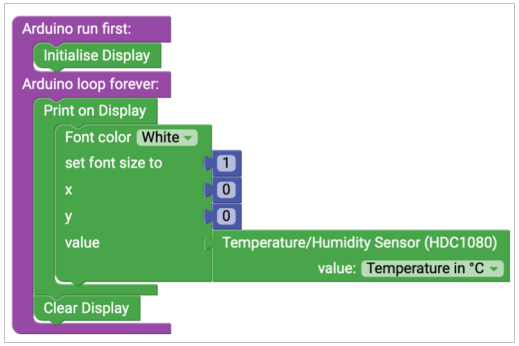


I2C/Wire-Ports

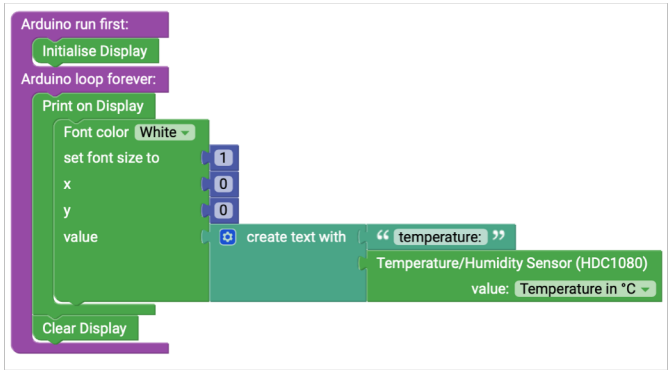


Display and describe measured values

To display measured values on the display, either link the sensor block directly to the display blocks,

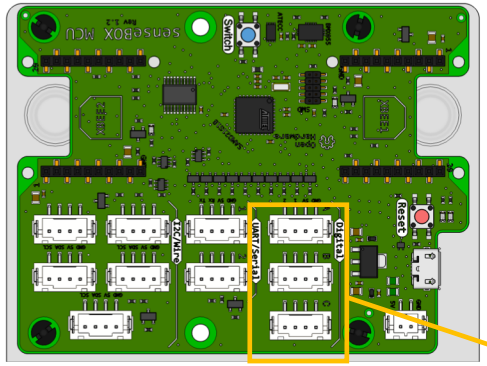


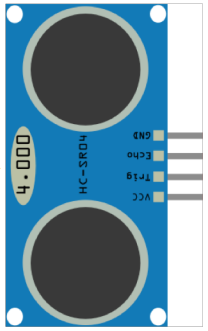
or create a text with a description and the measured value.

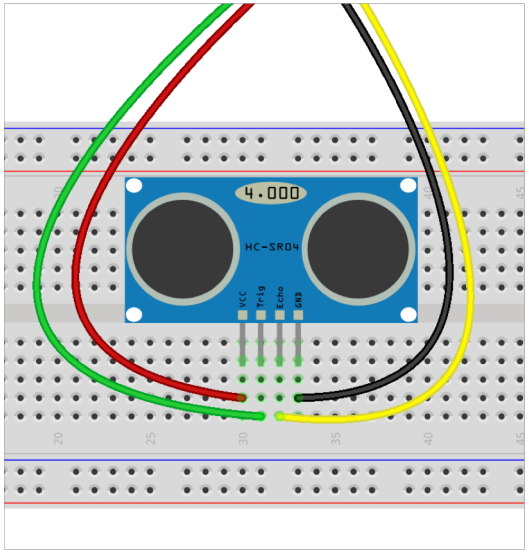


INFO: The display has a resolution of 128x64 pixels.

The ultrasonic distance sensor





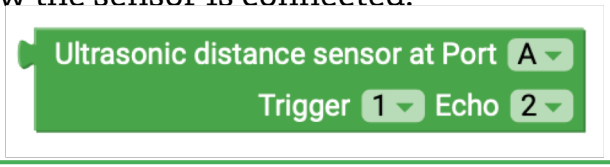


You will need a JST adapter cable to connect the ultrasonic distance sensor. The sensor itself has four different pins: VCC, Trig, Echo and GND. These four connectors must be connected to the four cables of the JST adapter cable.

GND	GND	(black cable)
Echo	2	(yellow cable)
Trig	1	(green cable)
VCC	5V	(red cable)

Note: If you connect the sensor to another port, the assignment for trigger and echo also changes.
You can find out more about how the JST adapter cables work on the card **The senseBox and JST cables**.

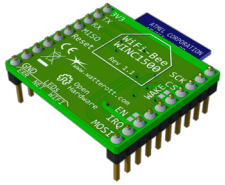
The ultrasonic distance sensor, like all other sensors, has its own block in which it is defined where and how the sensor is connected.



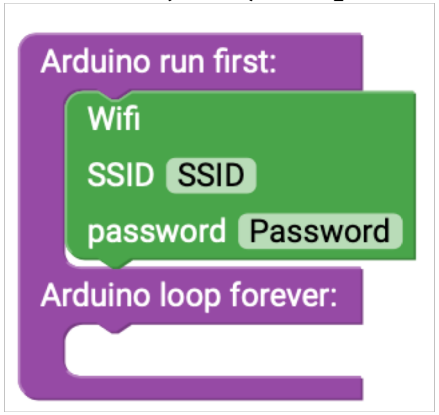
Broadcast on the openSenseMap

Establish Wifi connection

Connect the Wifi Bee to the **Xbee-Port 1**

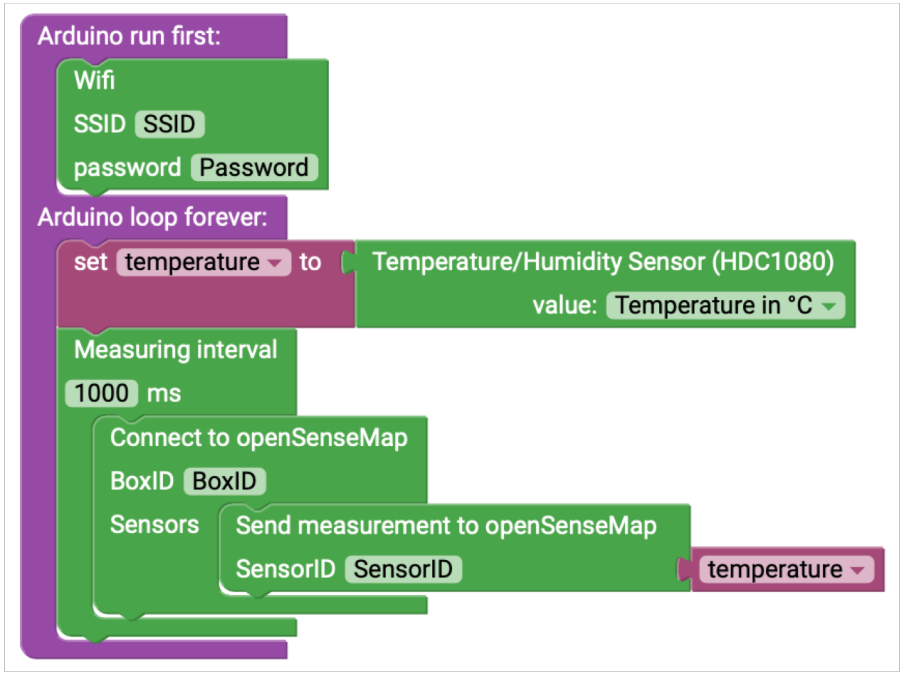


Then you have to drag the Wifi block into the setup and enter your network name (SSID) and password.



Send to the openSenseMap

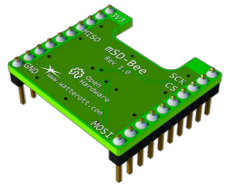
After registering your senseBox on the openSenseMap you get a BoxID and for each sensor a SensorID. Now enter the BoxID into the "connect to openSenseMap " and the SensorID into the "send measurement to openSenseMap". The measurement interval determines how often the measurement should be performed.



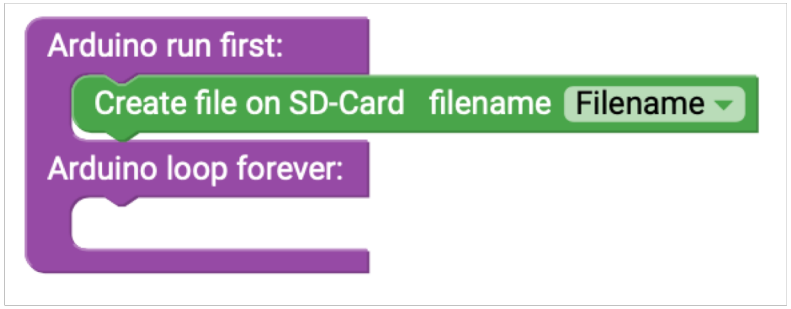
Save on SD-Card

Create File

Connect the SD-Bee to Xbee-Port 2.



A file must be created in the Setup( ) loop. You can find the block for this in the category senseBox Output - SD

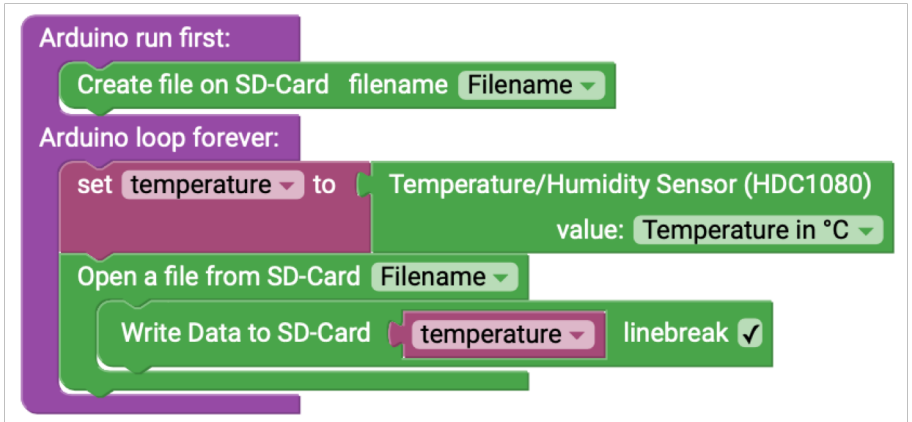


To save data to the SD card, the file must always be opened. After writing the measured values, the file can be closed again.

Write measured values to the file

First the measured temperature value must be written into a variable. To learn more about variables look at the GI1 card.

To save the variable in the file, first open it with the "Open file on SD card" block and then write the variable to the file with the "Write data to SD card" block. The "Open file on SD card" block closes the file automatically after writing.



Variables – placeholders

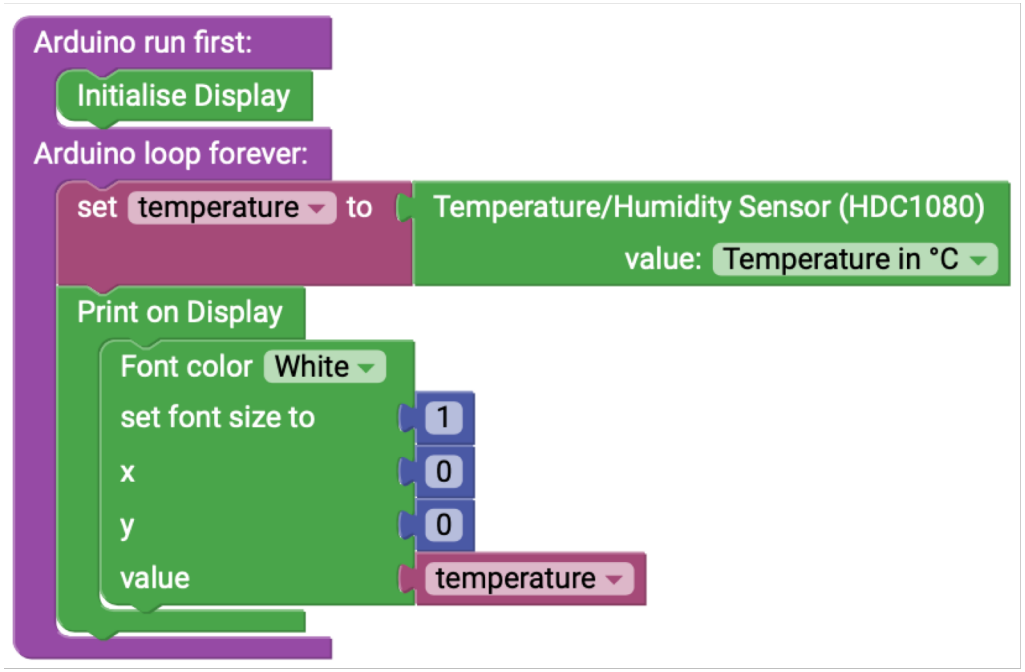
Variables or placeholders are used in computer science for many different things. They are a kind of box, which is provided with a name, into this box you can deposit now different things, e.g. numbers or also texts, and call these up later again.

Variables – Data types

Depending on what you want to store in a variable you have to choose the right data type.

character:	For single text characters
Text:	For complete sentences or words
Number:	For numbers between -32768 and 32768
Long:	For large numbers between -2147483648 and 2147483647
Float:	For comma numbers (e.g. 25.56)

Variables can change their value in the course of the program, so that for example you always assign the currently measured temperature to the variable "Temperature".



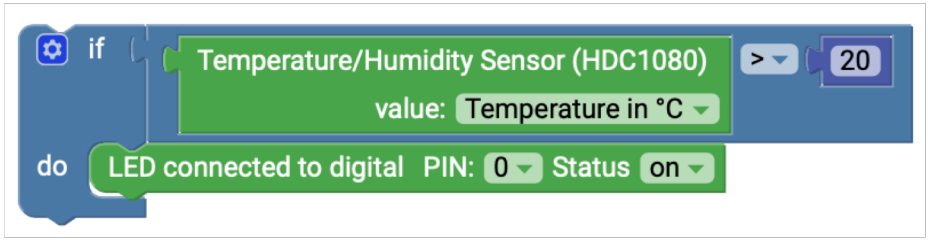
If-Then-Else?

The "If-condition" is one of the most important control structures you will get to know when learning programming. The “If-condition” allows the senseBox to perform certain actions when something specific (e.g. a button has been pressed) has happened.

Logical Comparision

This block allows you to compare two statements. An explanation for the different symbols can be found on the card **Operators**.

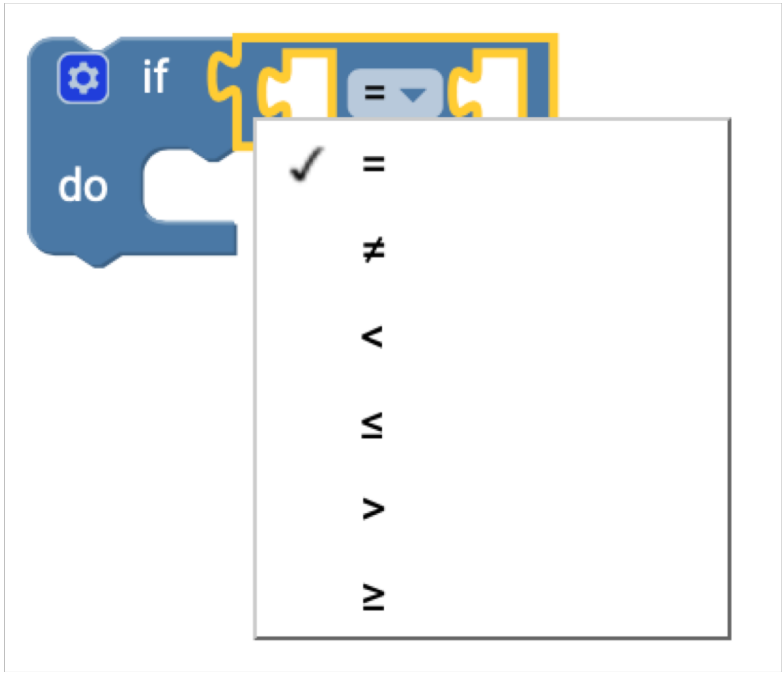
This is the Block for the „If-condition“. This allows you to program what exactly is to be done when a condition is met.



Beispiel: **If** the temperature is higher the 20°C, **then** the LED will be switched on.

Operators

Operators are needed in many situations when programming. Operators can be used to check conditions or compare values.



The following operatros can be found in **logic**:

The **=-sign**: Lets you have the senseBox execute a programming command if two values are the same size.

The **≠-sign**: Lets you have the senseBox execute a programming command if two values are different.

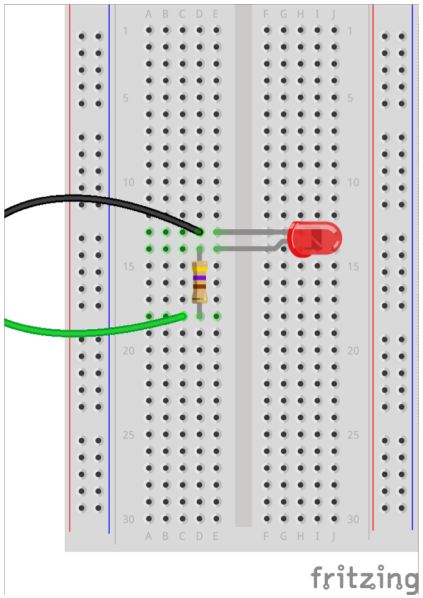
The **< -sign**: You can use this character to compare two values. If the top of the symbol points to the smaller value, the senseBox executes the next command.

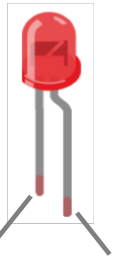
The **≤-sign**: This character is an extension of the "smaller" character (<) and also includes values that are smaller and equal.

The **>-sign**: You can use this character to make the senseBox compare two values. The opening of the symbol this time points to the larger value.

The **≥-sign**: This character is an extension of the "greater" symbol (>) and also includes values that are greater than or equal to one another.

Basic module: Hello World





negative Pole

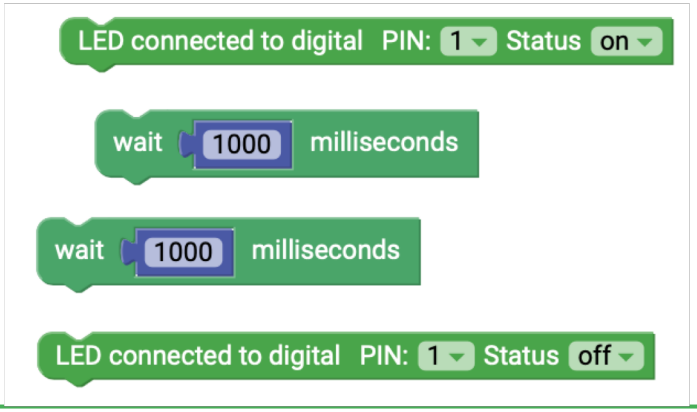
positive pole

470 ohm resistor

In order to connect a LED, you need a resistor (470 ohms) and an adapter cable.
The adapter cable is plugged into the Digital A Port on the senseBox MCU. Then connect the **black** and green cables as shown above.

To make the LED light up you only need this one block:

Task: Build the shown circuit and first light up the the LED. In a second step make the LED Blink. Use the following blocks.



A
02

Module 1: Future of the city I

Traffic counter

Problem: A new pedestrian crossing is to be built. There are two roads to choose from. However, it is unclear which of the streets a pedestrian crossing would be worth more. To make a decision, the traffic on both roads should be measured.

Task: Build and program an automatic traffic counter with the help of the ultrasonic distance sensor

Steps:

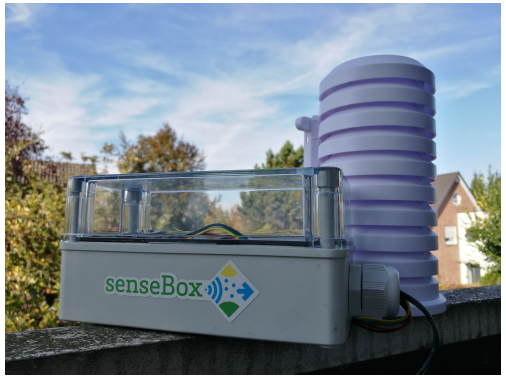
1. First, look at the SB08 "Ultrasonic Distance Sensor" card. There you will find all the important information to measure distances with the ultrasonic distance sensor. Have the readings of the ultrasonic distance sensor displayed on the screen (SB07).
2. Think about a condition on how to use the distance values to see if a car has passed. Then look at the cards GI02-03.
   Tip: Build a small model road on the table and define suitable track widths. Further help can be found on the help card H01.
3. Show the number of cars counted on the display by looking at the SB07 card again.

Help

In order to prevent cars from being counted twice you need a second condition, which checks if the track is free again. So a holding car in front of the sensor is not counted multiple times. For further help you can get the help map H02 to the traffic counter.

Module 1: Future of the city II

Environmental measuring station with internet connection



Problem: Environmental monitoring stations are often very expensive, which is why there usually only set up in cities. In the future it will be normal to involve citizens in the collection of environmental data and to send the data directly e.g. from the balcony to the internet.

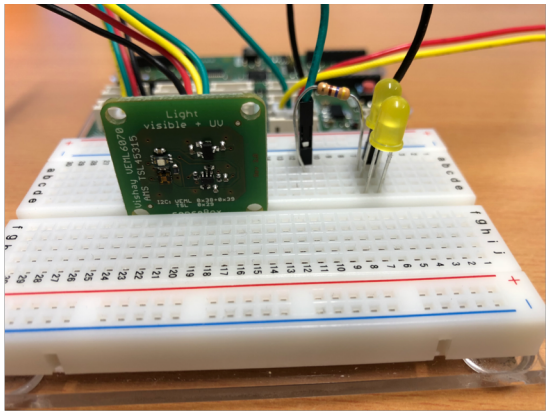
Aufgabe: Build an environmental measuring station that can display their readings on the display and upload them to openSenseMap (www.opensensemap.org).

Steps:

1. First, look at the card SB05 "Reading out environmental sensors". There you will find all the important information to read out the environmental sensors.
2. Have the readings of the environmental sensors displayed on the screen. See the SB07 "The Display" card for more information.
3. The openSenseMap is a free web application for environmental data. Register a "senseBox:edu" and have your station's readings transmitted over Wi-Fi. Information on this can be found on the SB09 "Transfer to openSenseMap" card. If there is no open Wi-Fi, you may be able to create a Wi-Fi hotspot with your phone.
4. Search your station on the openSenseMap and see if the data is transmitted live.

Module 2: Future of the energy I

Intelligent street lighting



Problem: In Germany, about 750 million euros are spent annually for the illumination of streets, squares and bridges. The lighting is turned on at a specific time, not depending to the actual brightness.

Task: Build and program an intelligent street lighting that only lights up when it's dark.

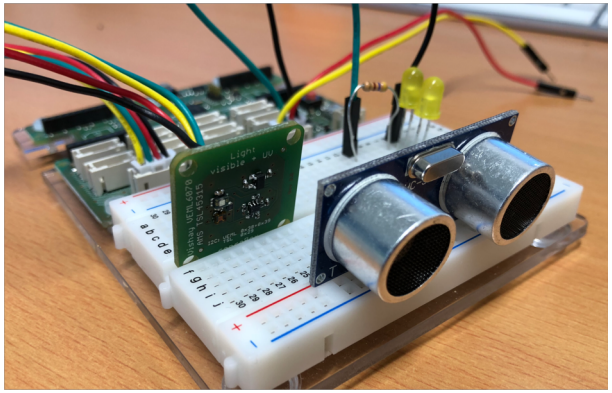
Steps:

1. First, look at the card SB05 "Reading out environmental sensors". There you will find all the important information to connect and program the brightness sensor.
2. Build a model street lighting of LEDs.
3. Set a brightness value from which the lighting should be switched on. Then look at the cards GI02-03.

Tip: To set a brightness value, it is best to display the measured values on the display and check the value range by covering and illuminating. Help for the display can be found on the SB07 "The Display" card.

Module 2: Future of the energy II

Intelligent street lighting



Problem: In Germany, about 750 million euros are spent annually for the illumination of streets, squares and bridges. The lighting is not necessary, as long as nobody uses the street or the sidewalk.

Task: Build and program smart street lighting that only lights up when it's dark and someone is passing or driving by.

Steps:

1. First, look at the card SB05 "Reading out environmental sensors". There you will find all the important information to connect and program the brightness sensor.
2. Build a model street lighting consist of LEDs.
3. Set a brightness value from which the lighting should be switched on. Then look at the cards GI02-03

Tip: To set a brightness value, it is best to display the measured values on the display and check by covering and illuminating the value range. Help for the display can be found on the SB07 "The Display" card.

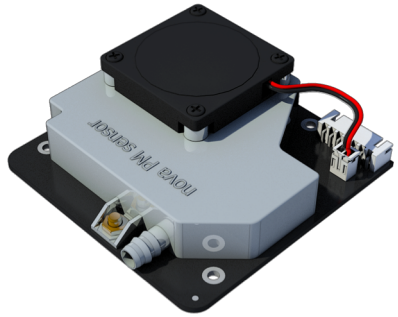
4. Consider an additional condition, so that the lighting is only switched on when someone passes or drives by. Look at the SB08 card.

Help

For further help you can get the help card H01.

Module 3: Future of health

Measurement of particular matter



Problem: Many pupils are driven to school by their parents. As a result, there is not only a traffic chaos in front of the schools, there might be also an increased fine dust pollution

Task: Build and program a particulate matter sensor that stores readings on SD card.

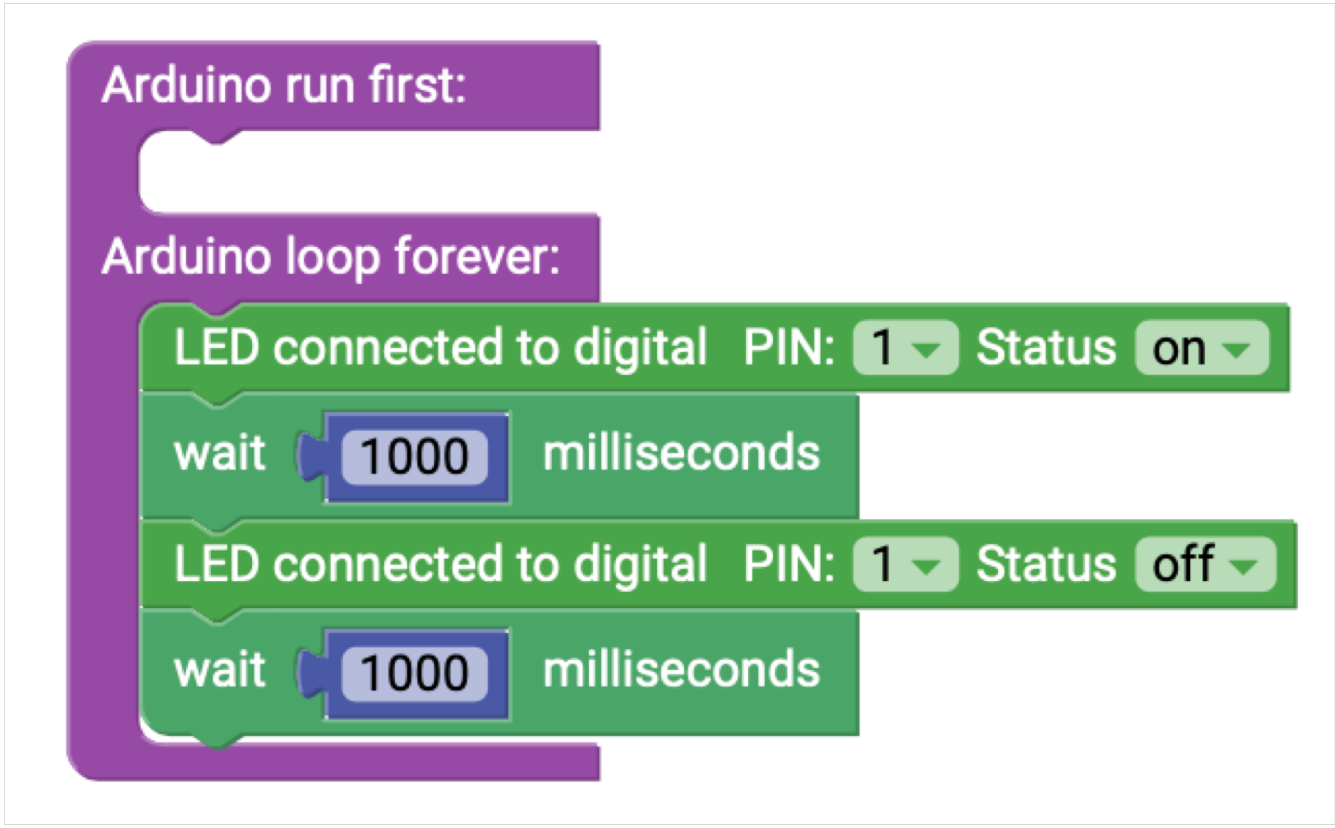
Steps:

1. First, look at the card SB06 "Read particulate matter concentrations". There you will find all the important information to measure particular matter.
2. Display the readings of the particular matter sensor on the OLED-Display. See the SB07 "The Display" card for more information.
3. The particular matter concentration is not constant over the course of the day. To create a time series of the measured values, you need to store them on an SD card. All information about saving to SD card can be found on the card SB10 “Save on SD card”.

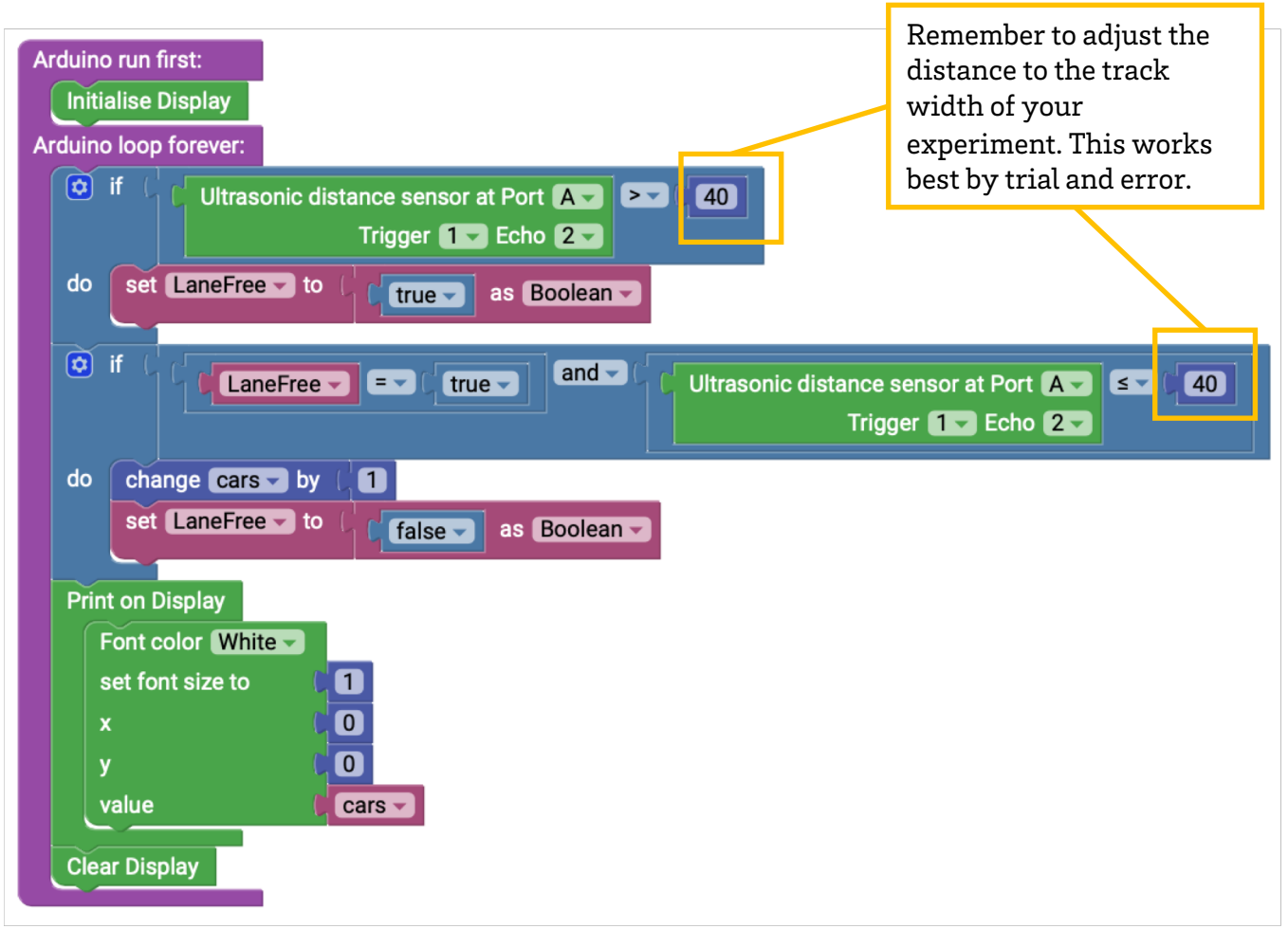
Help

Create two variables, one for PM 2.5 and one for PM 10.

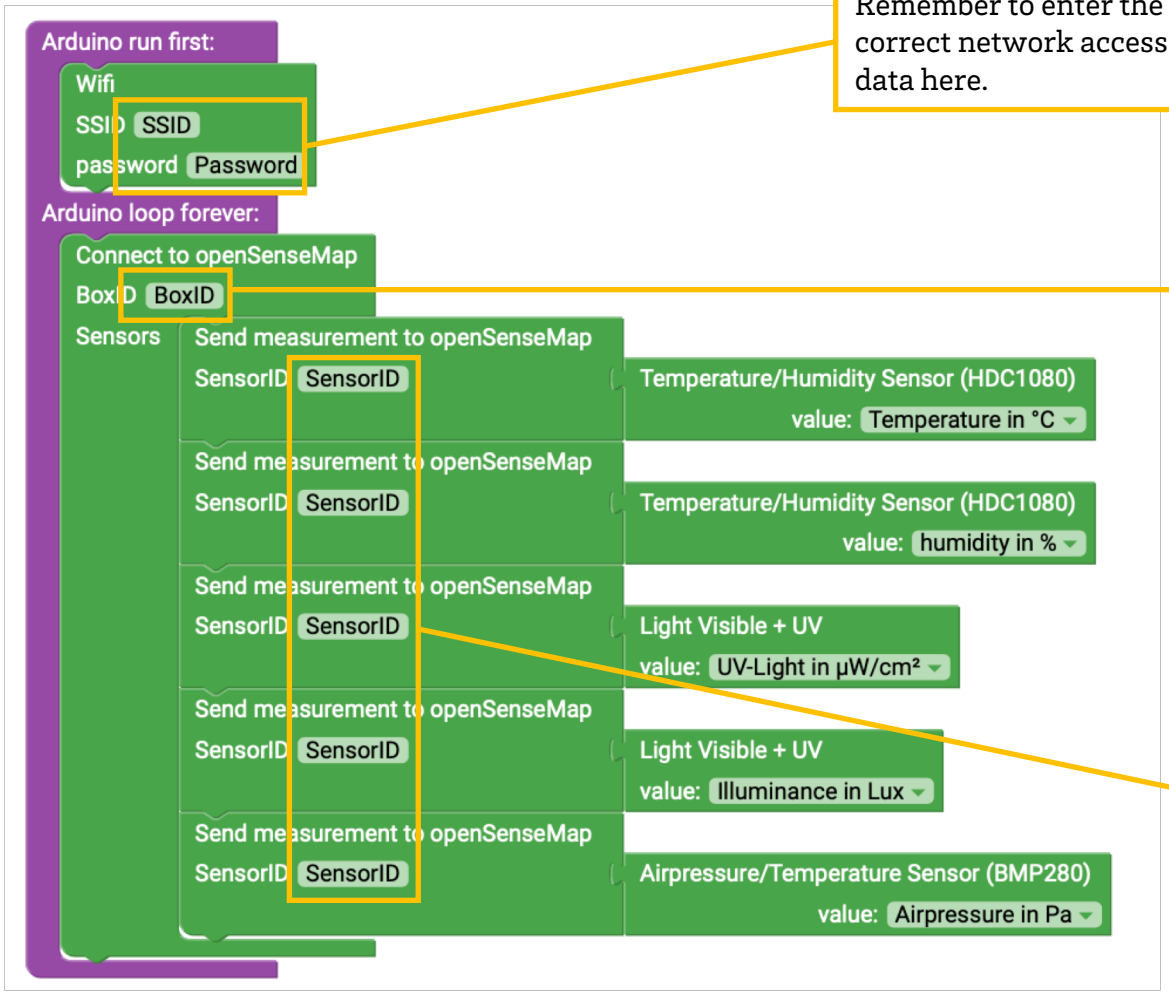
Proposed solution: Basic module



Proposed solution : Module I – Future of the city 1



Proposed solution : Module I – Future of the city 2

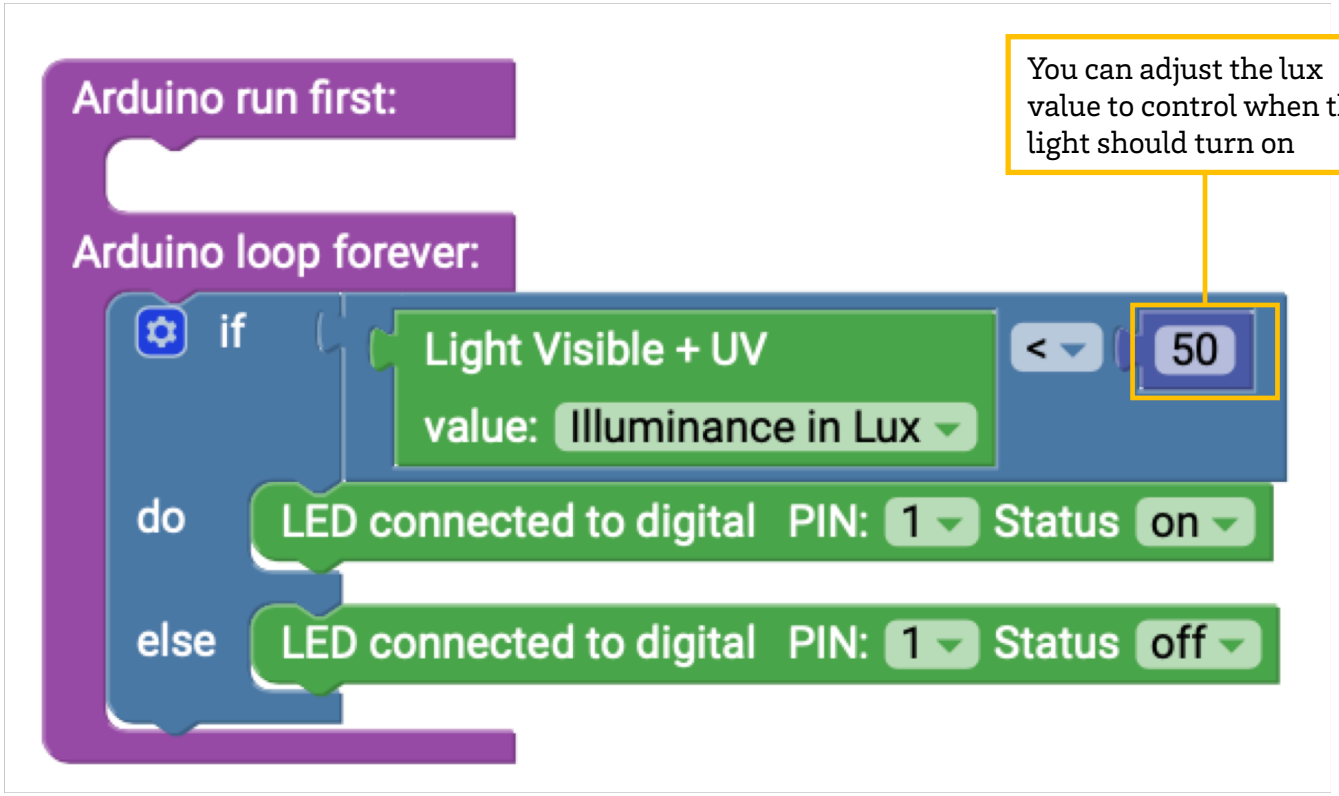


Remember to enter the correct network access data here.

Enter your Box ID here, which was displayed when registering your senseBox.

Enter here the respective Sensos IDs that have been shown when registering your senseBox.

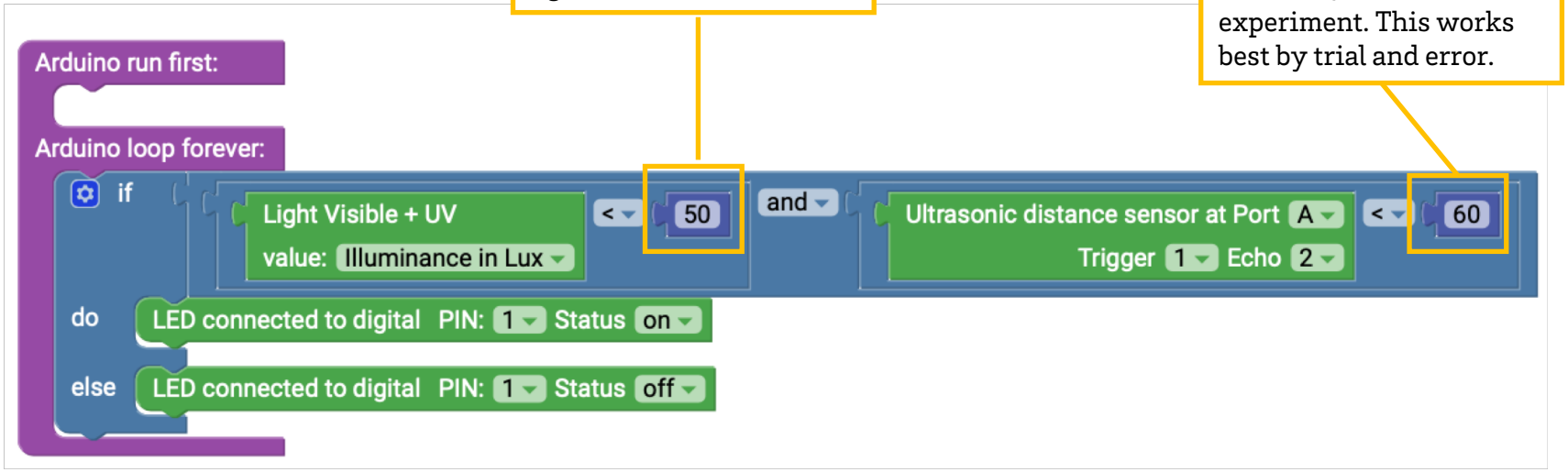
Proposed solution : Module II – Future of the energy 1



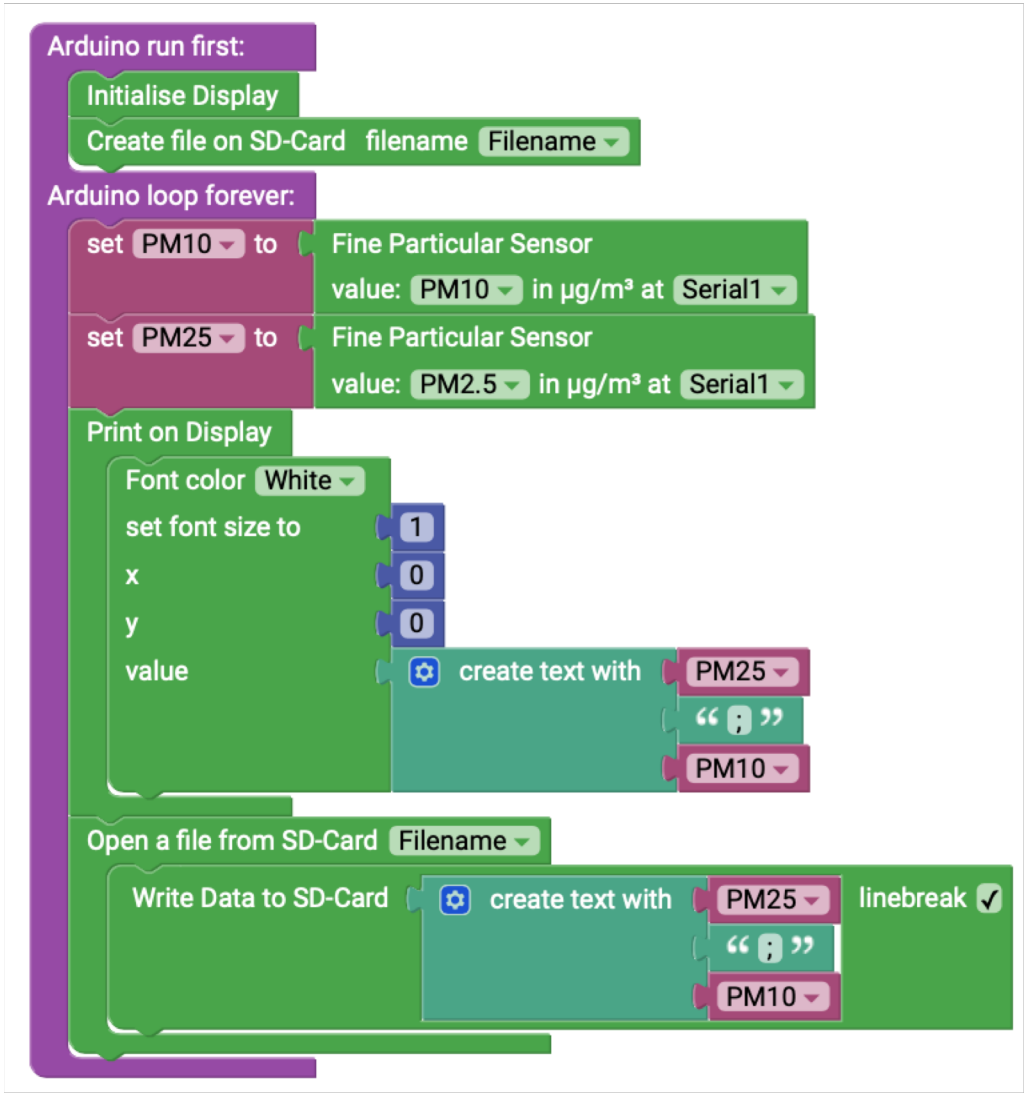
Proposed solution : Module II – Future of the energy 2

You can adjust the lux value to control when the light should turn on.

Remember to adjust the distance to the track width of your experiment. This works best by trial and error.



Proposed solution : Module III – Future of health

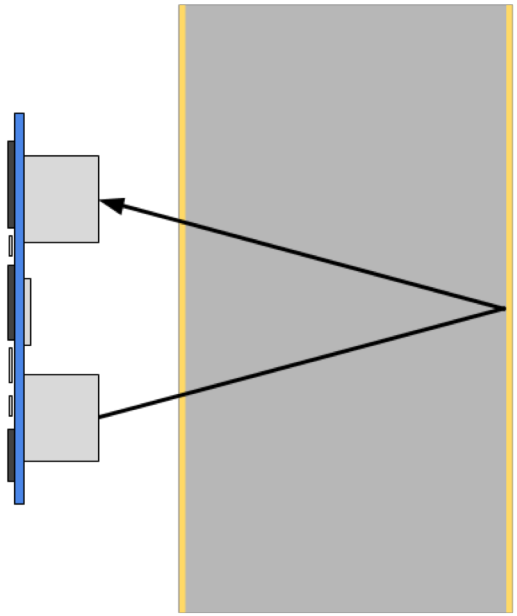


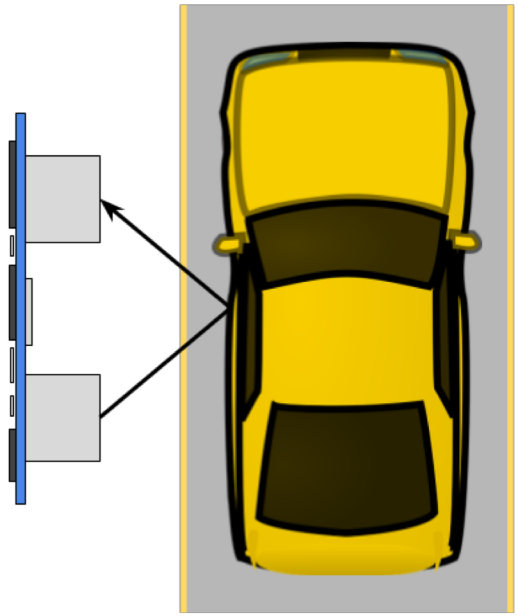
H
01

Help card 1: Counting cars

How to count cars with a distance sensor?

The road, whether in the model or in reality, has a certain width. What happens to the measured distance values, if a car drives by?





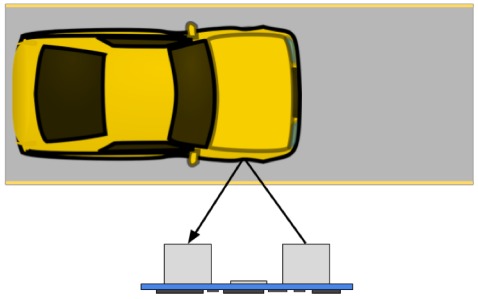
H
02

Help card 2: Incorrect counts

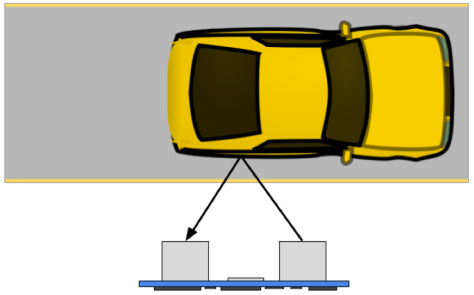
Computers, including the senseBox MCU, work very fast. For your traffic counter this means that it counts so often per minute that a slow car cannot pass before it is measured again. Now probably the first idea would be to just program a pause between measurements. But what if a very fast car drives by?
Exactly, the fast car could be overlooked.

So how can you prevent slow cars from being counted multiple times, but fast cars from being overlooked?

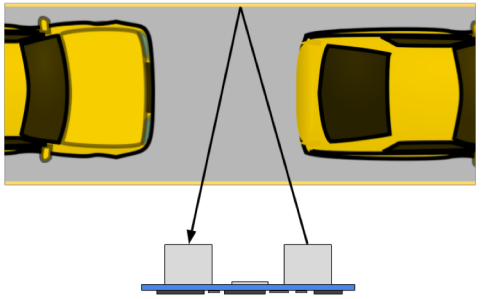
A reliable solution to this problem is to set an additional condition. This condition should stipulate that after a car is counted, the maximum track width must first be measured before it can be counted up again.



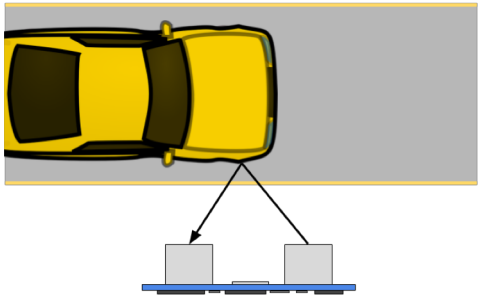
1. Distance A is undershot, the car is counted and the second condition is simultaneously set to "false".



2. Distance A is undershot again, but the second condition equals "wrong". The car is not counted.



3. Distance A is no longer untershot. The 2nd condition is set to "true" again.



4. Distance A is undershot, the car is counted and the second condition is simultaneously set to "false" again.