

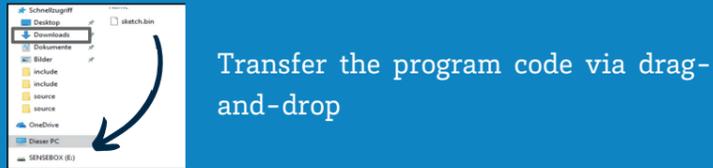
## Info: Uploading the program code



Compile



Turn on the senseBox



Transfer the program code via drag-and-drop

Alternative: senseBox Connect App  
Instructions for transferring the program code using a tablet:



## Environmental Measuring Station



Level: ★☆☆



A glance at the weather app on your smartphone tells you today's temperature, the probability of rain, and the UV index. But where does this data actually come from? Extensive measuring stations operated by the German Weather Service collect these measurements throughout Germany and you can too!  
In this project, you will build your own environmental measuring station.

## The code isn't working? Troubleshooting tips

- Are your cables plugged in exactly as shown in the illustration?
- Are your command blocks really connected like small "puzzle pieces"?
- Have you deleted all blocks that are not connected to your main block?
- Have you compiled the latest version of your program code and uploaded it again after making changes in Blockly?

Still having trouble?  
Then get in touch with a mentor!

## iCODE Coordinates

### INFO: COORDINATE SYSTEM IN COMPUTER SCIENCE

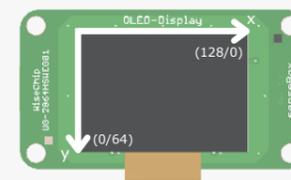
You will use the senseBox display to read the measurements from your environmental station.  
The display allows you to specify x and y coordinates for the position of the text. But be aware that the coordinate system of the display is "upside down"!



You probably know this coordinate system from math class, which has the point (0/0) at the bottom left.

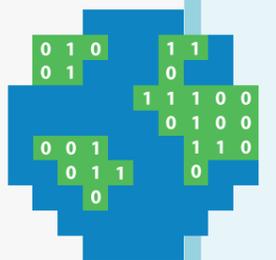


However, in computer science the coordinate systems start at the top left. The values then go downwards on the y-axis into positive territory.



Note: A letter in font size 1 is approximately 10 pixels high.

Here you can see how you can translate the coordinate system used in computer science to a display with 128x64 pixels.



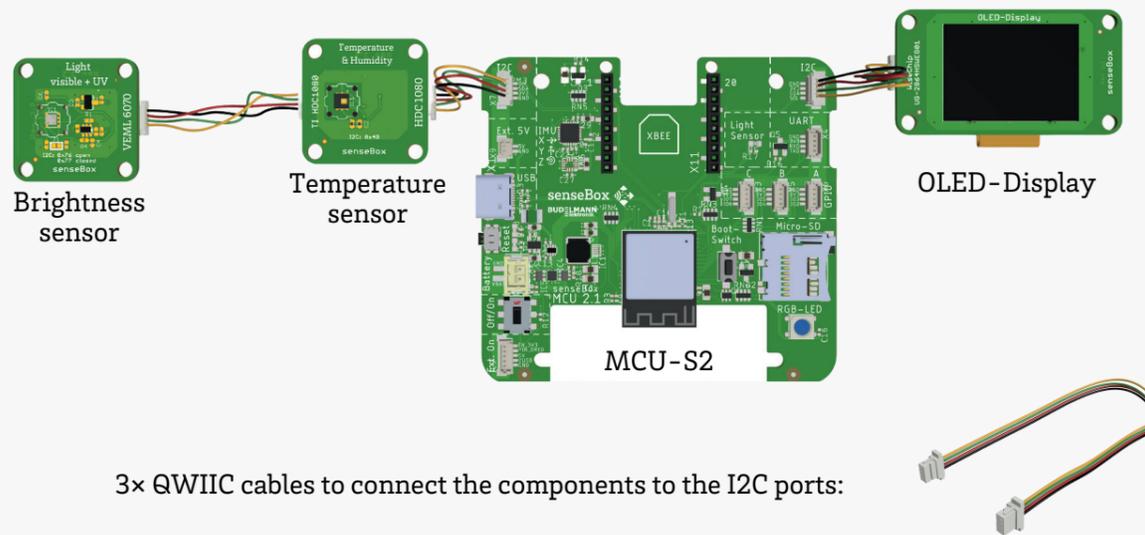


# Environmental Measuring Station

Connect the OLED display and the sensors to the microcontroller.

- 1) Create a program that shows temperature, humidity, brightness, and UV intensity measurements on the display.
- 2) Expand your program code so that the measurements shown include their name and unit on the display.

## Hardware-Setup



### Step 1A

1. For programming: [blockly.sensebox.de](http://blockly.sensebox.de)
2. In **Setup**, some components need to be activated once at the beginning of the program.

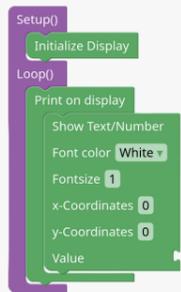


3. The **display** has to be **initialized** in the setup:

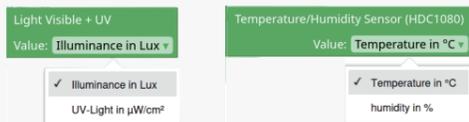
Initialize Display

### Step 1B

1. To display the measurements on the screen, you need the blocks **"print on display"** and **"show text/number"** in the loop.



2. Under **"Value"**, you can insert the block for the respective sensor. Some sensors can measure two environmental phenomena simultaneously.



3. To display all four values, you need the **"Show text/number"** block **four times**. You also need to adjust the **y-coordinate** so that the values are displayed one below the other. The **y-values** can for example differ in steps of 15.

Test your code!

### Step 2B

1. Due to the added text, not all measured values now fit on the display. To change this, **two measured values** should now be displayed alternately on the display for **5 seconds** each. To do this, you need the **"Wait 1000 milliseconds"** block. Adjust the time here (note: 1000 milliseconds = 1 second).

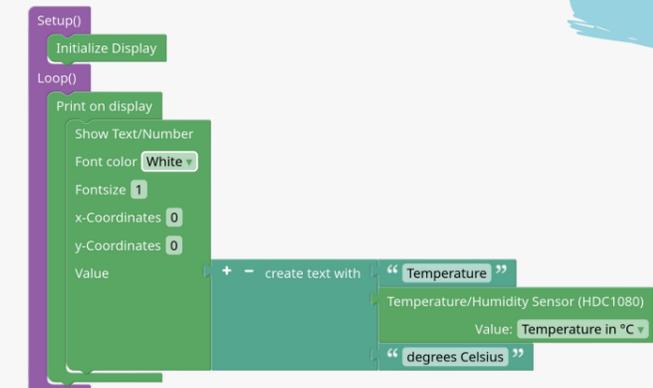


2. To clear the display after **5 seconds** and show the next two measurements, you need the **"Clear display"** block. Then use the same blocks repeatedly to show the other two measurements (**brightness & UV intensity**). Also adjust the position of the text on the display by changing the **y-coordinate**.



### Step 2A

To label the four different measurements, you now need the block **"Create text from"** from the **"Text"** category. Here you can add another empty space using the **"+"** sign. You will also need **two empty text blocks** from the **"Text"** category, in which you can write any text you like, e.g., the **name** (temperature) and **unit** (degrees Celsius) of the respective measured value. You will need to repeat this step **four times** as well.



Test your code!