senseBox

Flashcards S2

The senseBox Flashcards

The senseBox flashcards help you to experiment the senseBox. Besides the basics of computer science you get important information on how to use the components of the senseBox.

The flashcards are divided into two categories:

SB	Important components and programming of the senseBox
GI	Basics and concepts from computer science

For tips on programming and transferring programs, visit: www.sensebox.de/de/go-edu-s2

Explore more information, materials, and projects at: www.sensebox.de/en

Mario Pesch & Verena Witte
senseBox, 4. Auflage 2024

Content

senseBox MCU-S2 – Ports

SB 01

XBEE-Plug-in

XBEEs are small additional modules with which the senseBox can be expanded with functions such as Bluetooth or LoRa. Connect the respective module here.

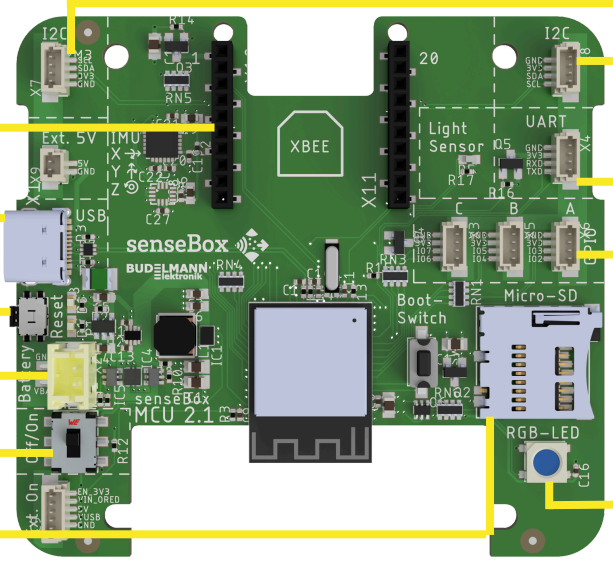
USB Port

Reset-Button

Battery Plug-in

On-/Off-Button

SD card slot



I2C/Wire-Ports

All green enviromental sensors and the display are conntected here.

UART/Serial-Ports

Sensors such as the fine particulate sensor are connected here.

Digital/Analog-Ports

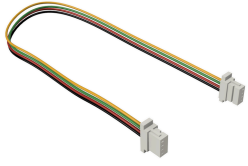
Sensors and actuators are conntected here via the breadboard.

RGB-LED

senseBox Cable

SB 02

There are three different types of senseBox cables that are compatible with the MCU S2:

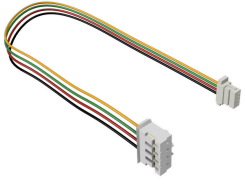


QWIIC cable

These cables are used to connect sensors and actuators directly to the MCU-S2.

JST-QWIIC cable

This cable is used to connect sensors with JST connectors to the I2C ports of the MCU-S2.



Charge controller cable

This cable connects the battery, including the holder, to the battery plug-in of the MCU-S2.

Digital/ Analog Ports

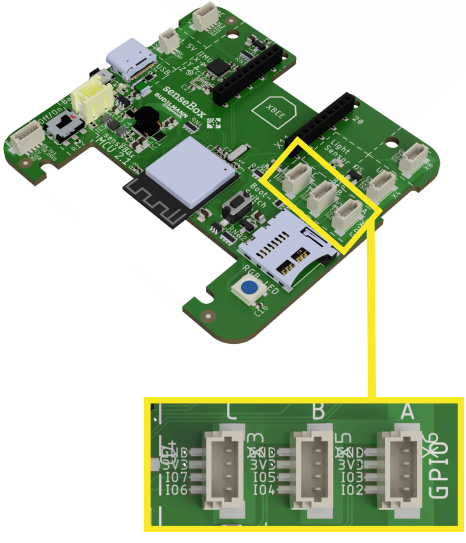
SB 03

With the plug-in cables you can connect sensors or actuators to digital/ analog ports.

Each digital/ analog port on the senseBox MCU has four different pins:

- The GND pin is the negative terminal and is always connected to the black cable
- The 5V pin is used for permanent power supply and is connected to the red cable
- The pins labeled 1 and 2 are the digital and analog pins 1 and 2, respectively This numbering continues up to pin 6 on port Digital C.

So that your programs can work correctly, you must select in some blocks the pin to which your consumer (e.g. an LED) is connected.



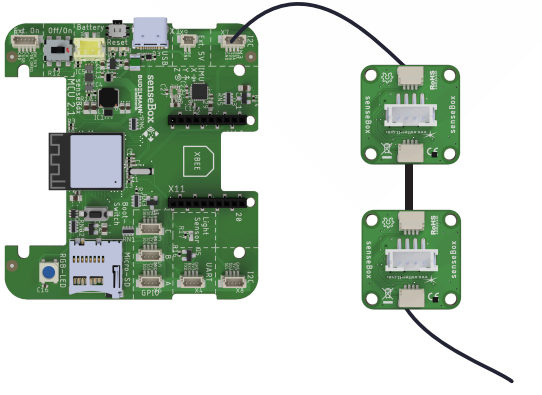
Connection to the I2C/Wire Ports

SB 04

If you want to use more than two components connected to the I2C/Wire ports, you have two options:

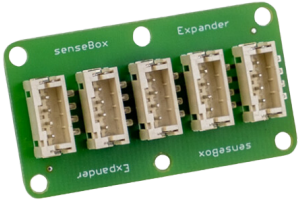
Option 1: Series connection

If you have connected a sensor to the I2C/Wire port of the MCU-S2, you can connect the next sensor to the previous sensor using a QWIIC cable. This is possible thanks to the dual QWIIC connectors on the sensors. It does not matter whether you use the input or output connector.



Option 2: I2C Expander

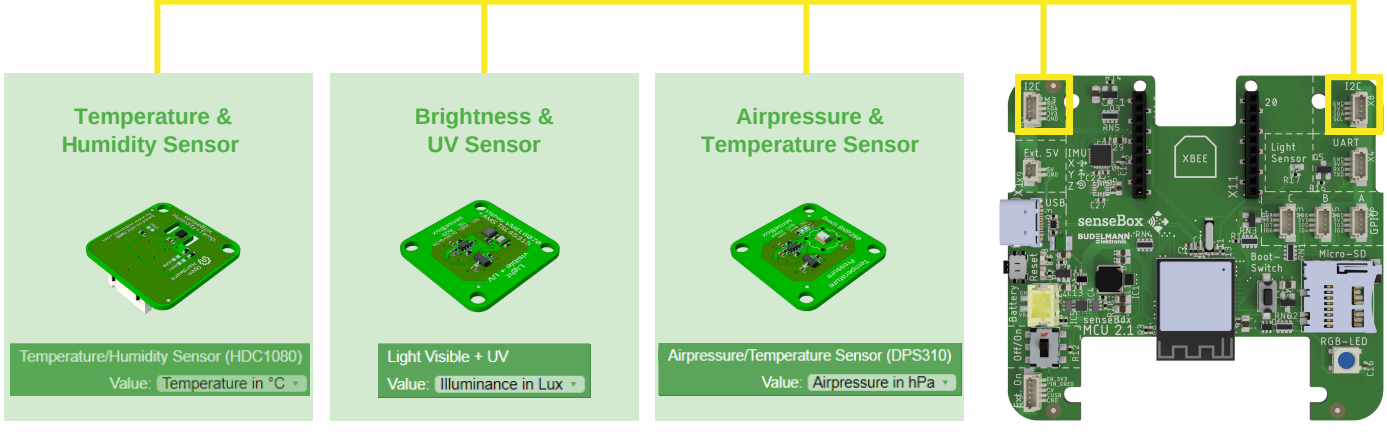
If you want to connect sensors with a JST connector to the MCU-S2, you must first connect an expander to the I2C/Wire port of the MCU-S2 using a JST-QWIIC cable. You can then connect up to 5 sensors with JST connectors to the expander, which would otherwise also be connected to one of the I2C/Wire ports.



The Green Environmental Sensors

SB 05

The green environmental sensors of the senseBox are connected to the I2C/Wire ports via a QWIIC cable.
The following blocks give you the values for the individual environmental phenomena:

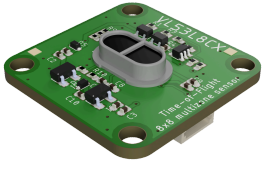


Note: If you are using more than two environmental sensors, you can connect several sensors in series as a chain. To do this, connect the first sensor to an I2C wire port using a QWIIC cable. Then, connect each additional sensor to the port of the previous sensor using another QWIIC cable.

The ToF Sensor

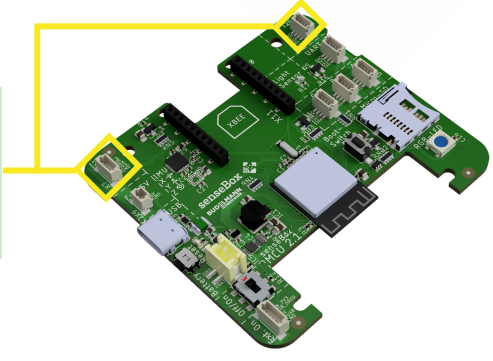
SB 06

You can measure distances using the ToF (Time of Flight) sensor. It is based on a camera with 8x8 resolution, which allows depth images to be generated and displayed on the LED matrix as required. The maximum measurable distance is 400 cm, but this varies depending on the lighting conditions.



CONNECTION

The ToF sensor is connected to one of the **I2C/Wire ports**.



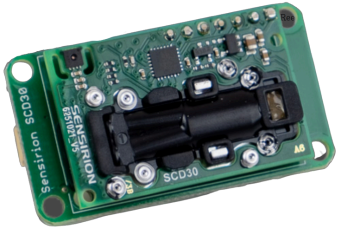
PROGRAMMING

Use this block in an endless loop to embed the ToF sensor in your program code. The connection and programming follow the same principle as for the environmental sensors. If you want to create a depth image on the LED matrix, select the ‘distances as bitmap’ option from the drop-down menu.

The CO2 Sensor

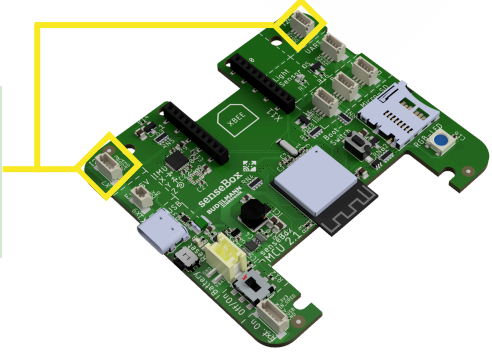
SB 07

With the CO_2 sensor, you can measure the CO_2 concentration in the room air. The measured value is output in parts per million (ppm). In addition to the CO_2 concentration, the temperature and humidity can also be measured. The CO_2 concentration is an important measurement value for indoor air quality. Indoors, the treshold of 1500ppm should not be exceeded for a long period of time.



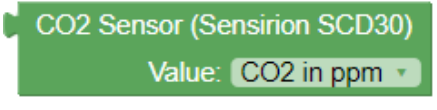
CONNECTION

The CO2 sensor is connected to one of the **I2C/Wire ports**.



PROGRAMMING

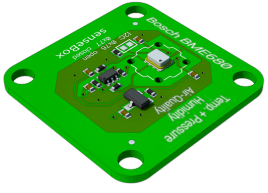
Use this block to read the CO_2 sensor. In the drop-down menu you can select which environmental phenomenon you want to collect. In this case the measured value of the temperature is the temperature in the sensor, which can be higher than the actual ambient temperature.



The Environmental Sensor BME680

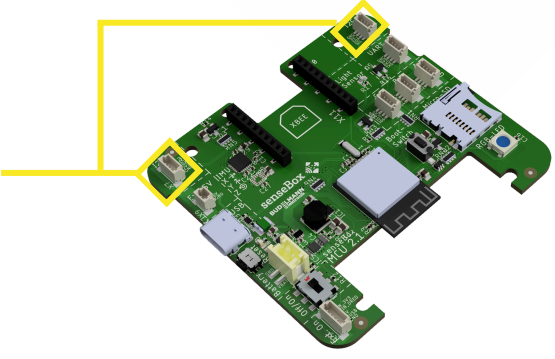
SB 08

With the environmental sensor BME 680 you can measure pressure, humidity, temperature and volatile gases. Note that you do not use the BME680 together with the BMP280 in the setup, because both sensors use the same I2C address, which can lead to conflicts.



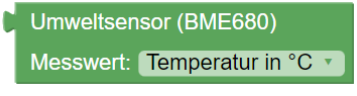
CONNECTION

The environmental sensor is connected to one of the **I2C/Wire ports**.



PROGRAMMING

Use this block to read out the environmental sensor. In the dropdown menu you can select which environmental phenomenon you want to collect.



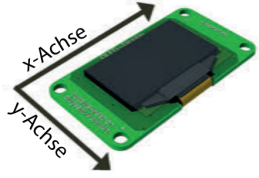
Note: The sensor requires a certain amount of time to calibrate. The status of the calibration can be read from the calibration value. It is either 0 (sensor is stabilized), 1 (value is inaccurate), 2 (sensor is calibrated) or 3 (sensor successfully calibrated).

The measured values for temperature, humidity and air pressure can be used directly.

The Display

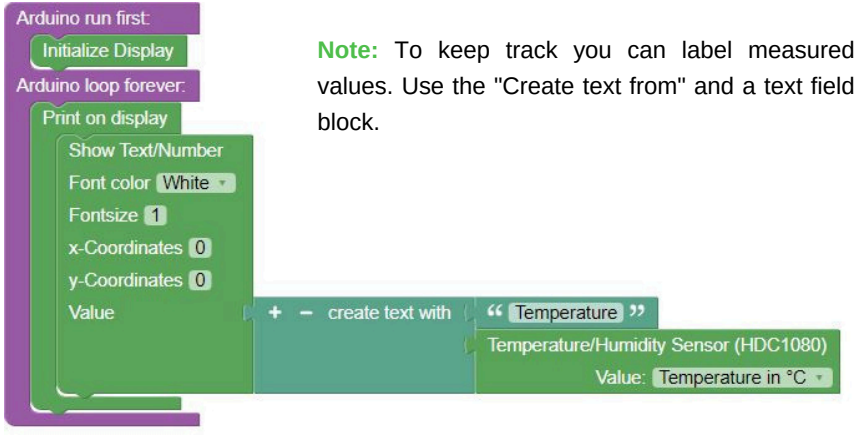
SB 09

The display can show you text, numbers and diagrams. The display has a resolution of 128x64 pixels. With the help of the x- & y-coordinates you can define where to write on the display.

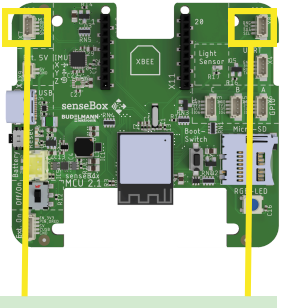


PROGRAMMING

For this, it must be initialized in the setup( ) and programmed in the infinite loop( ).



Note: To keep track you can label measured values. Use the "Create text from" and a text field block.



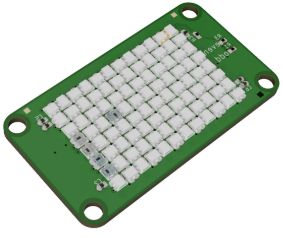
CONNECTION

The display is connected to one of the **I2C/Wire ports**.

The LED Matrix

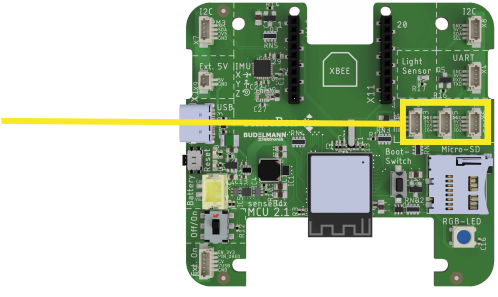
SB 10

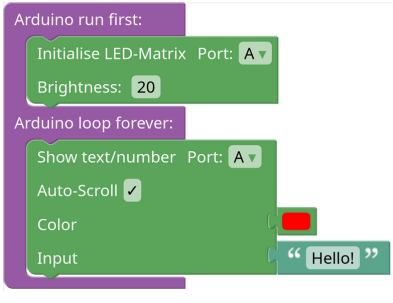
With the LED matrix, you can display text and shapes in different colors. It has a total of 8x12 pixels, and you can decide for each pixel whether it should light up and in which color.



CONNECTION

The LED matrix is connected to one of the **I2C/Wire ports**.





PROGRAMMING

Initialize the LED matrix in “Arduino run first” before using it in the infinite loop. Select the port used during setup. You can specify the font color and insert a text field for input.

Alternatively, you can display a bitmap with any motif.

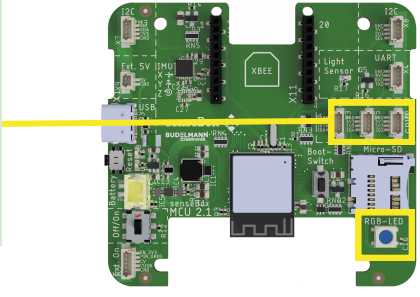
The RGB LED

SB 11

The RGB LED can display all colors. RGB stands for red, green, and blue. By combining these three colors, all other colors can be created. The color value ranges from 0 to 255.

CONNECTION

An RGB LED is installed on the board and does not need to be connected separately.
The modular RGB LED is connected to the digital/analog port via the input. If several RGB LEDs are to be connected in series, you can connect them to each other using additional cables (output to input). You can determine which LED is controlled by specifying the position in Blockly.



PROGRAMMING

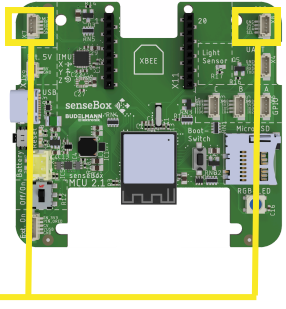
Initialize the RGB LED in “Arduino run first” before using it in the infinite loop. For the modular RGB LED, the connection port on the MCU S2 must be specified. The LED color can be selected either via the color picker or by setting individual color channel values (0–255).

The Particulate Matter Sensor (SPS30)

SB 12

With the fine dust sensor, you can measure the amount of the smallest dust particles in the air in four different particle sizes:

PM1.0: Specifies the amount of fine dust particles <1 µm in µg/m3
PM2.5: Specifies the amount of fine dust particles <2.5 µm in µg/m3
PM4.0: Specifies the amount of fine dust particles <4.0 µm in µg/m3
PM10: Specifies the amount of fine dust particles <10 µm in µg/m3



CONNECTION

The particulate matter sensor is connected to one of the two **I2C/Wire ports.**

PROGRAMMING

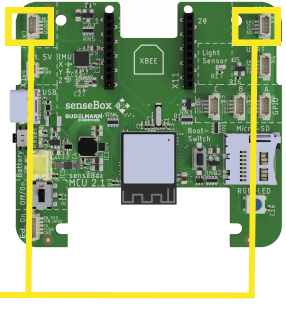
In the drop-down menus of the sensor block, you can select the desired measured value and the port to which the sensor is connected.

The GPS Module

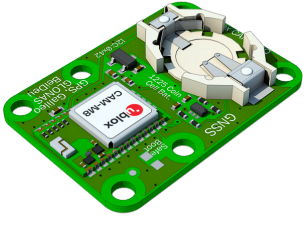
SB 13

With the GPS module you can get different location information.
It can output six different measured values:

Height above sea level in m	Longitude
Speed in km/h	Date
Latitude	Time



CONNECTION: The GPS module, like all green environmental sensors, is connected to one of the two **I2C/Wire ports**.



PROGRAMMING

Use this block to read the GPS module:

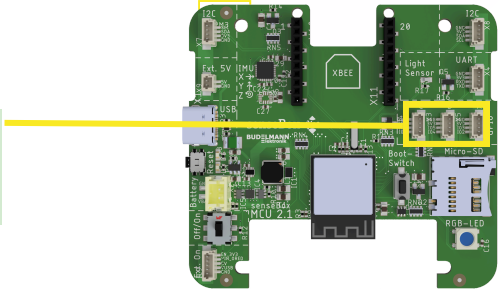
The Ground Sensor

SB 14

With the soil sensor you can measure two different soil parameters:

- Soil temperature in °C
- Soil moisture in %

Soil moisture is expressed in values from 0 to 50% volumetric water content.

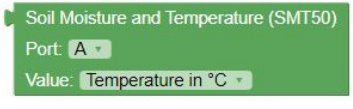


CONNECTION

The floor sensor must be connected to one of the **digital ports**.

PROGRAMMING

Use this block to display the measured values of the soil sensor:



In the dropdown menu you can select the desired measurement value and the port to which the sensor is connected.

Data Transfer to the openSenseMap

SB 15

ESTABLISH WIFI CONNECTION

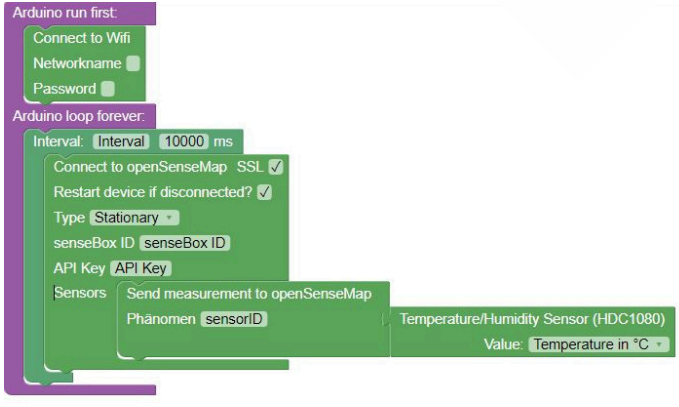
The WiFi-Bee is integrated on the MCU-S2 board and does not need to be connected separately.

PROGRAMMING

After that, you have to drag the "Connect to Wifi" block into the 'Arduino run first' and enter your network name (SSID) and WLAN password.

SEND TO THE OPENSENSEMAP

After registering your senseBox on openSenseMap, you will receive a BoxID and a SensorID for each sensor. Enter the BoxID in the “Connect to openSenseMap” block and the corresponding SensorID in the “Send measurement to openSenseMap” block. The measurement interval defines how often data is recorded and sent.



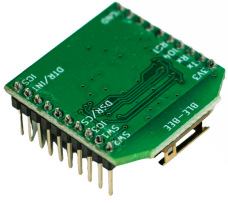
Data Transfer to the Phyphox App

SB 16

ESTABLISH BLUETOOTH CONNECTION

Connect the Bluetooth Bee to the XBee slot.

Then download the Phyphox app (https://phyphox.org/download/).



PROGRAMMING

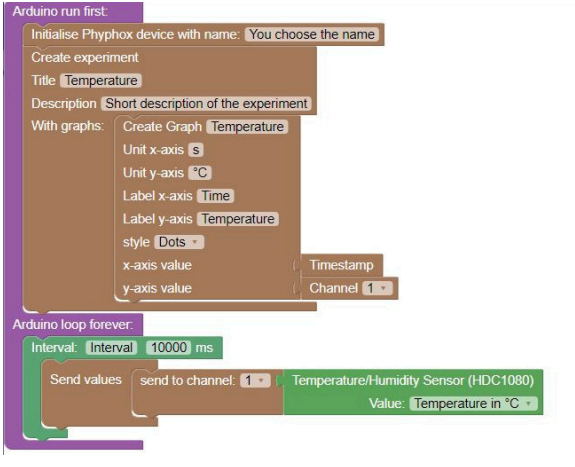
Initialize the Phyphox device in “Arduino run first” and create an experiment. There, you can define the basic settings for the generated graph.

SENDING THE MEASURED VALUES TO THE PHYPHOX APP

Set a measurement interval in the infinite loop and send each environmental phenomenon to a new channel.

PHYPHOX APP

Add a new Bluetooth measuring device in the Phyphox app using the + button, activate the automatic timer, and start the measurement. The measured values are then displayed in the app as a diagram.

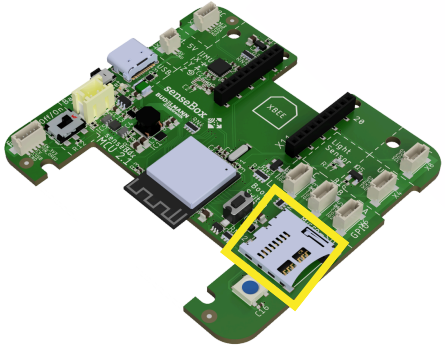


Data Storage at the SD-Card

SB 17

CREATE A FILE ON THE SD CARD

The SD card slot is located on the MCU-S2 board.

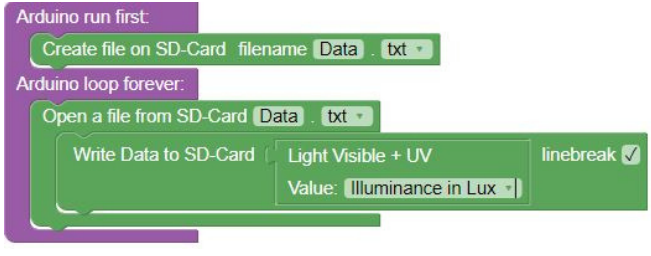


PROGRAMMING

In the 'Arduino run first', create a new file on you SD-Card with the block "Create a file on SD card".

WRITE MEASURED VALUES TO THE FILE

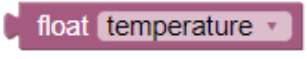
To save a measured value to a file, first open the file using the “Open File” block, then write the measured value using the “Write Data” block. The “Open File” block automatically closes the file after the data is written.



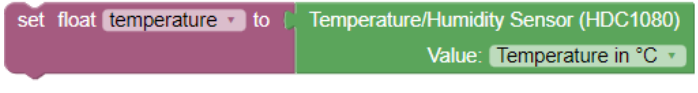
Variables– Placeholder

GI 01

Variables, also called placeholders, are used in computer science for various things. They are a kind of box that is given a name. In this box you can store different things (e.g. numbers and texts) and retrieve them later.



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



DATA TYPES

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

Characters (char):	For individual text characters
Text (string):	For whole words or sentences
Number (int):	For numbers from -32768 to +32768
Large number (long):	For numbers from -214748364 to +2147483648
Decimal number (float):	For decimal numbers (e.g. 25,3)
State (boolean):	true or false

If ... then – What?

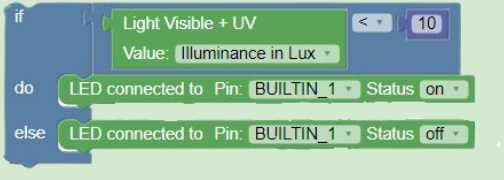
GI 02

The "If-then Condition" is one of the most important control structures in programming. With the help of the "If-Then" condition, the senseBox can perform certain actions when something specific (e.g. a button press) has happened.

With the "Logical comparison," you can compare two values. An explanation of the different symbols in this block can be found on the card GI03 „Operators“.

EXAMPLE

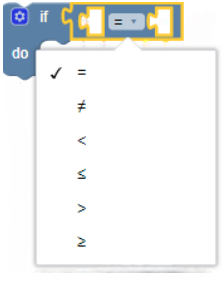
If the temperature is greater than 20°C,
do the built-in LED is to be switched on.
Else, the built-in LED should be switched off.



Operators

GI 03

Operators are needed in many programming situations. With the help of the operators, conditions can be checked or values can be compared.



You can find the following operators in Blockly:

= You can use this character to check whether two values are equal.

≠ You can use this character to check if two values are different.

< With the help of this character you can check if one value is smaller than another.

≤ This character is an extension of the "less than" character and includes values that are the same size.

> You can use this character to check whether one value is greater than another.

≥ This character is an extension of the " greater than" character and also includes values that are equal in size.