Table of Contents

First Steps

Components

Help & More

The senseBox:home Book

In this book you learn how to built and code your senseBox:home. Have fun!

About the book:

In this book you learn how to built and programm your senseBox:home. Additionally, you learn a lot about how the measurement station, how the data is transmitted and how to integrate the data on openSenseMap.

Before you start, please read this introduction. It will help you to understand the book and gives you insights into the structure of the book.

Introduction

The senseBox:home book is mainly done for owners of the senseBox:home, but also for all interested people which want to get in contact with senseBox. This senseBox:home is a DIY project, this means you decide on your own how you senseBox will look like and which data will be collected.You always have the option to adapt more sensors afterwards to your senseBox to measure fine dust, UV and illumination, air preasure or even NOx. All this data you can be upload on openSenseMap[1] to share them with all other senseBox owners as "open data" and to observe and interpret the data history of your senseBox.

This book will help you first to do the Erste Schritte[1] with your senseBox:home. Afterwards, it explains you the several components and their features. Finally, you find some information about common issues and frequently asked questions.

On your way through the book you will be guided with the side bar on the left side and the green arrow buttons. Each chapter has a small introduction in a grey boy with green letters about the content of the page at the beginning.

This will look like this:

Headline

This is the content which will be in the chapter.

Moreover we added some notice boxes for specific content. There are four types of boxes:

> This type of notice shows you Information such as links to additional information sources or advanced background knowledge.

> This type of notice contains a checklist. The checklist is typically a summary about the things you should have done in the chapter. With the list you can check if you did not forget any important steps.

> This type of notice is a warning sign. It guides your attention at situations in the book in which you have to work very carefully.

> This type of notice informs you about typical errors. It shows you for example what you can do if you already did a mistake.

In extension we offer some passages which are hidden. They contain information about further topics and examples. If you click on this passage it will drop out. Here is an example for a hidden passage:

▼ **Ein versteckter Absatz**

Nice, you got it! Click again on the gread headline to hide this passage again.

The last element in this book which you should familiar with are "tabs". You will know them from your internet browser to open more than one websites at the same time. In this book we use tabs to reduce the content of the page and let you only see what you really need. An example is that we use them to display differences between the operating systems (Windows, OSX(Mac) and Linux). In the most cases you will only be interested in one of the operating systems, so just click on the tab with the system you use. Just try it in the following example:

- First Tab
- Second Tab
- Third Tab

Erster Tab

The first tab is opened initially.

Zweiter Tab

You chose the second tab.

Dritter Tab

You got it! You opended the third tab.

You checked out the tabs and all the other elements of the book?

Then you can start now with your senseBox:home!

We wish you a lot of fun!

Your senseBox[2] Team!

---

1. https://opensensemap.org/ ↩

1. See *2.1 Step 1: Software Installation* ↩

2

> [2]. https://sensebox.de/ ↩

Overview

In this book, we'll show you what you can do with your senseBox, and explain everything you need to know to build a ready weather station. We also give you tips and suggestions for further experiments.

So that everything runs smoothly we start from the very beginning and lead you step by step through the installation of software and hardware up to the construction of your operational measuring station. First of all, you get an overview of how this book is structured.

1. Getting Started - Install needed programs, test your sensors and start with your senseBox
    i. Installation of required software
    ii. Installing the Board Support Packages and SenseBox Libraries
    iii. Connection and Wiring
    iv. Test components
    v. Encryption
    vi. Registration on the openSenseMap
    vii. Assembling in the housing
    viii. Setting up the senseBox
2. Components - Look at all parts of the senseBox and their functions
    i. Contents
        i. senseBox MCU
        ii. Bees
        iii. Sensors
        iv. Accessories
3. Help - Frequently asked questions, answers and help with problems
    i. FAQ
    ii. Add External Libraries
    iii. Firmware Update Wifi-Bee
    iv. Update Windows USB Bootloader Driver
    v. Updating the Board Support Package & senseBox Libraries
    vi. Contributing

Inventory

Here you will find a overview what you can find in your senseBox:home. Please note that not all senseBox:home have the same setup. However, there is always a basis setup with the microcontroller and some parts to setup the box, but there can be different sensors, bees and more additional equipment entsprechend deinen Bedürfnissen.

Basis-Setup

The senseBox:home has a Basis-Setup which comes with each senseBox and consist of the following components:

- senseBox MCU[1]
- Temperatur- & relative Luftfeuchtesensor[2]
- Strahlenschutz[3]
- Outdoor-Gehäuse[4]
- Netzteil und USB-Kabel (3m)[5]

More components which can differ

We wanted to create the senseBox as flexible and customized as possible. Therefore sensors and the way to upload data (we call them "bees" can differ depending on you reseller. Check here which components exist.

Bees

Bees are the interface of your senseBox. Thanks to them you can upload, save and transmit data. You need minimum one bee to have a fully functional senseBox.

Currently there are four differnt bees. However, you can put only two Bees at the same time on your microcontroller. Click on the name of a bee to learn more about it:

- Wifi-Bee[6]
- LAN-Bee[7]
- SD-Bee[8]
- LoRa-Bee[9]

Sensors

The sensors are the heart of your senseBox. Depending on what you want to measure you can add and choose the sensors for your senseBox. Click on the name of a sensor to learn more about it:

- Luftdruck- & Temperatursensor[10]
- Beleuchtungsstärke- & UV-Strahlungssensor[11]
- Feinstaubsensor(PM10 & PM2.5)[12]
- Temperatur & Luftfeuchte[13]

Additional Components

There are some more components you can buy for your senseBox. Check them out by clicking on the names.

- GPS-Modul[14]
- LED-Display[15]
- HUB[16]
- Micro-SD Karte[17]

Currently this is the set of components we offer. However we constantly increase the set of components we offer and try to fullfill all the wishes of the community. The senseBos is a open source microcontroller which you can extend with many more sensors. Feel free to adapt your own sensors on the senseBox if you feel able to make it on your own. Please understand that we cannon help you in that case with additional support. However, you might find help in the senseBox community[1]

But let us start now with some action! Have fun with your senseBox:home!

---

1. See *3.1.1 senseBox MCU* ↩
2. See *3.1.3.1 Temperature- and Air Humidity Sensor (HDC1080)* ↩
3. See *3.1.4.1 Radiation Protection* ↩
4. See *3.1.4.2 Housing* ↩
5. See *3.1.4.3 Power Supply and USB-Cable* ↩
6. See *3.1.2.1 Wifi-Bee* ↩
7. See *3.1.2.2 LAN-Bee* ↩
8. See *3.1.2.3 mSD-Bee* ↩
9. See *3.1.2.4 LoRa-Bee* ↩
10. See *3.1.3.2 Airpressure- and Temperature Sensor* ↩
11. See *3.1.3.3 Illumination and UV-Radiation Sensor* ↩
12. See *3.1.3.4 Fine Dust Sensor* ↩
13. See *3.1.3.1 Temperature- and Air Humidity Sensor (HDC1080)* ↩
14. See *3.1.4.7 GPS* ↩
15. See *3.1.4.4 LED-Display* ↩
16. See *3.1.4.5 Expander* ↩
17. See *3.1.4.6 Micro-SD Karte* ↩
1. https://forum.sensebox.de/ ↩

Step 1: Software Installation

Before you start measuring phenomena, you have to consider some things like the installation of different drivers and software. No worries, it is an Open-Source-Software so you don't have to pay anything. However you should read the steps 1 to 8 carefully so there will not be any problems later.

Installation of Arduino IDE

Before the senseBox can be activated, you have to install a driver as well as a software on your computer. Furthermore it is advisable to conduct a test run to check if all the sensors operate correct to ensure the communication with the internet without any problems.

Please have a look at the instructions for your operation systems and follow the indicated steps.

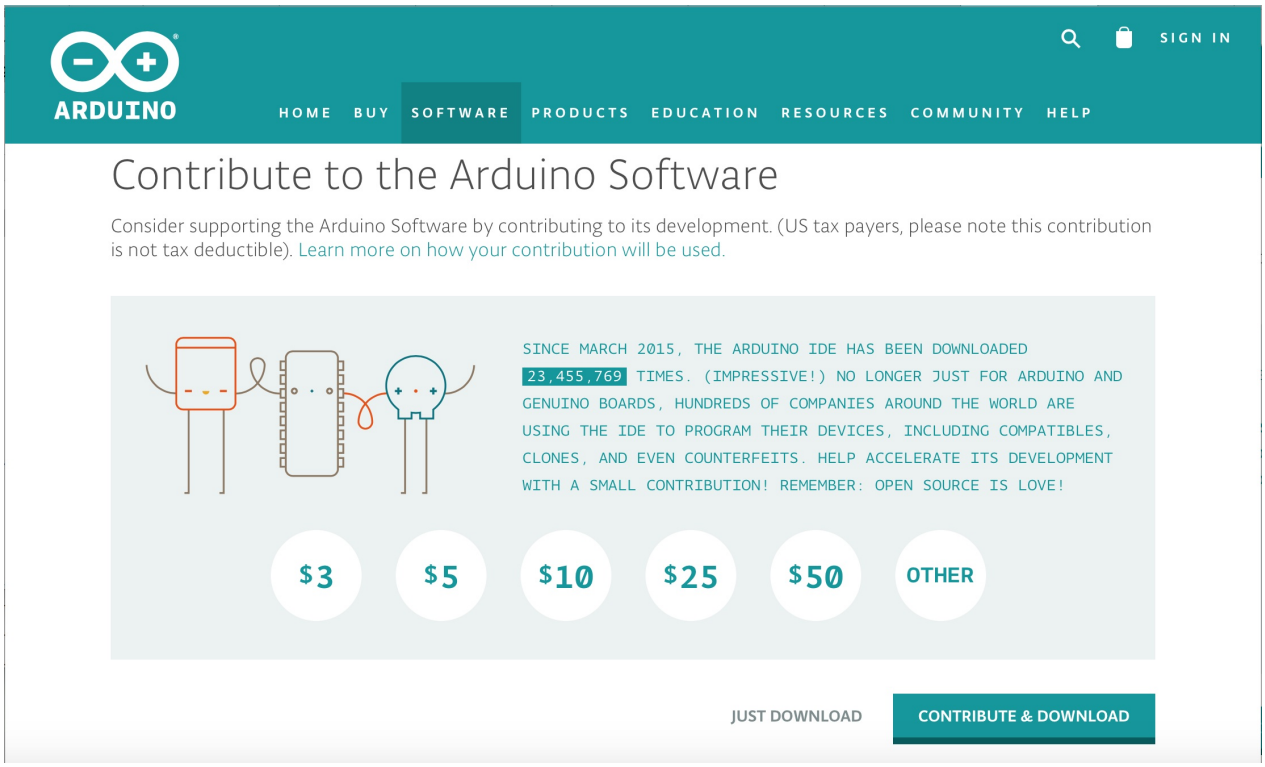
- Windows
- Mac(OSX)
- Linux

Download of the Arduino Software for Windows

> Please use Arduino 1.8.5 or higher for a fluent procedure.

The senseBox is a microcontroller with diffent components and sensors. It is going to be programmed with the development environment Arduino IDE. Please download the latest version as a zip-file from the Arduino Homepage[321]

Arduino is an Open-Source project and is financed by donations. Thus you are going to be asked for a donation; you can skip by clicking on `JUST DOWNLOAD`.



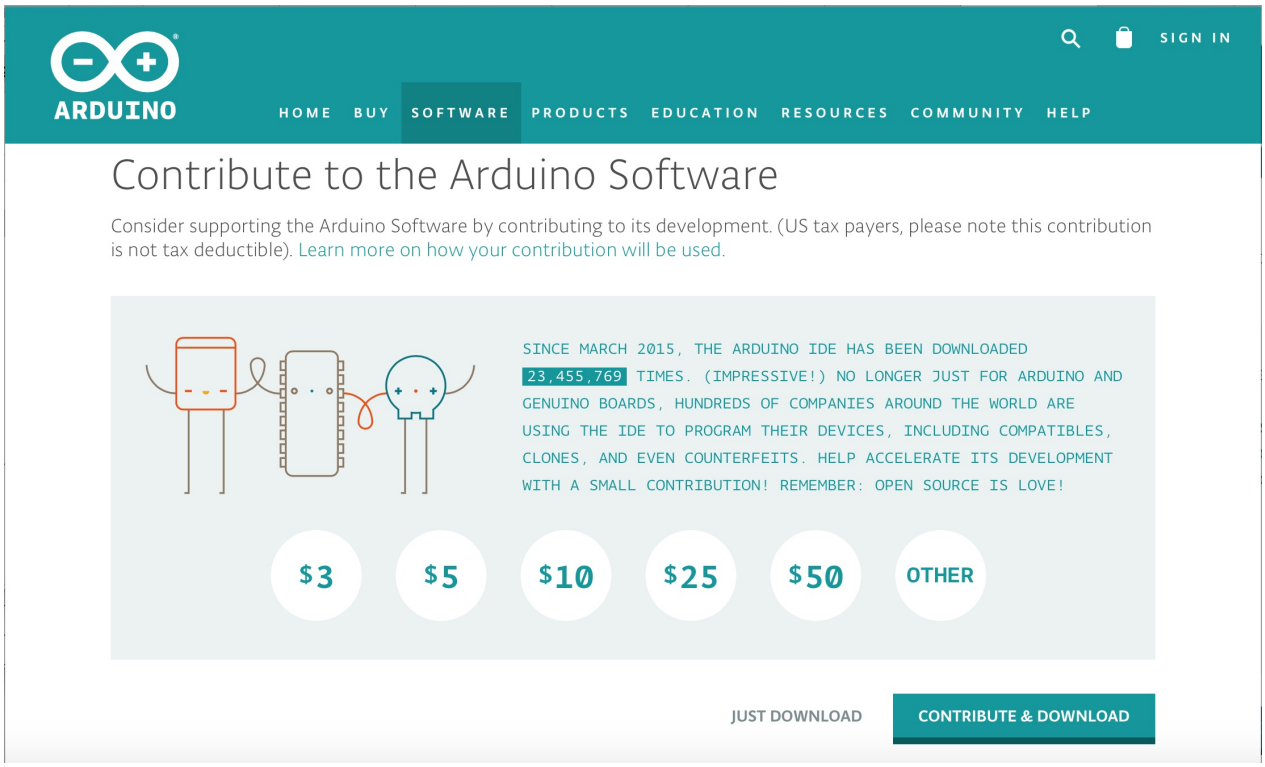
Compile a new folder and unzip the zip-file in the new folder. By starting the file `arduino.exe` the IDE can get started.

Download the Arduino Software for Mac(OSX)

> Please use Arduino 1.8.5 or higher for a fluent procedure.

The senseBox is a microcontroller with diffent components and sensors. It is going to be programmed with the development environment Arduino IDE. Please download the latest version as a zip-file from the Arduino Homepage

Arduino is an Open-Source project and is financed by donations. Thus you are going to be asked for a donation; you can skip by clicking on `JUST DOWNLOAD` .



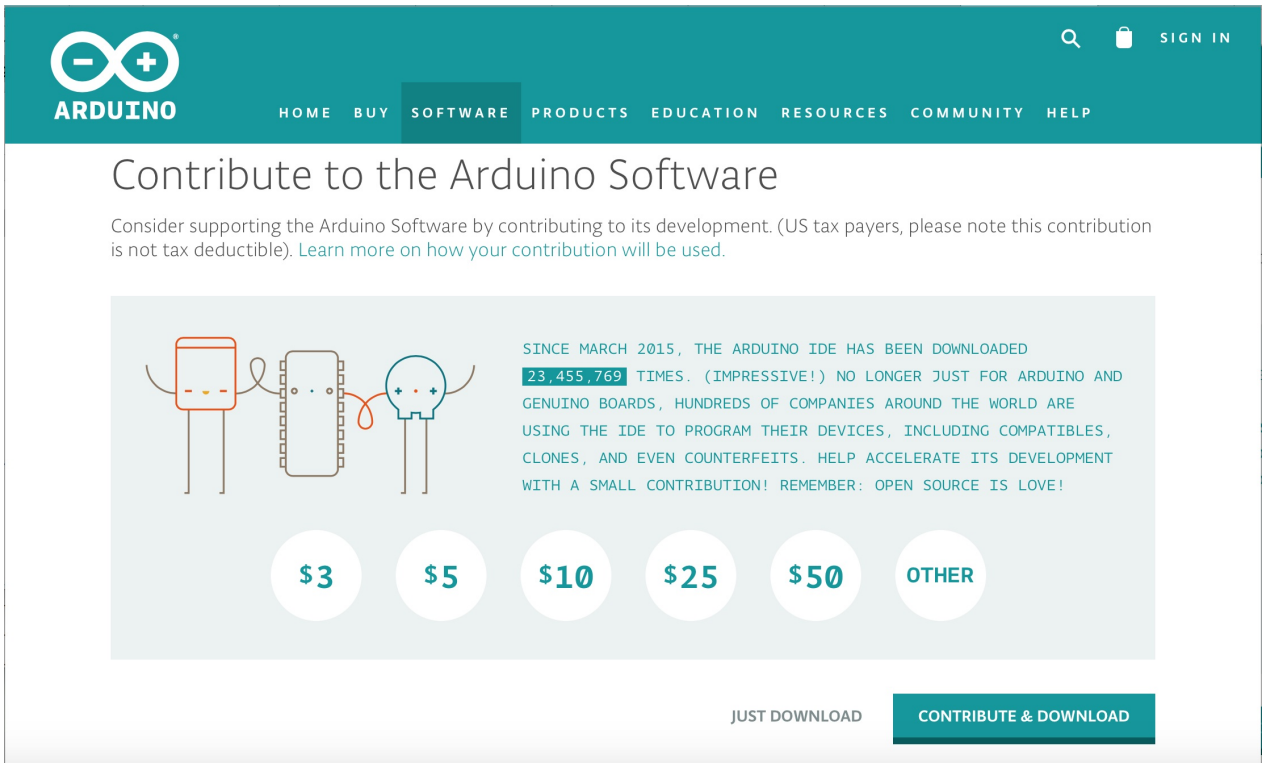
In your download file should appear a Arduino.app file. Relocate the file into your "Programme"-folder. By starting the file `Arduino.app` you can run the IDE.

Download the Arduino Software for Linux

> Please use Arduino 1.8.5 or higher for a fluent procedure.

The senseBox is a microcontroller with diffent components and sensors. It is going to be programmed with the development environment Arduino IDE. Please download the latest version as a zip-file from the Arduino Homepage

Arduino is an Open-Source project and is financed by donations. Thus you are going to be asked for a donation; you can skip by clicking on `JUST DOWNLOAD`.



Installation of the IDE with Linux

Linux-user can download and unzip a Linux version. The contained `install.sh` -script compiles a desktop shortcut automatically. The fastes way is to use the terminal. Open the Terminal therefor by hitting the keys `Ctrl + Alt + T` and enter the following commands:

```
# if the downloaded file is not stored in the download-folder, replace "downloads" by the path to the according folde
r cd downloads
```

```
# unzip the file with the following command and install Arduino
tar -xvf arduino-1.8.5-linux64.tar.xz
cd arduino-1.8.5
./install.sh
```

> Watch out, that the command is coordinated to the downloaded Arduinoversion! If you are downloading for example Arduino 1.8.6 everywhere where arduino-1.8.5 stands arduino-1.8.6 has to stand to. Watch the name of the download-file to check which version you have downloaded.

To programme Arduino ther are additional rights necessary at Ubuntu 14 & 16. Those can be established by the following commands (benötigt Admin-Rechte):

Implement `udevadm monitor --udev` and connect Arduino by USB to determine the Device-ID. The indicated label at the end of the output (zB. `ttyUSB0`) is the device-ID. Finish `udevadm` with `ctrl+C` , and run the following command whereby the found out Device-ID has to be implemented:

```
sudo usermod -a -G dialout $(whoami)
sudo chmod a+rw /dev/<device-id>
```

After a logout and again a Login the Arduino should be programmable out of the Arduino IDE!

---

1. https://www.arduino.cc/en/main/software ↩
2. https://www.arduino.cc/en/main/software ↩
3. https://www.arduino.cc/en/main/software ↩

Step 2: Installation of the Board-Support-Package

Therewith the Arduino IDE supports your senseBox MCU and you are able to relay the programme on to it, you have to start with installing two board-support-packages. Those contain the necessary drivers and software to communicate with the processor. The board-support-package of the senseBox contains already our senseBox-libraries. This will give you all the basic methods for programming the enclosed sensors.

Libraries

For the programming of the senseBox, the senseBox libraries should be included at the beginning. We have integrated those libraries into the senseBox board-support-package to make the inatallation as easy as possible for you.

▼ 'Library' - What is it and why do I need it?

As the name suggests, a library is a collection of something - a collection of methods to be more specific. Methods are programming smaller sections of code that can be applied to an object. For example, with the senseBox, a method can be invoked to turn the LEDs on and off on the MCU. There are a lot of such standard methods that are used by a variety of programs. In order not to have to transfer these methods individually into the program code, they can be stored in libraries. So a library is a file that stores many methods. You can include libraries in your code. For this it is enough, if they are stored in the Arduino folder for libraries and then they are integrated with a single line at the beginning of the program code. This looks like this in Arduino for the library named "senseBoxIO"

```
#include <senseBoxMCU.h>;
```

If the library is included, all methods contained in it can be used in the code.

Integrate the Board-Support-Package

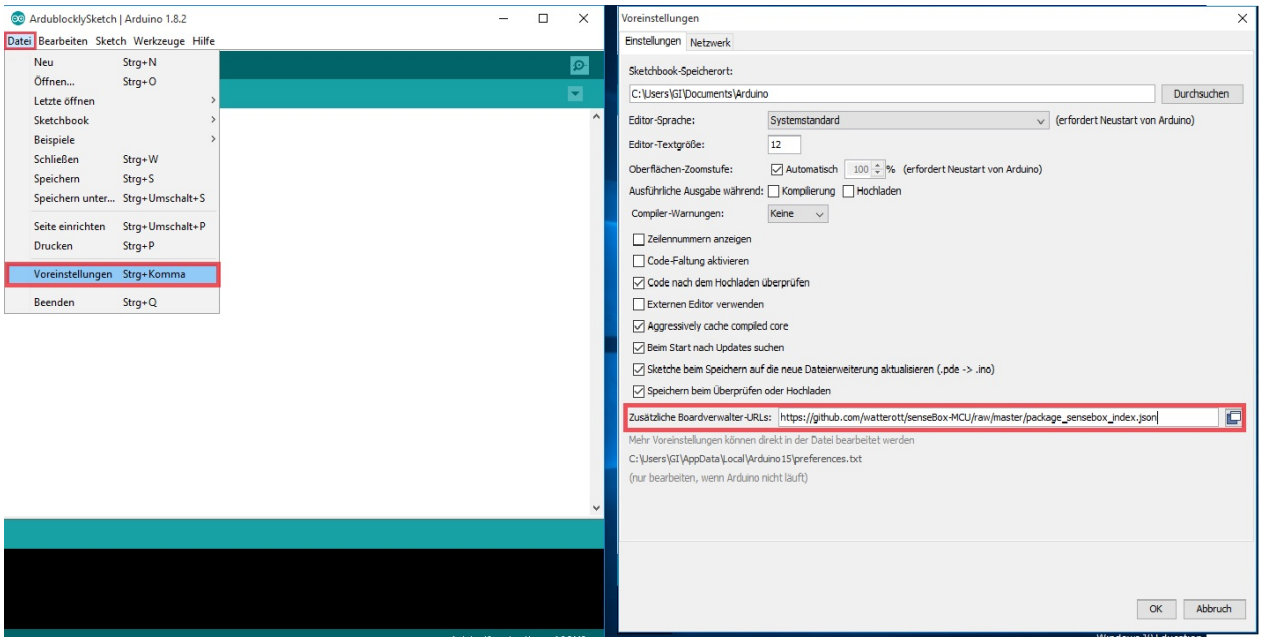
Choose your operating system to see the fitting instruction:

- Windows
- Mac(OSX)
- Linux

Instruction for Windows

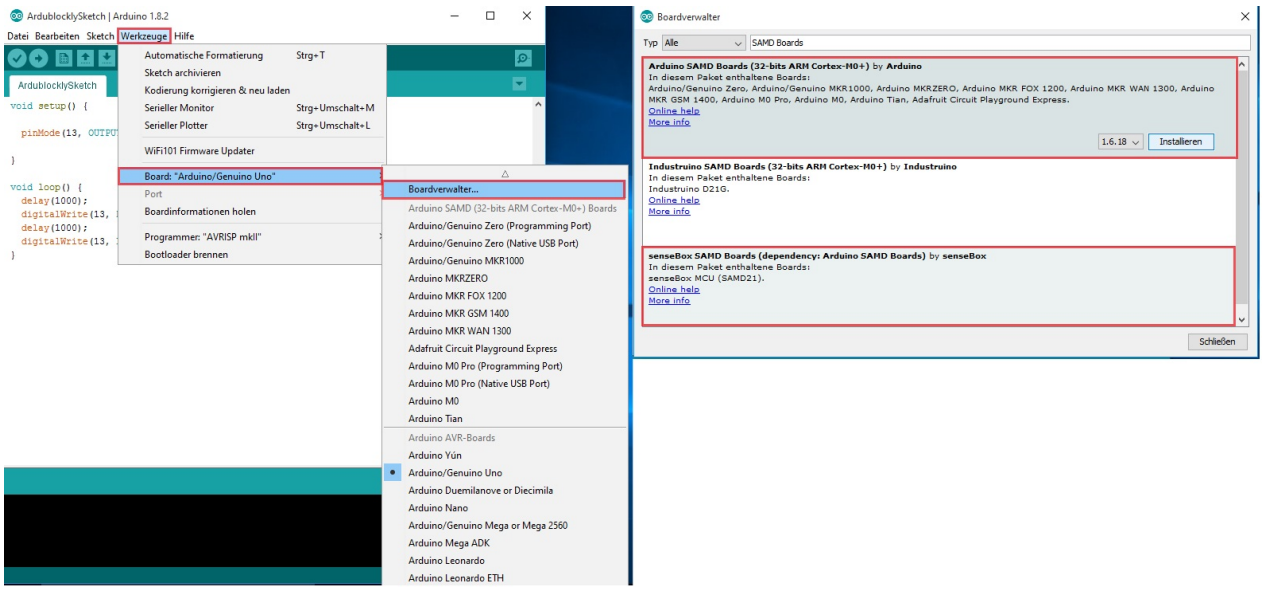
Paste the following URL into your Arduino IDE under Data -> Preferences in the field for Additional Boardmanager-URLs :

```
https://github.com/sensebox/senseBoxMCU-core/raw/master/package_sensebox_index.json
```



Open the Preferences and paste the URL

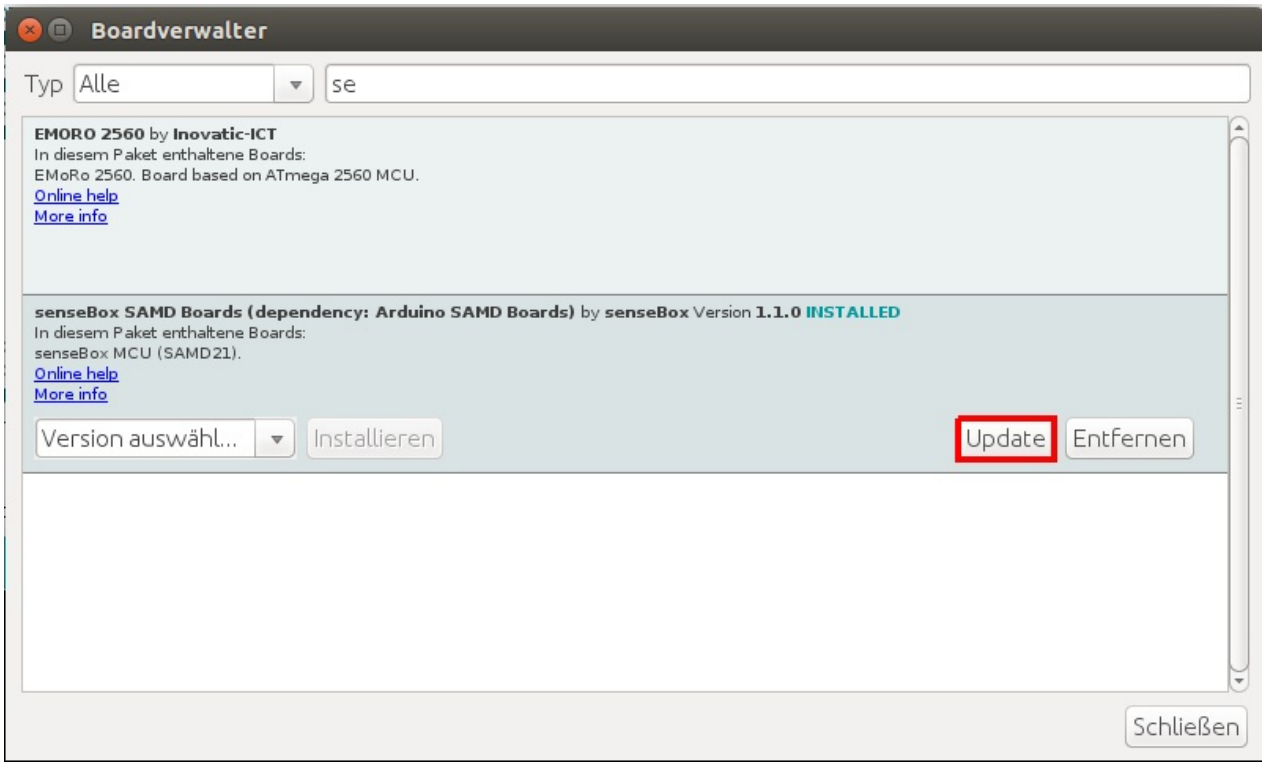
Please open then the Boardmanager under Tools -> Board:"..." -> Boardmanager and install there the two board-support-packages with the name Arduino SAMD Boards by Arduino and senseBox SAMD Boards by senseBox.



Open the Boardmanager and install both Packages

> Enter "SAMD" in the search bar above to find the packages faster

Because we are updating the senseBox SAMD Boards-package regularly for you, you should from time to time have a look into the boardmanager to check if the senseBox SAMD Boards-package still is the latest. Therefore please open, like described above, the boardmanager and search for senseBox SAMD Boards. If you click at the entry in the list, there will pop up an update-Button in case that a new version is available. Click the button to install the latest version.



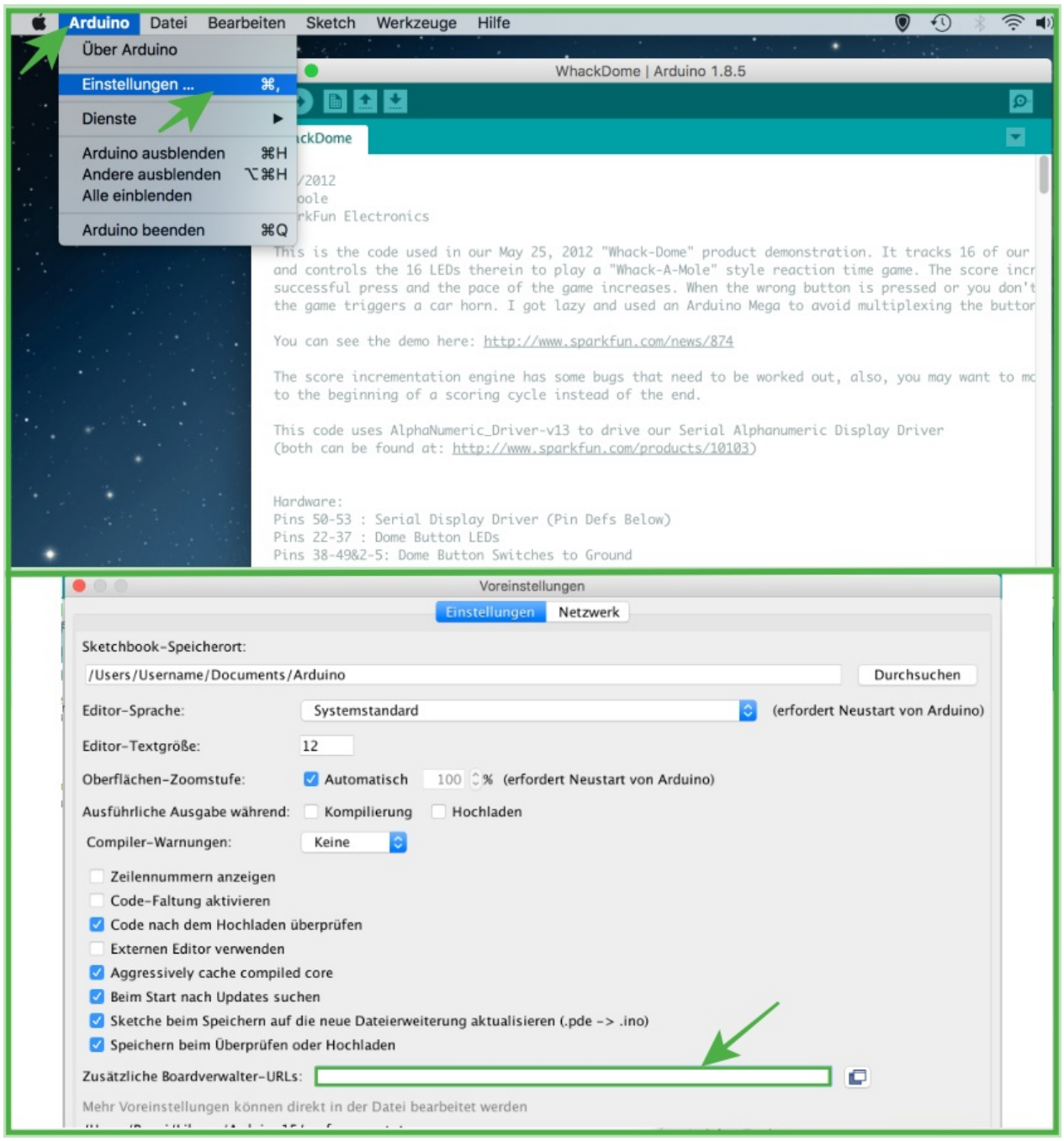
Click on 'Update', to refresh the Board-Support-Package

> It is important to first click on the entry. Otherwise the update button will not show up even if there is a new version.

Instruction fo Mac

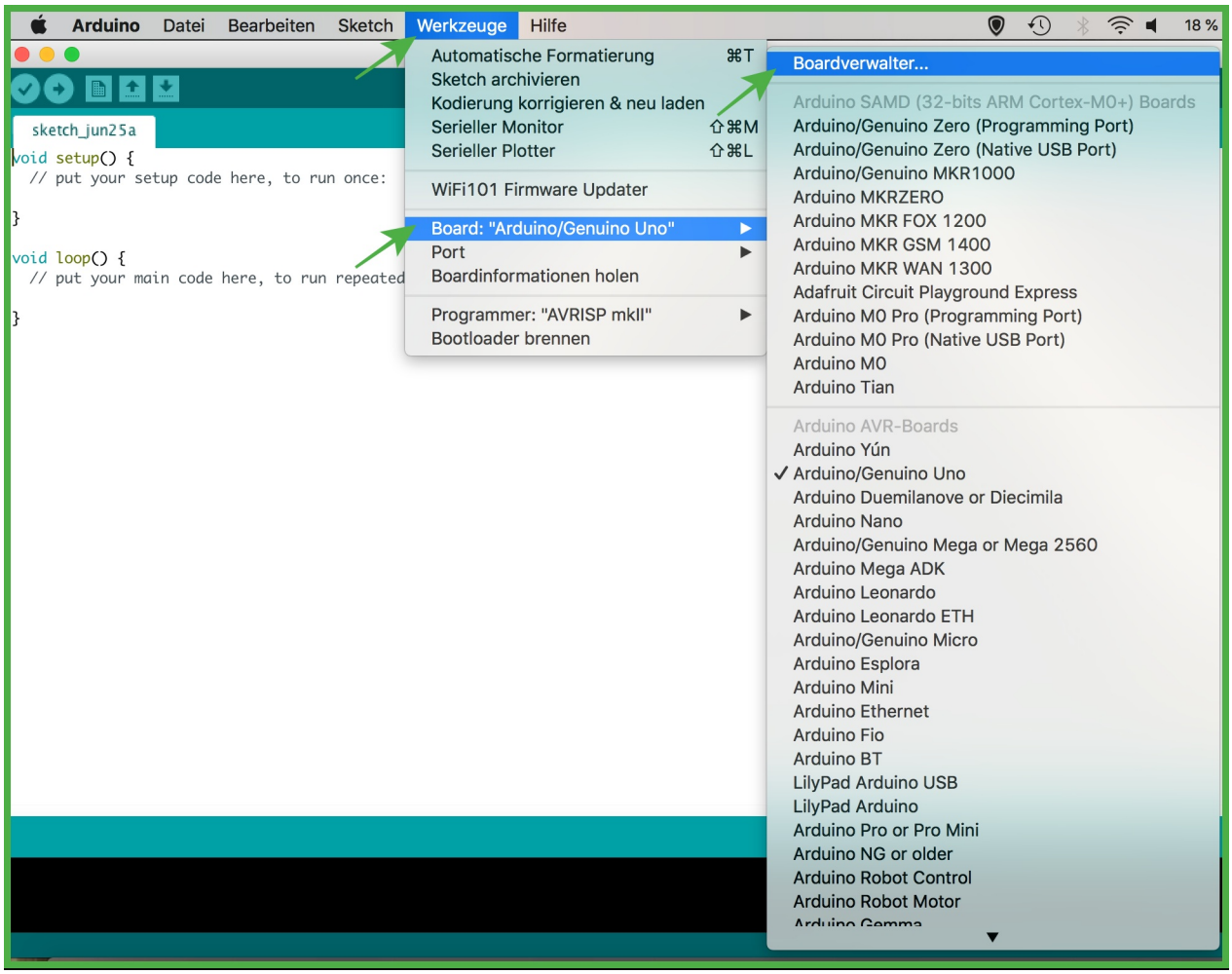
Please paste the following URL into your Arduino IDE under `Arduino -> Einstellungen...` in the field for additional Bordmanager-URLs:

```
https://github.com/sensebox/senseBoxMCU-core/raw/master/package_sensebox_index.json
```



Open the Preferences and paste the URL

Please open then the Boardmanager under Tools -> Board:"..." -> Boardmanager and install there the two board-support-packages with the name Arduino SAMD Boards by Arduino and senseBox SAMD Boards by senseBox.

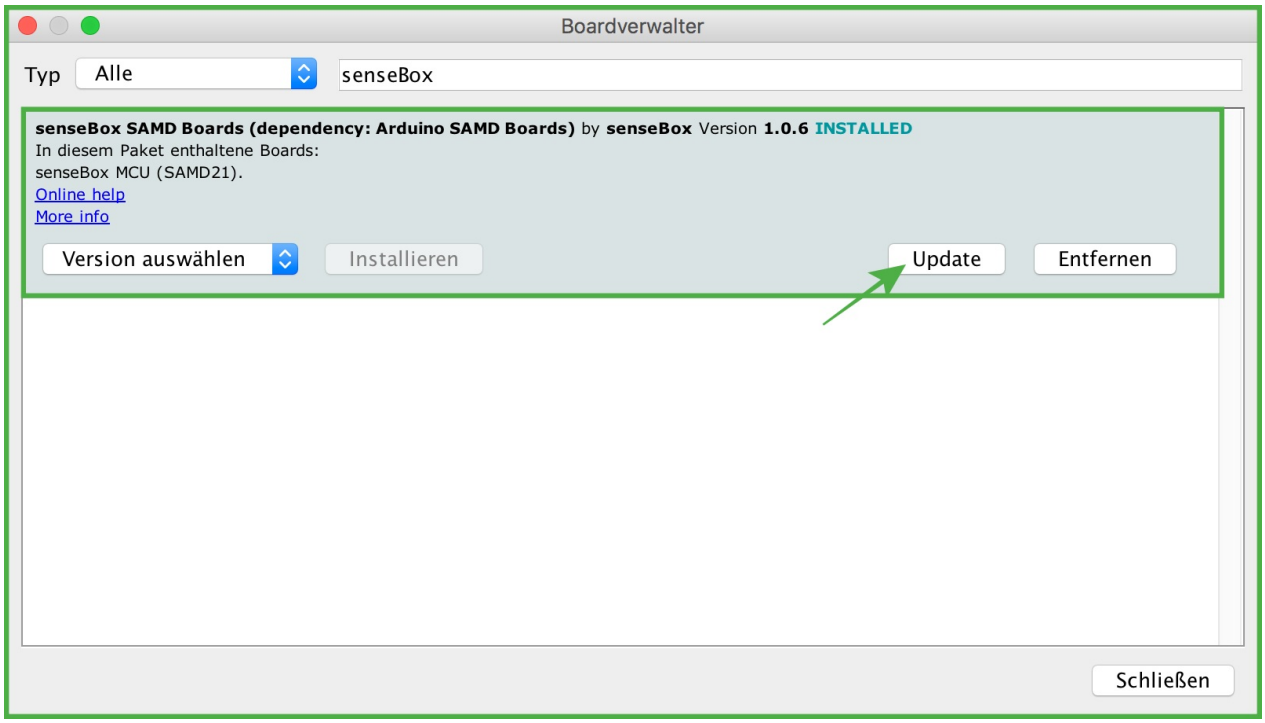


Open the Bordmanager

Install the two marked Packages

> Enter "SAMD" in the search bar above to find the packages faster

Because we are updating the senseBox SAMD Boards-package regularly for you, you should from time to time have a look into the boardmanager to check if the senseBox SAMD Boards-package still is the latest. Therefore please open, like described above, the boardmanager and search for senseBox SAMD Boards. If you click at the entry in the list, there will pop up an update-Button in case that a new version is available. Click the button to install the latest version.



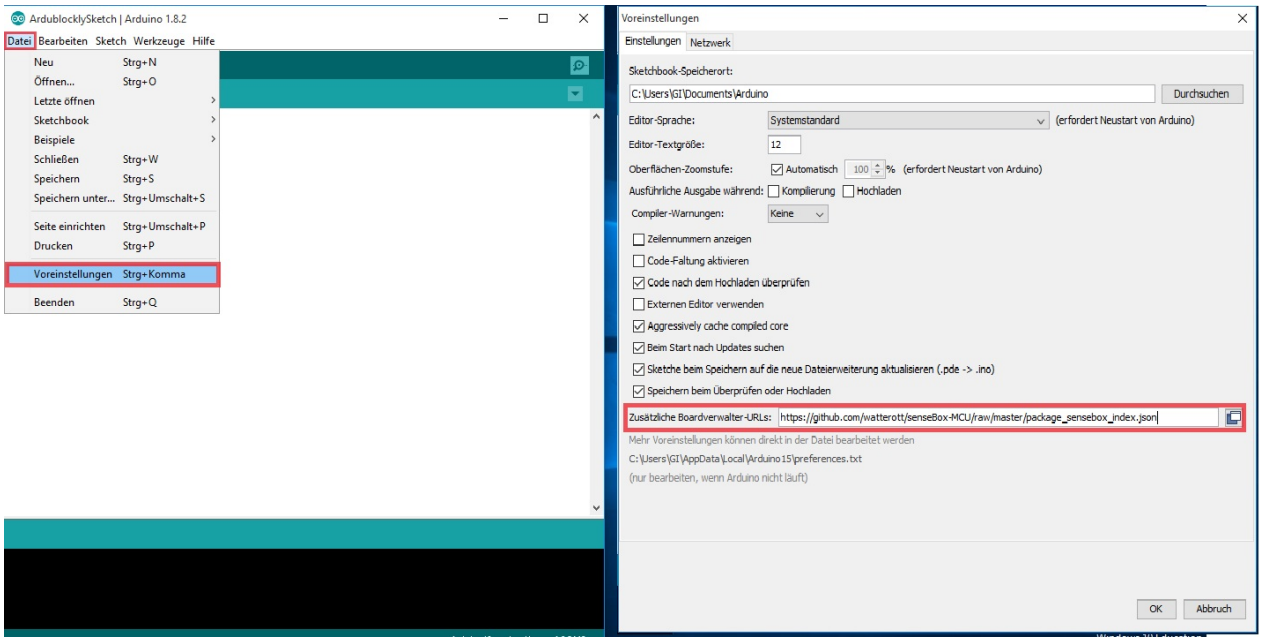
Click on 'Update', to refresh the Board-Support-Package

> It is important to first click on the entry. Otherwise the update button will not show up even if there is a new version.

Anleitung für Linux

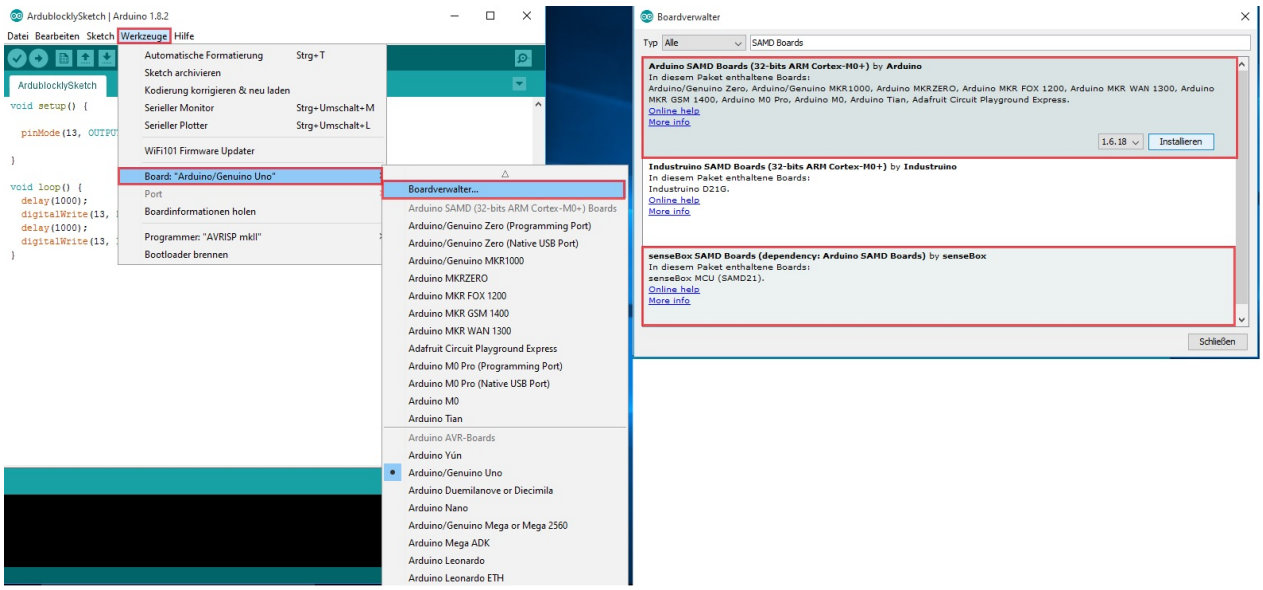
Paste the following URL into your Arduino IDE under Data -> Preferences in the field for Additional Boardmanager-URLs:

```
https://github.com/sensebox/senseBoxMCU-core/raw/master/package_sensebox_index.json
```



Open the Preferences and paste the URL

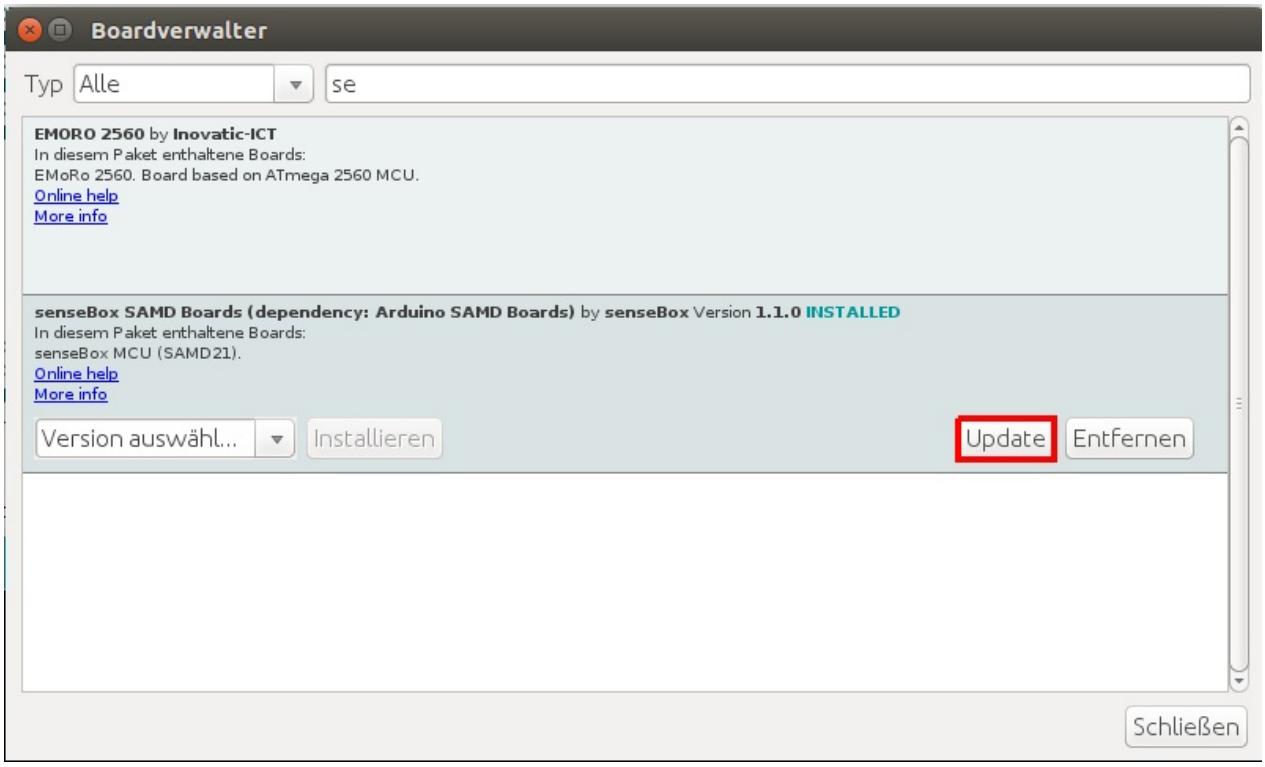
Please open then the Boardmanager under Tools -> Board:"..." -> Boardmanager and install there the two board-support-packages with the name Arduino SAMD Boards by Arduino and senseBox SAMD Boards by senseBox.



Open the Boardmanager and install both packages

> Enter "SAMD" in the search bar above to find the packages faster

Because we are updating the senseBox SAMD Boards-package regularly for you, you should from time to time have a look into the boardmanager to check if the senseBox SAMD Boards-package still is the latest. Therefore please open, like described above, the boardmanager and search for senseBox SAMD Boards. If you click at the entry in the list, there will pop up an update-Button in case that a new version is available. Click the button to install the latest version.



Click on 'Update', to refresh the Board-Support-Package zu

> It is important to first click on the entry. Otherwise the update button will not show up even if there is a new version.

Step 3: Pin and Wiring

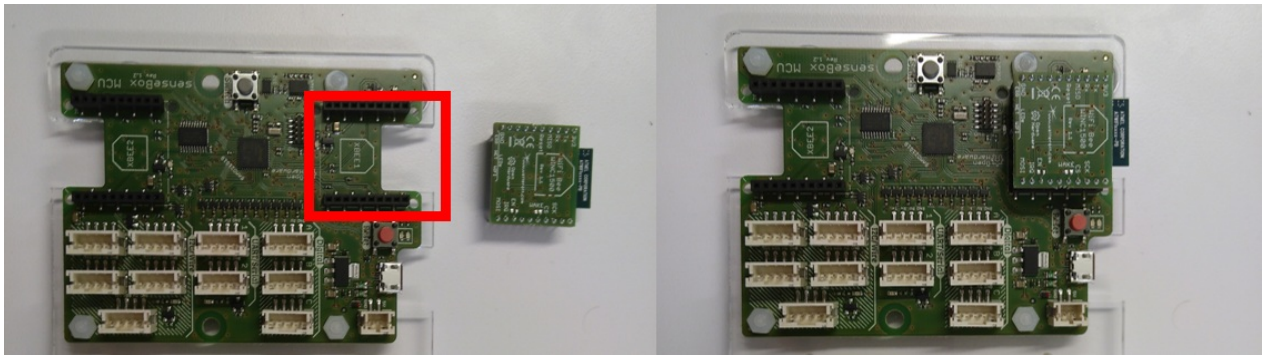
Due to the I2C plug-in system, the pin of the sensor and components to the senseBox MCU is very easy. You will here see a complete overview of all the components.

Pin of Bees

The pin of the bees is very simple. The plug-in system makes it possible to just plug the bee on to the microcontroller. You have to consider only two things: 1. The orientation on the board and 2. the correct portpin on the microcontroller.

WiFi-Bee, Ethernet-Bee and LoRa-Bee

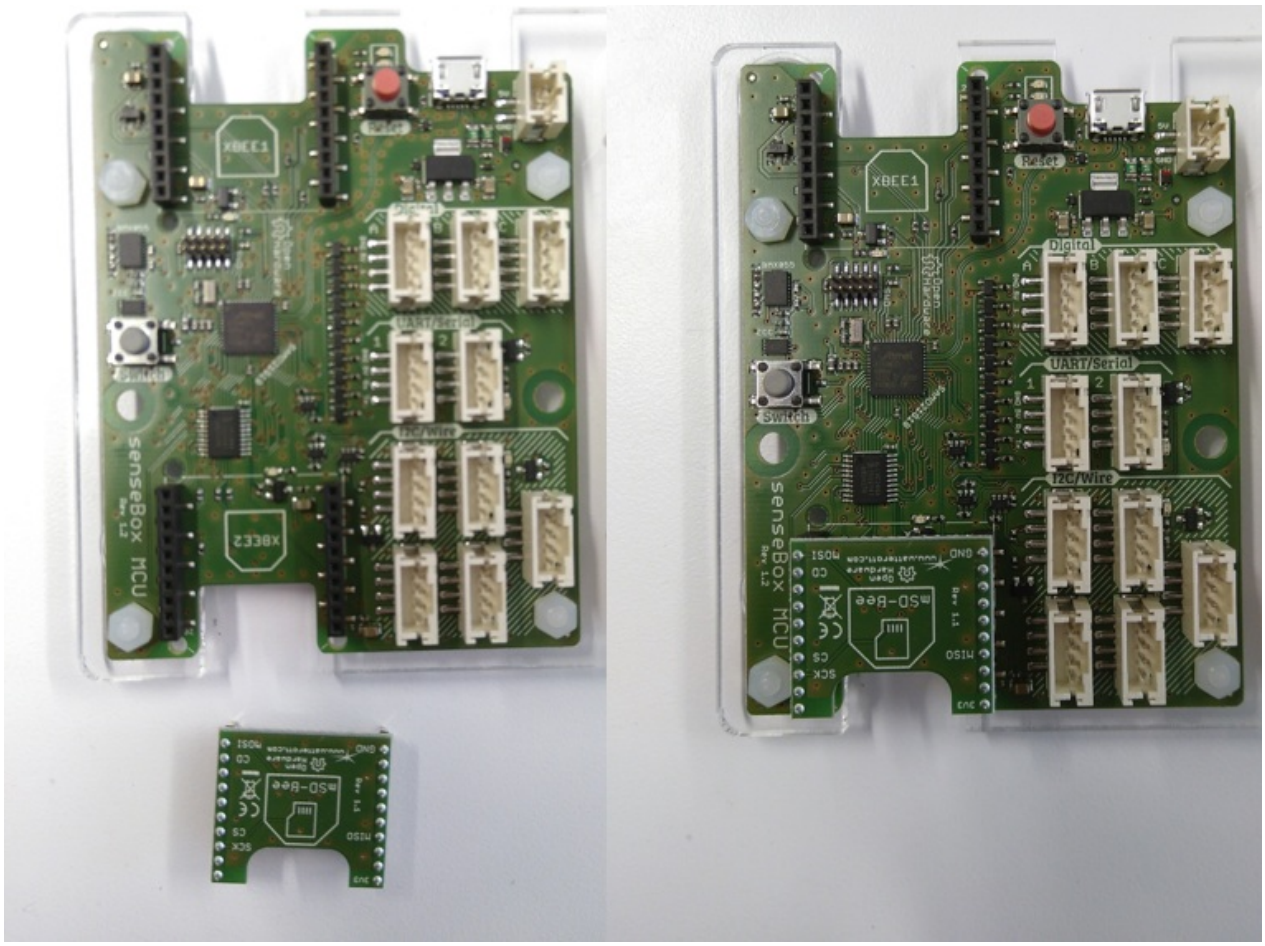
These bees all have to be plugged in port 1. You can identify the correct port by the marking: `XBEE1` . The 7-angled sign on the board and the bee shows you the correct plug-in direction.



Exemplary connection of the WiFi-Bee to the MCU (XBEE1)

mSD-Bee

The SD-bee has to be plugged on to port 2, which is by default activated. The marking: `XBEE2` shows you the correct port and the 7-angled sign the correct direction of the pin.



Connecting the mSD-Bee to the MCU (XBEE1)

Pin of the Basic Sensors

The sensor, which can be bought with the senseBox are easy to connect with the enclosed "I2C to I2C" wires. Please use the slots that are marked with `I2C/Wire` .

Connection of simple sensors

Pin of the Fine Dust Sensor

The fine dust sensor, which can be bought with the senseBox has an enclosed and fitting wire that connects sensor and board. Therefore you have to use the slots with the marking: `UART/Serial` . Here[1] you can find further information to the pin of the fine dust sensor.

Connection fine dust sensor

[1]. See *3.1.3.4 Fine Dust Sensor* ↩

Step 4: Programming the Hardware

In this chapter we describe how to program the senseBox using Arduino IDE and how to test the sensors and components of the senseBox.

> The software we code for senseBox is also called "sketch" in the following.

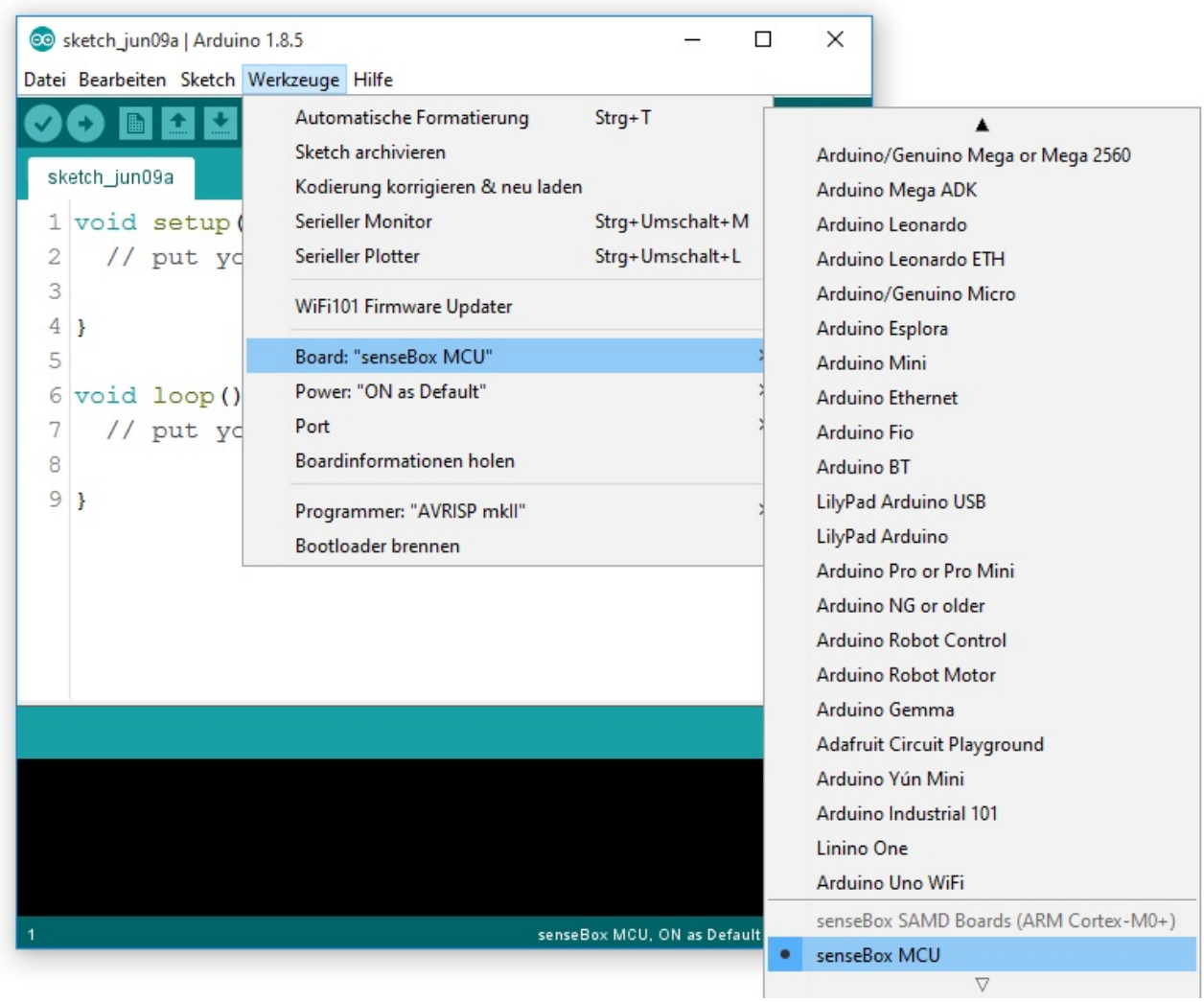
Programming with Arduino IDE

Using the Arduino IDE you can write code, compile the code and upload it on the senseBox microcontroller unit (MCU). Therefore you first have to connect the senseBox with the computer using hte USB-cable. Then follow the following steps.

Configuration of the Arduino IDE

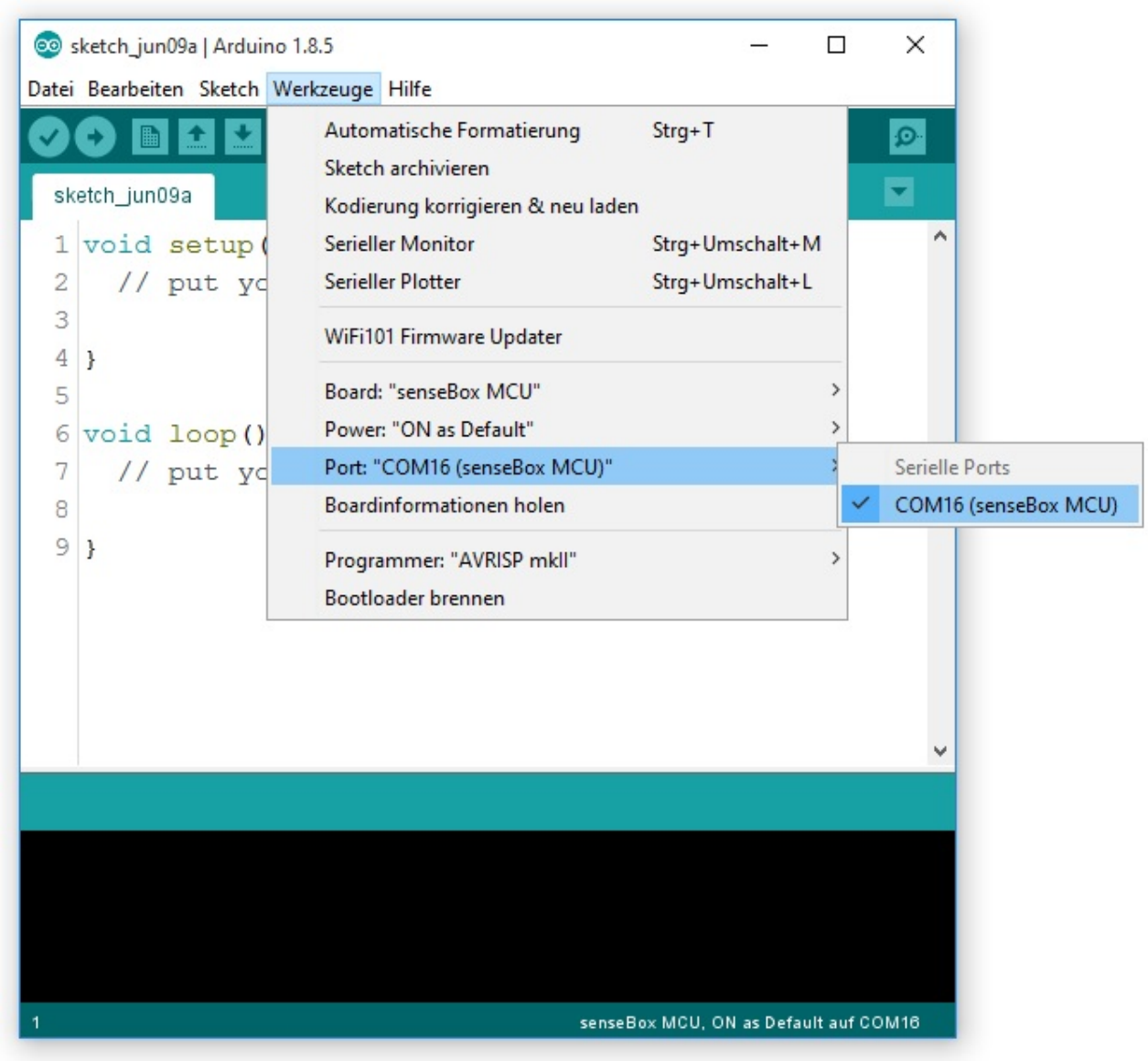
Before you can start programming the senseBox, you have to change some settings in the Arduino IDE.

First, you choose `Tools` in the menu bar and search the senseBox MCU in the option `Boards`.



Boardauswahl

Second, you choose again `Tools` and `Port` and search for the USB Port of your computer which is aligned to the senseBox MCU. Usually the computer gives you only one option. Depending on your operation system you have to choose the port repeadetly when putting code on the board.



Portauswahl

> Please consider that you can only choose a port when the senseBox is connected to the computer via USB.

Hello World Example

Now you can start putting your first code on the senseBox. Therefore copy the following example into your Arduino IDE and click the arrow to upload the code on your senseBox.

In the down part of your Arduino interface you receive feedback about your uploading process (in the black colored box). If everything worked out sucessfully, you will receive the note `Done upload` .

```
int ledPin = LED_BUILTIN;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                // wait for a second
}
```

Now you have seen your first Arduino code. The text which you can see behind the `//` is not code, but a comment of the person who write the code. In programming this helps other persons to understand the code and also yourself to remember your ideas when building it. However, this comments will not be read by the compiler.

> In contrast to your laptop or smartphone the senseBox does not use any operating system such as Windows, Linox, Android or MacOS. Always the last program which you put on the microcontroller will be accomplished.This means that if you plug out the senseBox of your computer and put it to another power source it will still remain to run the same code again and again.

Test sensors and internet connection

Before you connect and register your senseBox to the openSenseMap, you should check if all sensors and the network modul is working to see if you build your hardware in the right way. Therefoe, a testing programm exists which you can run on your Arduino.

> Please remember that you have to install the newest version of the Board-Support-Package as described in Step 2. There is also a information how to update the package at the end of Step 2.

Open the Test-Sketch

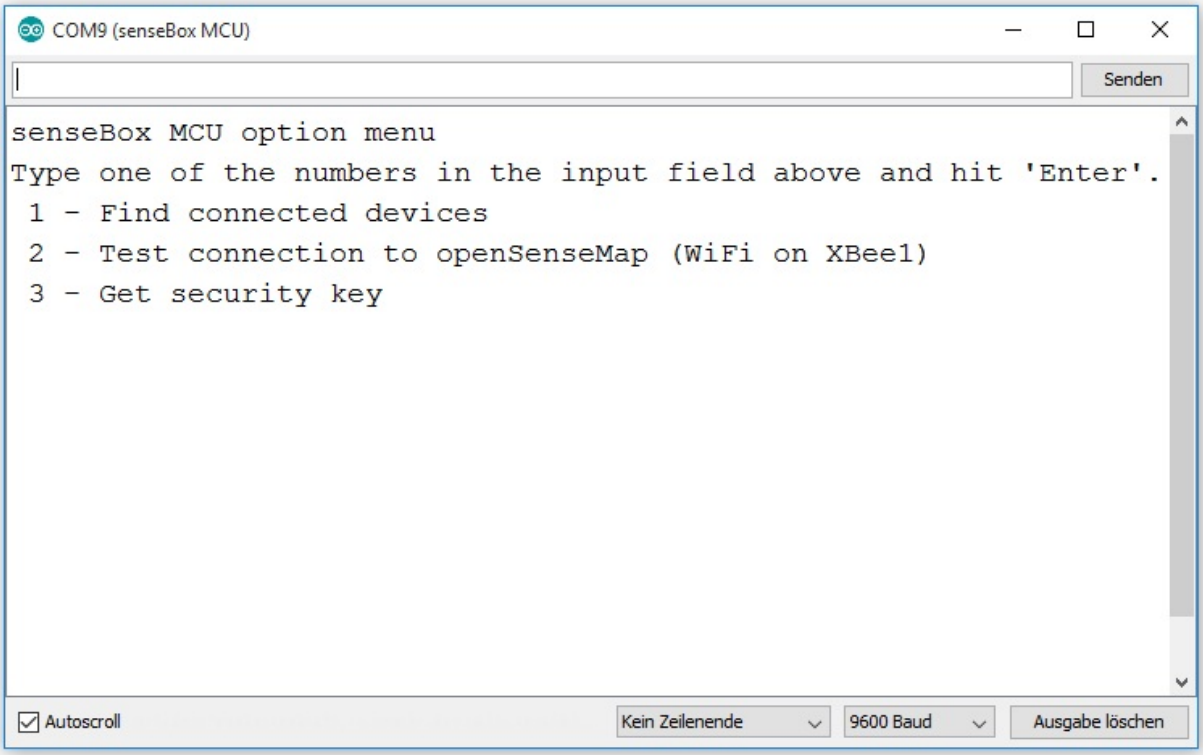
The Board-Support-Package comes wiht some test sketches for your senseBox. This file is named `mcu_component_test` and you can access it via `File` -> `Examples` -> `senseBox-Tests`). Now you can upload the code on your MCU by following the same instructions than you did for the Hello World Example.

After uploading the sketch on the MCU, you can start the `Serial Monitor` by clicking on the loupe icon on the right side. This should open a new window.

> If the monitor is not opening, you should check if the senseBox MCU is in "Program Mode". To do this press the reset button on the senseBox Board (the red button on your senseBox) and then make sure that the right port ist connected. Afterwards you click the loupe icon again.

Option Menu in the Serial Monitor

After opening the Serial Monitor you find a menu with different functions which you can access using the search field:

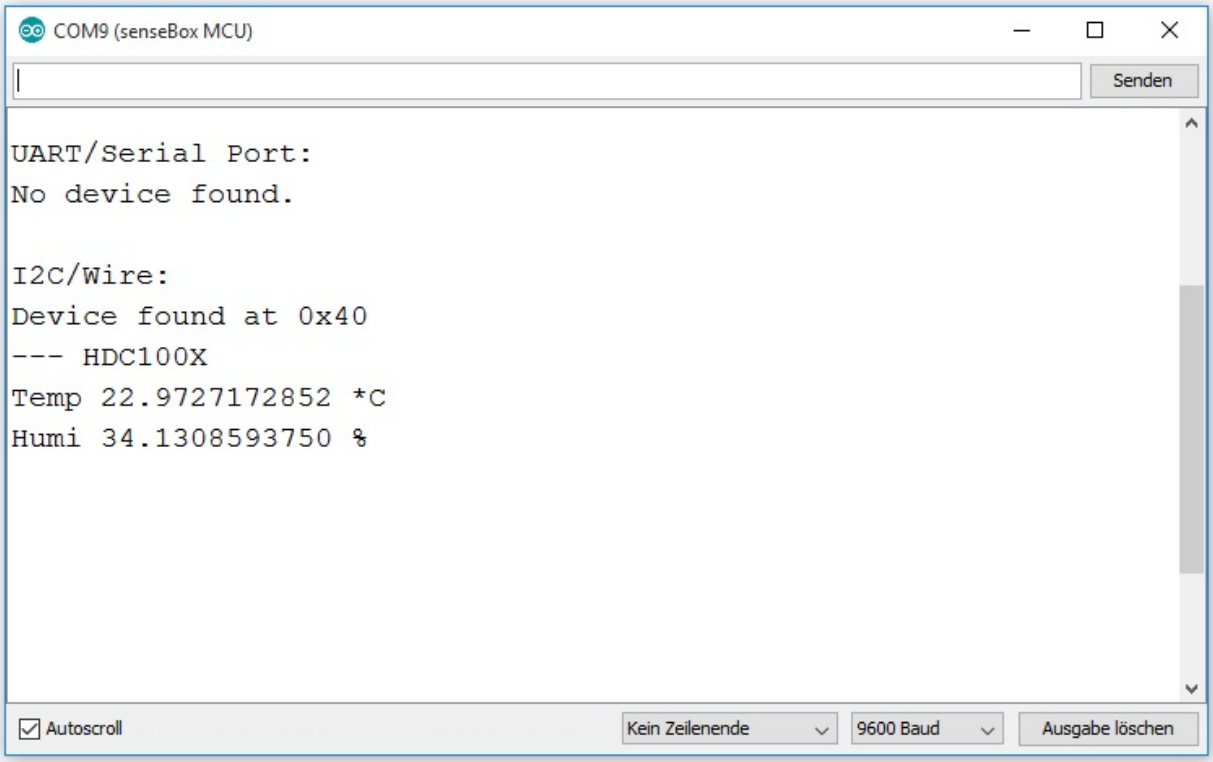


Optionsmenu

Just type the number of the option you want to test into the search field and click `Send`. Below you will find a list of options and a short description:

1. **Find connected sensors**

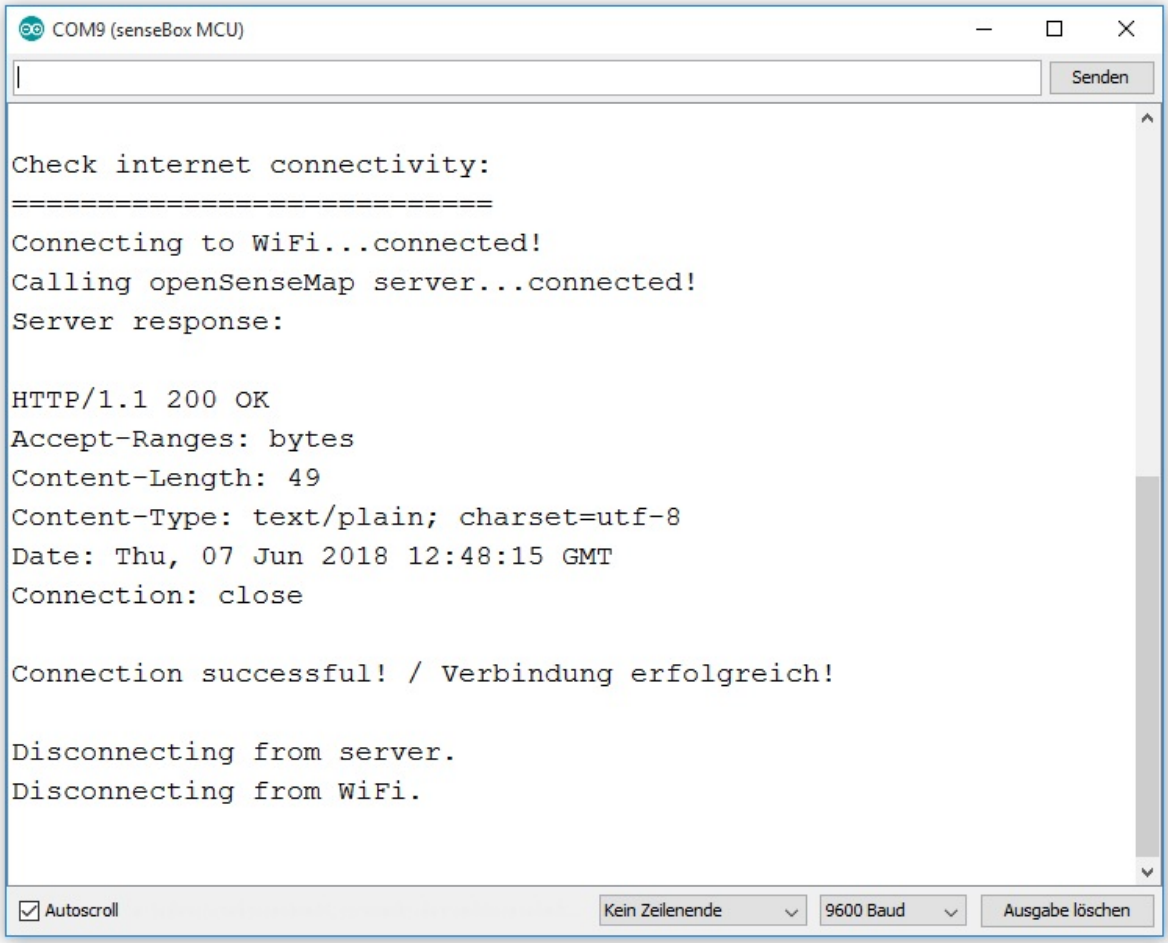
Here you can check if all sensors are installed in a correct way and if they are initialized and recognized by the senseBox. For each connected sensor you should receive a feedback and a test measurement. In the example below, a HDC1080 Temperature and Humidity sensor was connceted correctly to a `I2C/Wire` port.



If you are missing the data of a sensor which you connected with the senseBox, check all cable connections and try it again. You can also try different ports on your senseBox as well as exchange cables between the sensors.

1. Test connection to openSenseMap

   With this option you check your internet connectivity. If your senseBox successfully reaches the server you will get a response of the server in form of a HTTP-Status 200.



Sensortest

> If you use a Wifi-Bee, the test also checks if your firmware is updated to the newest version. If the version of your Wifi-bee is not the newest, you should update the firmware as described in the FAQ of this book.

2. Get security key

   Each senseBox board has a own and unique security key which you can receive here. This key is used to encrypt the connection between the senseBox and the openSenseMap.

> We are currently working on a system to register your senseBox on the openSenseMap by using this key. For the first you can proceed without remembering the key.

Step 5: Encryption

Soon you will find here information to the encryption of the senseBox with the help of the kryptochip on it. Unfortunately, we have not yet been able to fully test and document this feature.

Step 6: Registration on the openSenseMap

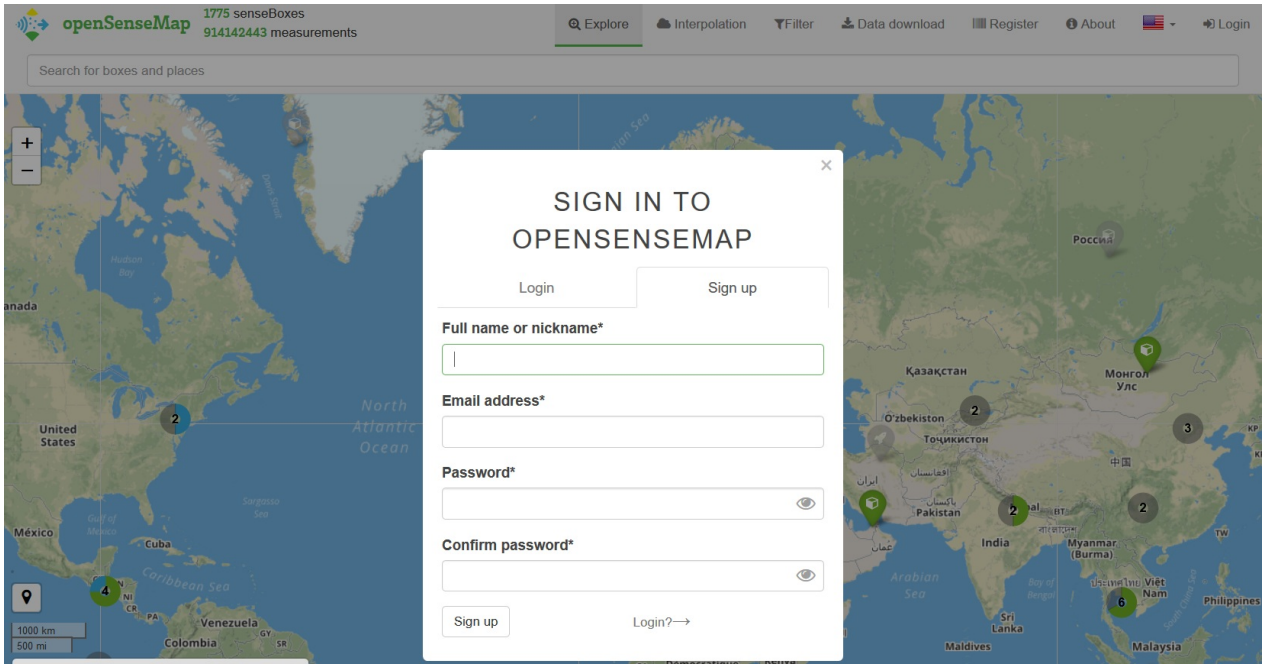
In order to transfer the measured data to the openSenseMap, you must first register there and create a new senseBox. How to do that you will find out in the following section

▼ What is the openSenseMap?

The openSenseMap is a project to file, compare and visualise sensordata. Take a look at it by clicking on www.opensensemap.org[1] and discover a vast data pool of sensordata from around the world. Not only senseBoxes send data to the openSenseMap but also other microcontroller with sensors. You can here[2] see how the individual functions of openSenseMap and their interfaces work.

1. Useraccount Registration

Please go to www.opensensemap.org and create an account. To do this, click on "Login" in the top right corner and then go to "Sign up". You need a valid e-mail address to register. After successful registration you will receive an e-mail with a confirmation link. Please click the link to complete the registration.



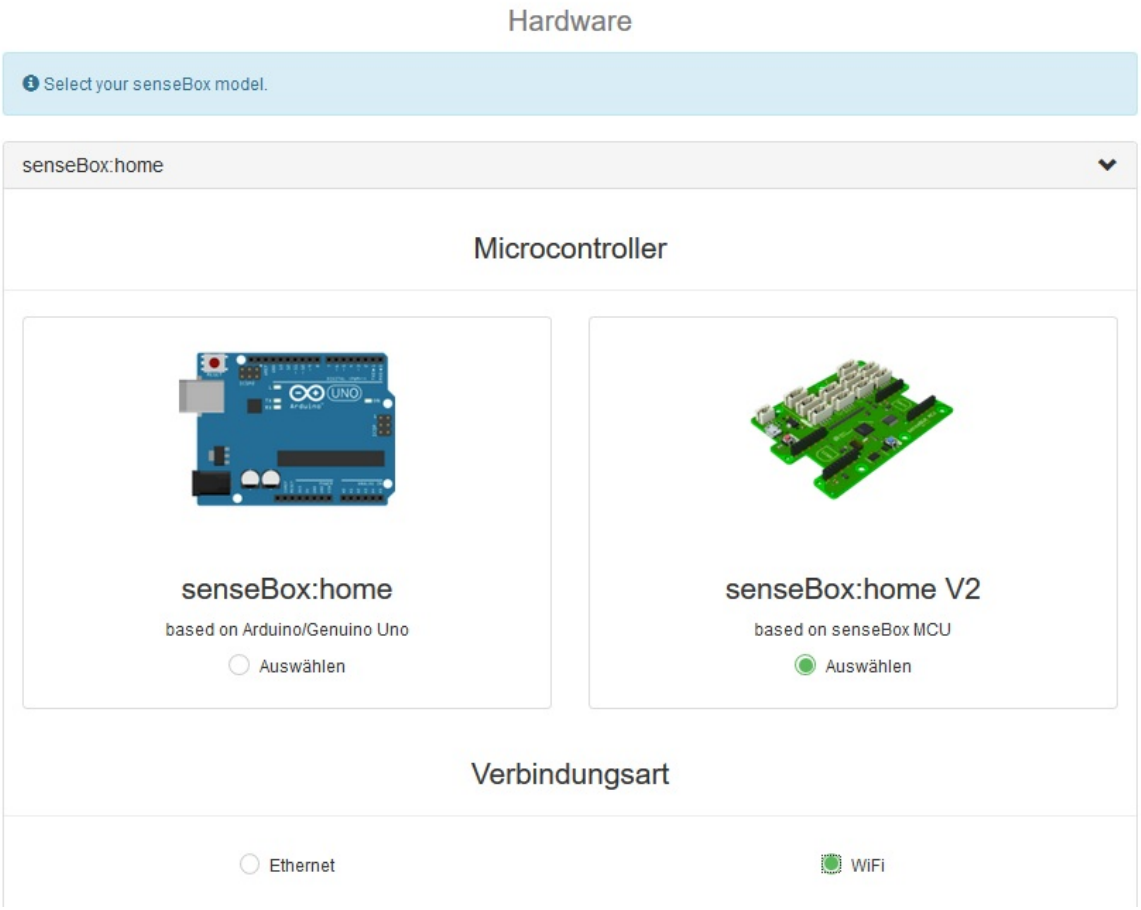
2. Set up a new sensBox

After successful registration you can start creating your senseBox. Click on "New senseBox / New senseBox" in your menu and confirm the terms of use. Follow the instructions of the registration and provide the following data:

- A freely selectable name of the senseBox
- The exposure, in which you want to set up your senseBox

- The location, where you want to set up your senseBox (you can also have your location determined automatically)

Since the openSenseMap is open to all types of senseBoxes, you will be asked about the used hardware. Select senseBox: home V2 and the bee you use (WiFi, or Ethernet)



Selection of senseBox:home V2 MCU and WiFi Internetconnection

You are not sure if you have an ethernet or a wifi-bee, or you cannot see the difference between the airpressure and temperature sensor? In the chapter Komponenten you are going to find pictures and further hints for the individual components.

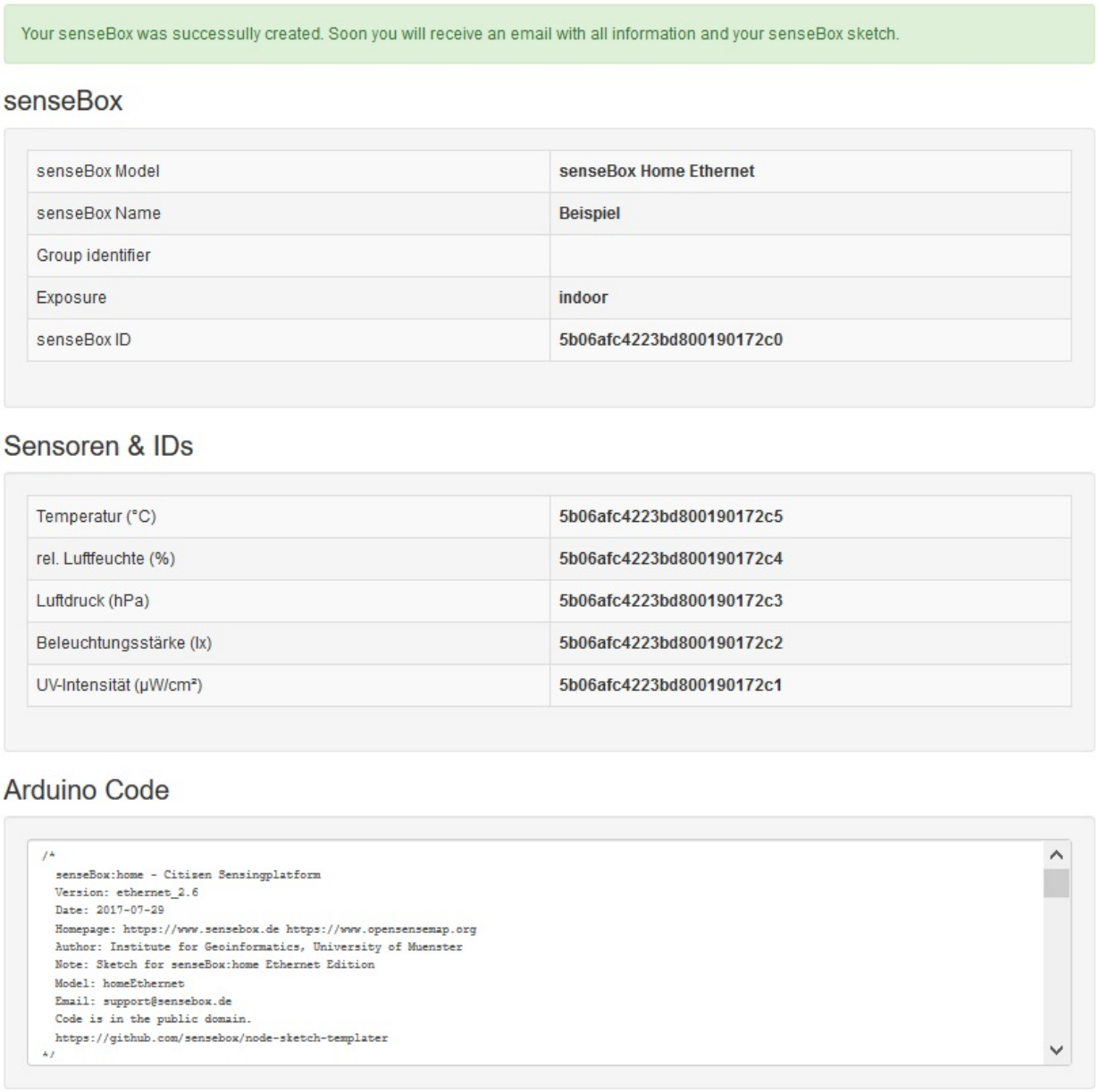
Now you are almost done! Quickly select the sensors that you will connect to your senseBox. Just click and ready. If you want to install a finedustsensor, you also have to specify which serial port you want to connect to. After that you can finish the process.

▼ Why do I have to connect a finedustsensor to the serial-port?

Here should be an explanation

3. Receiving a Summary of the Registration

Once you have completed the registration, you must once again agree to the publication of your data. After that you will receive a summary of your registration. There you will see your senseBox ID, your sensor IDs and the Arduino code (you will also receive it via e-mail)



Summary of the registration. Here you can see the IDs to your senseBox and the sensors!

▼ What is my senseBox ID

The senseBox ID is a unique identifier of your senseBox. With this ID you can find your senseBox for example with a URL on the opensenseMap (opensensemap.org/explore/HIER-DEINE-SENSEBOX-ID-EINGEBEN) or in a data pool. Furthermore, it is used for various applications and functions around the senseBox and the openSenseMap.

Loading the Arduino-Code on the senseBox

After you have downloaded the `.ino` attachment of the e-mail you have to load this program on to your senseBox. Of course you have to download the Arduino IDE (as explained in step 1) on your computer. In short, you can then follow the following steps:

(opensensemap.org/explore/ENTER-YOUR-SENSEBOX-HERE)

- WiFi-Bee
- Ethernet-Bee
- LoRa-Bee

WiFi-Bee

- Open Arduino application
- Select `Data` → `Open` in the data bar and the choose `sensebox.ino`
- The dialog field asks if you want to move the file. Confirm this with "Yes" or "Ok"
- Now you need to enter your WiFi network and password for the network between the `""`. Please mind to not have any spaces between the `"` and the letters of your password.
- Now you can load the program via the arrow icon on the microcontroller.
- Wait until the program has been transferred

Ethernet-Bee

- Open Arduino application
- Select `Data` → `Open` in the data bar and the choose `sensebox.ino`
- The dialog field asks if you want to move the file. Confirm this with "Yes" or "Ok"
- Now you need to enter your WiFi network and password for the network between the `""`. Please mind to not have any spaces between the `"` and the letters of your password.
- Now you can load the program via the arrow icon on the microcontroller.
- Wait until the program has been transferred

LoRa-Bee

Unfortunately we have not been able to write a tutorial for the LoRa-Bee yet. If you would like to help us, please send us an e-mail to info@sensebox.de

---

If everything went right, you can now select your station on the openSenseMap and watch how measurements are transmitted continuously!

> You can not remember how to transfer the code from Arduino to the senseBox? Look again Step 1 and Step 5 , the installation and the transfer will be explained there. In case you have further questions, you are welcome to use our forum, to get information or to write a post.

> [1]. https://www.opensensemap.org ↩
>
> [2]. https://sensebox.github.io/books-v2/osem ↩

Step 7: Assembly in Case

Below the assembling of the senseBox:home in the case is going to be explained.

Inventory

Basic Configuration

Komponentenübersicht der Standardversion

- case with lid
- mini-USB wire + adapter
- senseBox MCU

- sensor for air humidity and temperature
- protective case for air humidity and temperature sensor + attachments
- plexiglas with plastic bolts
- M20 screw thread
- bag with 8 screws
- Bee

Installation

1. Step

mounting of senseBox MCU in the case:

Therefore you have to align the MCU so that the micro USB-access and the red button are pointing to the hole in the case. Please screw the MCU to the case with the two provided screws. Therefore use the two holes in the middle of the long side of the MCU.

MCU in the housing

2. Step

Connecting Bee and humidity-/ temperaturesensor:

Independently which bee you are using, in the next step you have to mount the bee on the intended XBEE1 slot.

3. Step

Pin of the sensor and the mini-USB wire:

For this you have to first screw the M20 thread in the hole on the side of the case. Please then unscrew the cover and put the wire (without sensor) and the mini-USB wire whereby the opening of the screw thread. Now you can screw the cover on to the screw thread again and connect the sensor with the wire. The sensor plug can be attached to one of the 5 12C/Wire slots. Please connect the mini-USB wire not before everything is built together.

Temperature and air humiditysensor + USB

4. Step

mounting the sensor in the protective case

The protective case ensures that temperature- and air humiditysensors are not exposed directly to the sun. To install the sensor in the case you have to enlarge the hole in the underside of the case so that the sensor fits through.

Now you can mount the sensor with the two plastic bolts (see photo) or zip ties on to the little framework. You can now mount the case wherever you want with the attachments.

> Achte darauf, dass der Sensor dabei nach oben zeigt!

Temperature and air humiditysensor in the protective case

Additional Components (optional)

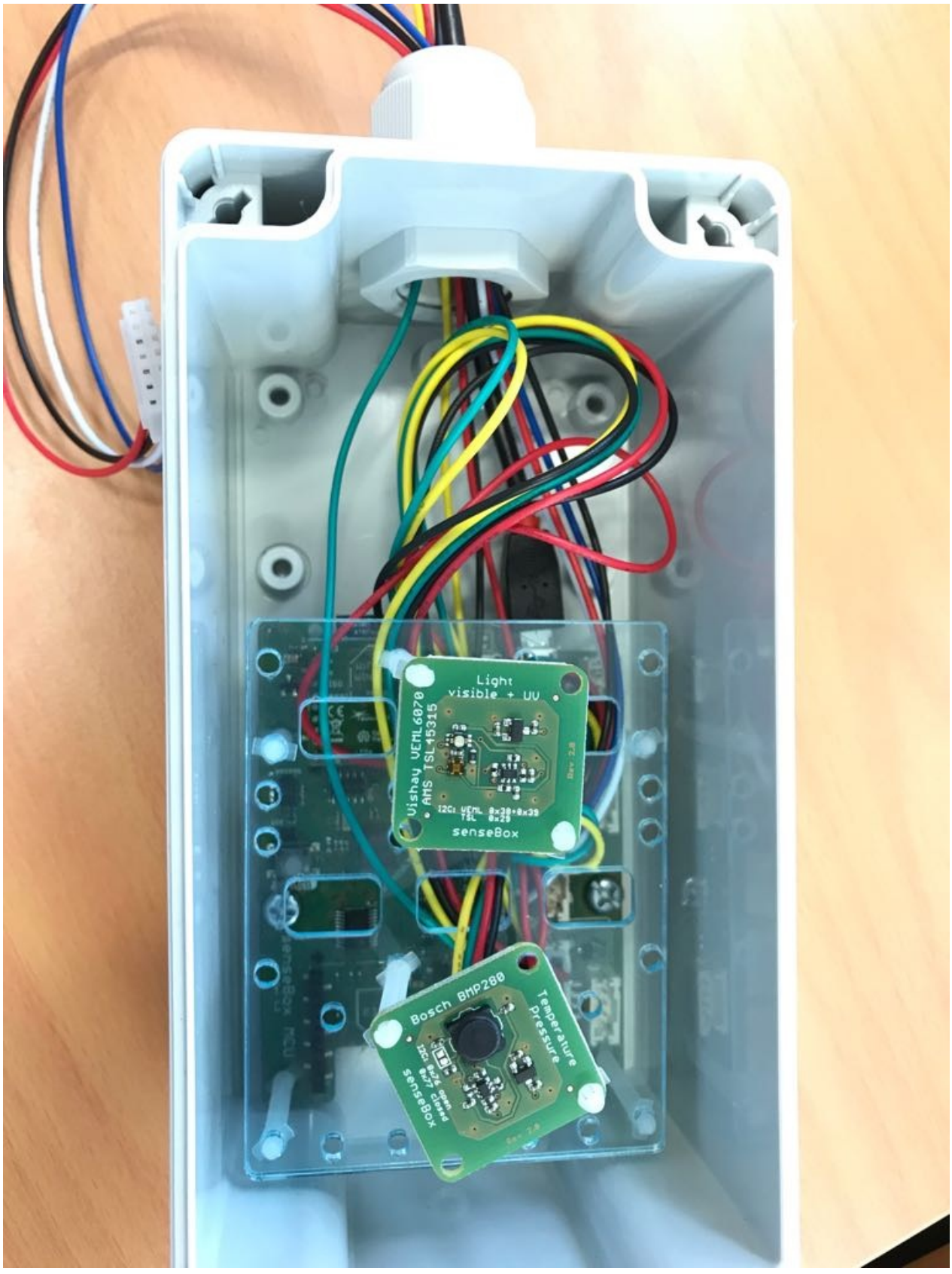
5. Step

In case you want to add further sensors, please initially connect the intended wire with the 12C/Wire ports. If you have a finedustsensor begin with putting the wire through the screw thread. Following connect the other end of the wire to the UART/Serial port 1. Now push the 4 plastic bolts in the outer holes of the MCU

Pin of the 3 standardsensors and the finedustsensor

6. Step

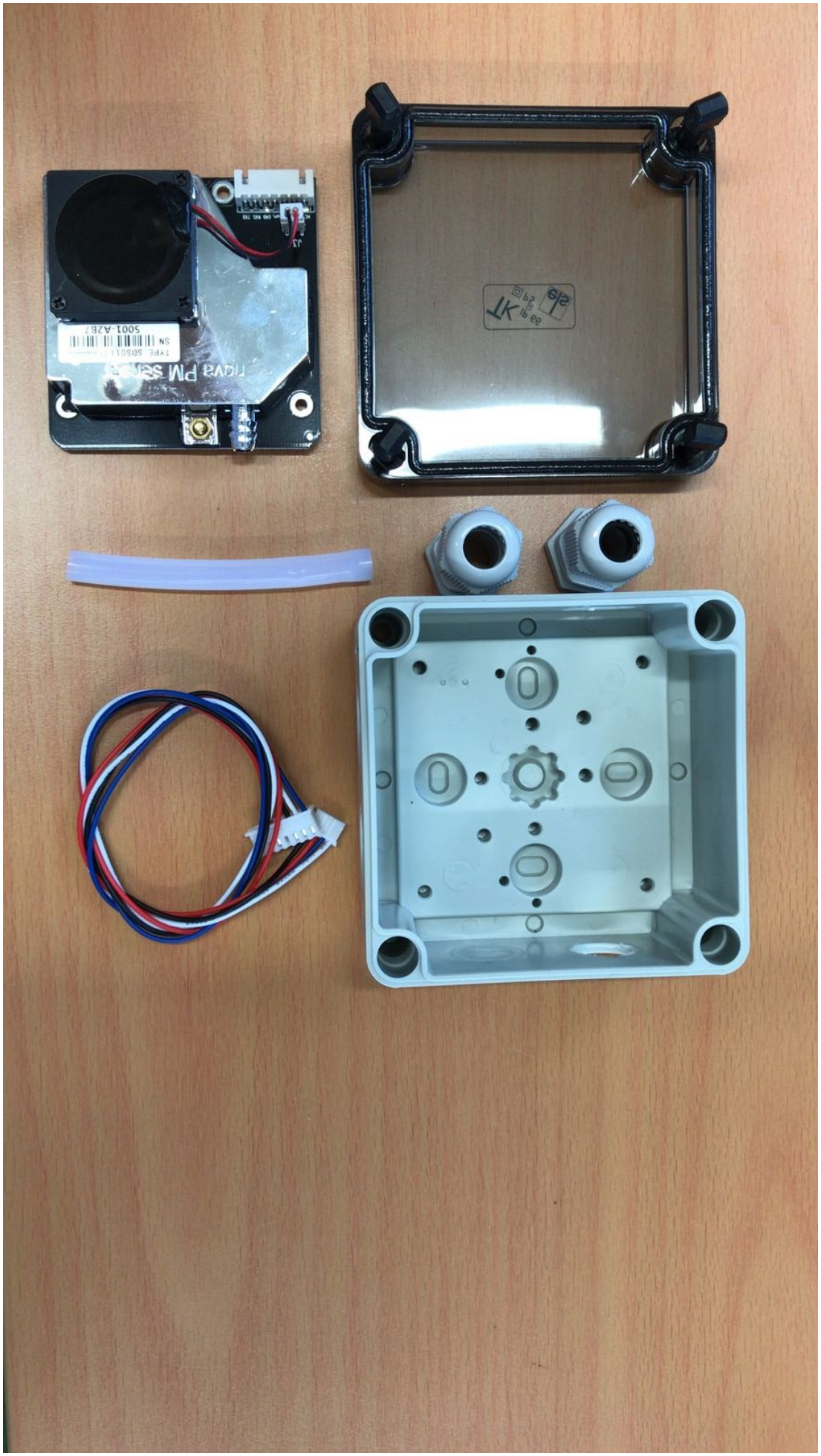
In the next step please remove the plastic foil of the Plexiglas. Now you can put each wires through the middle of the 3 holes. Position respectively 2 plastic bolts for one sensor and the little holes so that the sensors can be mounted central on the Plexiglas This is especially important for the UV sensor!



Pin of the sensors on the Plexiglass

7. Step

Installation of the finedustsensor

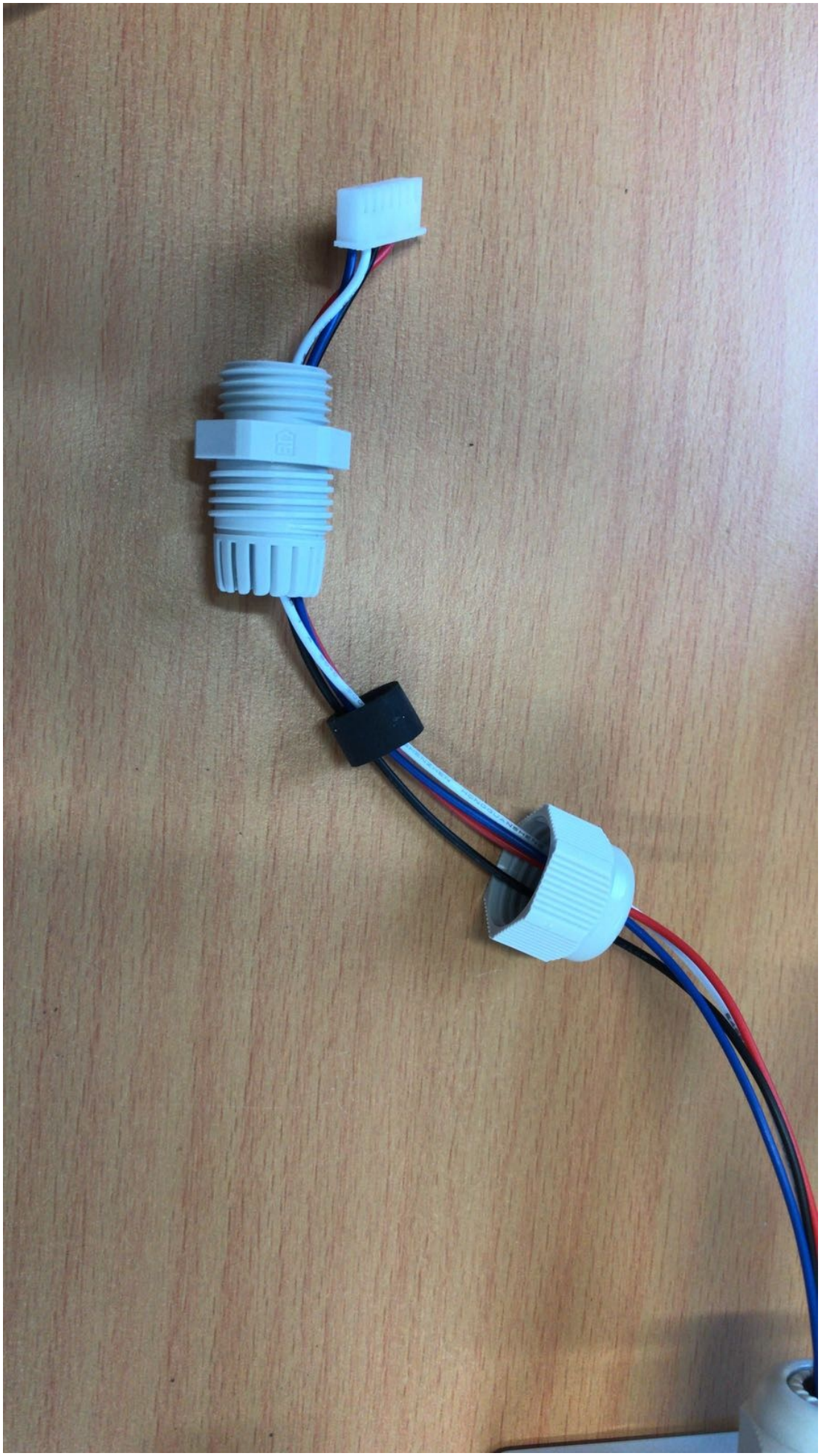


Components of the finedustsensor

- finedustsensor
- case
- 2 M16 screw threads
- wire
- plastic tube

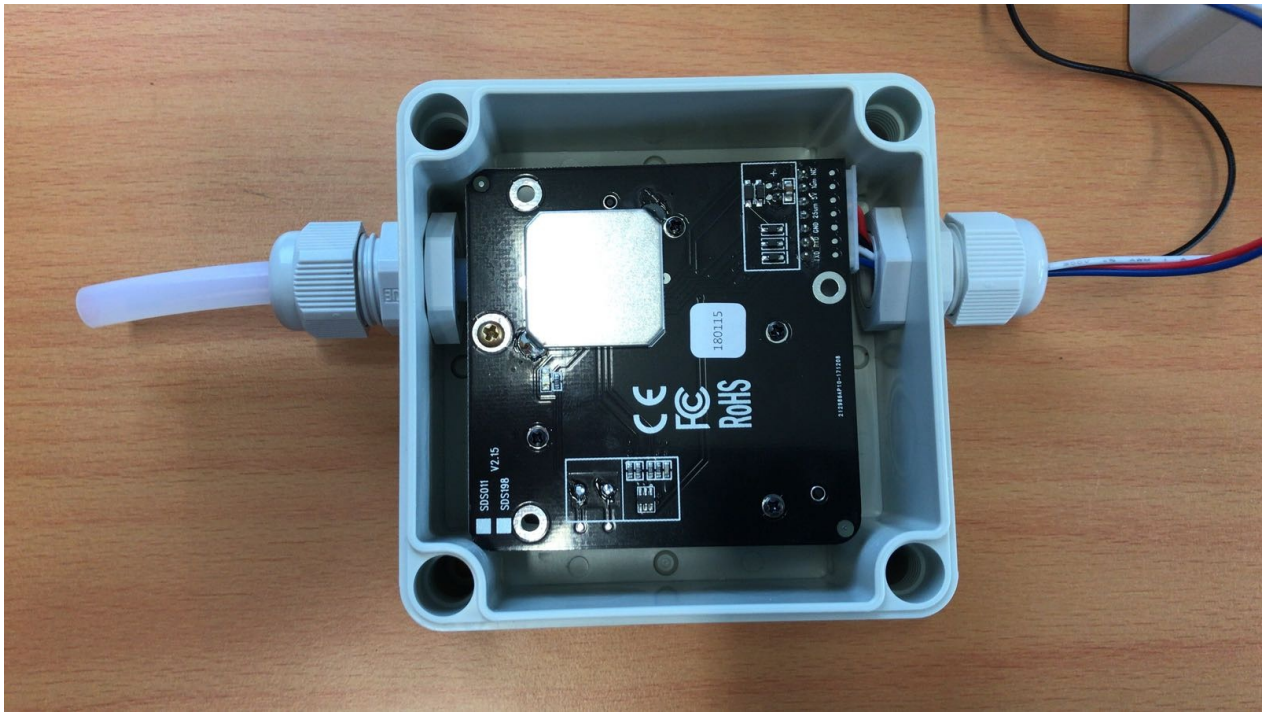
Please follow the instruction for the pin of the finedustsensor: First unscrew the cap and afterward the rubberseal in the screw thread. Now you can successively put the sensor wire through the just detached components. Then stick the wire through the hole in the case and connect it with the sensor.

> Please do not fasten the screw thread at this point!



Procedure to connect the wire of the finedustsensor

Stick now the plastic tube on to the entry of the finedustsensor and afterwards through the other hole in the case. Now you can fasten the screw thread on both sides of the case and mount the lid on to the case.



Completed connection of the finedustsensor

Completed Installation of the senseBox:home

The senseBox:home below includes temperature, humidity, finedust, UV and airpressure sensors.

Completed Connection

Step 8: Set up the senseBox

To maximize data transferability and data comparability, there are some tips you should follow when setting up the senseBox:home.

Technical Equipment at the Site of Installation:

1. Power supply: The senseBox needs a continuous power supply. The supplied cable has a length of 3m.
2. Data transfer: The senseBox needs a way to continuously transfer data to the Internet. The senseBox can be ordered with WLAN or LAN transmission option. Depending on the version you need access to the appropriate network (WLAN network, or LAN cable to the router)

Site of Installation:

1. Basically, there is no right or wrong site. We encourage you to build the senseBox where you are interested in measuring yourself.
2. The senseBox community is pleased if the senseBoxes have a good documentation on the openSenseMap. This includes: a. A picture of the station b. Procurement of the location, such as height, roofing, orientation. c. Other descriptions
3. From a scientific point of view, of course, there is an extensive investigation for optimal data acquisition. Many different factors influence the measurement quality. We refer to the following points:
4. Of course, different positions change the readings. If you place the senseBox directly in the sun, the temperature values can be very high.
5. It is also better to mount the senseBox slightly away from the wall of the house (for example, on a balcony railing), to avoid falsification by waste heat. Detailed information on the meteorological data collection can be found e.g. under the following link: https://library.wmo.int/pmb_ged/wmo_8_en-2012.pdf
6. The quality of the measured data also depends on the mounting of the sensors within the senseBox. This information can be found in our building instructions: www.books.sensebox.de[1]

Maintenance

1. It should be absolutely avoided that condensate collects in the senseBox. Therefore, a visual check of the senseBox is recommended at least once a month. If necessary, the Silica gel should be renewed.
2. The senseBox should be freed regularly of accumulating dirt.

You can find further tips on www.sensebox.de[2] and in our forum www.forum.sensebox.de[3].

[1]. https://www.books.sensebox.de ↩

[2]. https://www.sensebox.de ↩

[3]. https://www.forum.sensebox.de ↩

Overview of Available Components

Here you will find a list with all sensors, bees and other components of the senseBox. We created a page for each part to explain it and provide you the information necessary to work with the part.

If you want to know how and for what you can use the part, click on it.

1. senseBox MCU
2. Bees
   - Wifi-Bee[1]
   - LAN-Bee[2]
   - SD-Bee[3]
   - LoRa-Bee[4]
3. Sensors
   - Temperatur & Humidity (HDC1080)[5]
   - Air pressure & Temperatur[6]
   - Illumination and UV[7]
   - Fine dust[8]
4. Additional Components
   - Sensor Protection[9]
   - Casing[10]
   - Power Adabter und USB-cable[11]
   - LED-Display[12]
   - HUB[13]
   - Micro-SD Karte[14]
   - GPS[15]

There are many more sensors which you can adapt with some own initiation and do-it-yourself skills. However, currently we can only provide information about the listet components. If you are interested in other sensor technology you can visit our Forum[1]. There you reveice help from the senseBox community.

1. See *3.1.2.1 Wifi-Bee* ↩
2. See *3.1.2.2 LAN-Bee* ↩
3. See *3.1.2.3 mSD-Bee* ↩
4. See *3.1.2.4 LoRa-Bee* ↩
5. See *3.1.3.1 Temperature- and Air Humidity Sensor (HDC1080)* ↩
6. See *3.1.3.2 Airpressure- and Temperature Sensor* ↩
7. See *3.1.3.3 Illumination and UV-Radiation Sensor* ↩
8. See *3.1.3.4 Fine Dust Sensor* ↩

[9]. See *3.1.4.1 Radiation Protection* ↩

[10]. See *3.1.4.2 Housing* ↩

[11]. See *3.1.4.3 Power Supply and USB-Cable* ↩

[12]. See *3.1.4.4 LED-Display* ↩

[13]. See *3.1.4.5 Expander* ↩

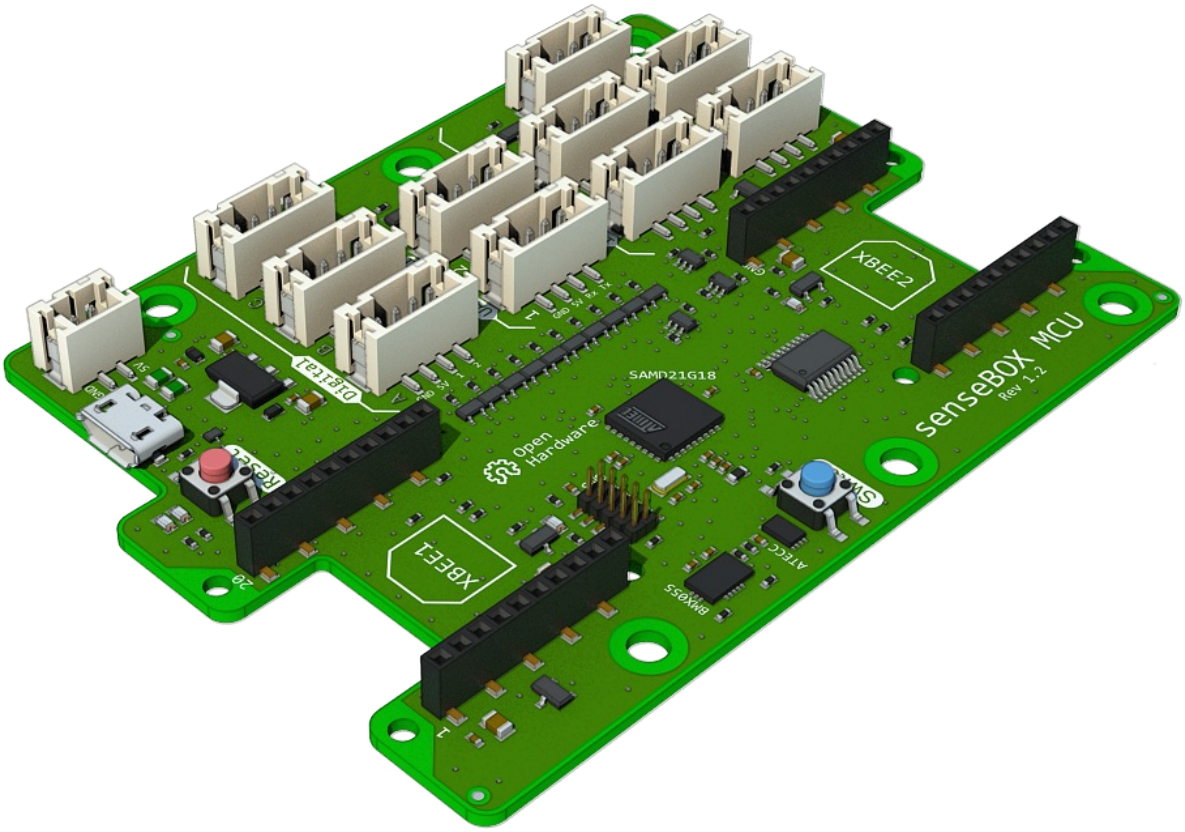
[14]. See *3.1.4.6 Micro-SD Karte* ↩

[15]. See *3.1.4.7 GPS* ↩

[1]. https://forum.sensebox.de/ ↩

senseBox MCU

The senseBox-Microcontroller (MCU) is designed and developped for the special needs of a senseBox. Therefore the microcontroller is focussed on especially the following three features: speed, low in energy usage and a big program storage.
However, if you want to participate in openSenseMap the senseBox team offers you a guideline to program it without prior knowledge in coding. For advanced programmer the Arduino IDE can be used to access the board and create individual projects.



The senseBox MCU

Technical Specifications

Processor

The processor is based on a ARM Cortex-M0+ processor form the SAM D21 family by Microchip.

Interface

Sensors and actors can be activated using standard interfaces like I2C, UART and digitale I/Os with a robust JST-Connecting system (5V tolerant).

Data transmission

Using the both Bee compatible sockets, UART or SPI models can be offered. Therefore, real time data transmission via Wifi, LAN, or LoRa is offered as well as saving data on a Mikro-SD card.

Features

1. Crypto Authentication for OTA (Over the Air)
2. Firmware-Upgrades using the ATECC608A by Microchip
3. Integrierted BMX055 sensor by Bosch, for measuring accelaration, affinity and orientation
4. USB CDC+MSC Bootloader (Arduino compatible)
5. Interfaces: I2C = 5 (more using a I2C Hub) | 2 UART | 6 analoge digital digitale IOs

Bees

There are four different Bees which you can add to your senseBox to save data or upload it to the openSenseMap.

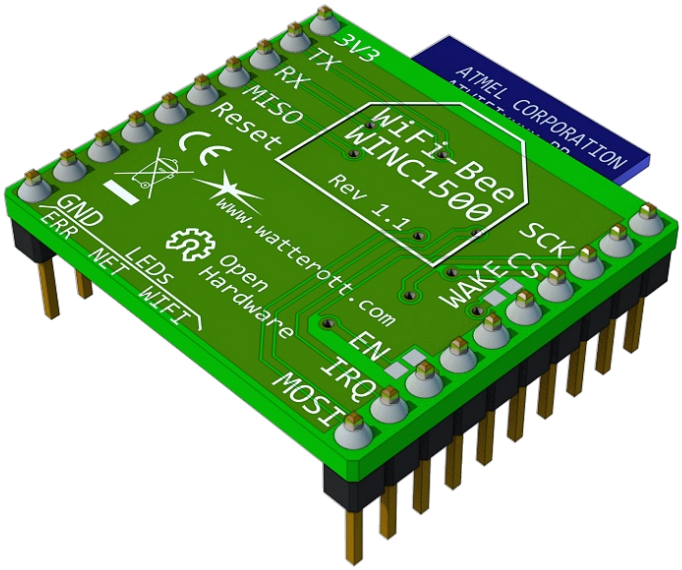
- Wifi-Bee[1]
- Ethernet-Bee[2]
- SD-Bee[3]
- LoRa-Bee[4]

1. See *3.1.2.1 Wifi-Bee* ↩
2. See *3.1.2.2 LAN-Bee* ↩
3. See *3.1.2.3 mSD-Bee* ↩
4. See *3.1.2.4 LoRa-Bee* ↩

Wifi-Bee

The Wifi-Bee is the connector between the senseBox and the internet. The data of the senseBox is transmitted via Wifi to the existing network. The Wifi-bee is based on the ATWINC1500 microchip by Atmel which has a very low energy consumption and a long range.

Some of our WINC1500 WiFi Bees may have outdated firmware (version 19.4.4) installed. This can lead to transmission problems. If these problems occur with you, please visit this website to refresh the firmware.



Wifi-Bee

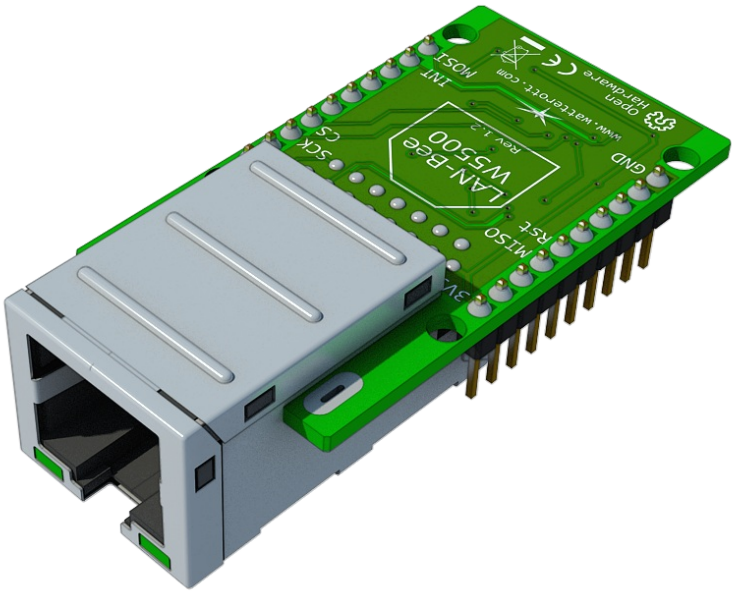
Technical Details

- "Plug-in-and-Go" senseBox compatibel
- Single-band 2.4GHz b/g/n
- Operating voltage: 3.0V to 4.2V

- Serial host interface: SPI
- Security protocols supported: WPA/WPA2 Personal, TLS, SSL
- Network services: DHCP, DNS, TCP/IP (IPv4), UDP, HTTP, HTTPS
- Name: WINC1500
- Measurements: 24mm x 25mm x 9mm
- Weight: 3,5 g

Ethernet-Bee

This Bee connects the senseBox with your router. The data of your senseBox will be transmitted using a ethernet cable to your router. The bee is based on the W5500 Mikrochip by Wiznet which allows a high ethernet data transmission rate.



Ethernet Bee

Technical Information

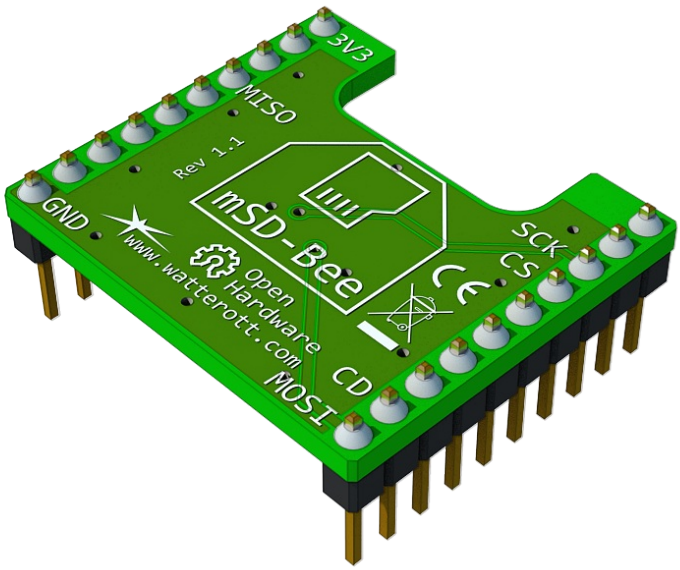
- "plug-in-and-go" senseBox compatible
- 3.3V operating voltage with 5V I/O signal tolerance
- Indication: W5500
- Measurements: 46mm x 25mm x 12mm
- Weigt: 9.2 g

Tipps and Hints

The Ethernet Bee will be delivered without Ethernet cable. We prefer to use flat cables, which can be also pulled below windows and doors.
Especially in combination with Power over Ethernet (PoE) the Ethernet bee is an interesting option.

mSD-Bee

With the SD-Bee, the data of the senseBox can be stored on an SD card. So you can measure, even if there is no Internet connection in the vicinity of the senseBox.



microSD-Bee

Technical Details

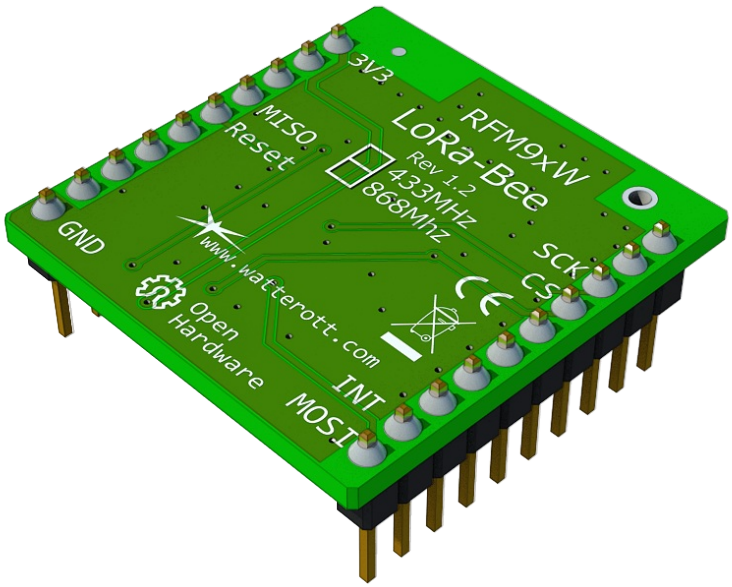
- "Plug-in-and-Go" senseBox compatibel
- Port for miniSD-card
- Name: mSD-Bee
- Measurements:24mm x 21mm x 9mm
- Weight:2,4 g

Information

Attention: The SD-bee is deliverd without a SD-card if you order in the senseBox-Shop.

LoRa-Bee

Use the LoRa-Inferface to upload your data on the openSenseMap. The LoRa WAN-Bee-Modul is a low energy and cost free option to upload your data unsing the LoRa-Radio-Standard. Therefore existing LoRa-Networks such as TheThingsNetwork are used for data transmission. The necessary infrstructure is provided by the community of TheThingsNetwork and available in more and more regions.



Lora Bee

Technical Information

- HopeRF RFM95W/RFM96W LoRa Transceiver
- LoRa-Bee 868 / 915 MHz uses RFM95W (SX1276 compatible)
- LoRa-Bee 433 / 470 MHz uses RFM96W (SX1276 compatoble)
- SPI interface
- Indication: RFN9xW
- Measurements: 46mm x 25mm x 12mm
- Weight: 1,1 g

Information

Please check if your area is already covered by LoRa before you get your senseBox with LoRa Bee: https://www.thethingsnetwork.org/community#list-communities-map

Attention: Due to the increased complexity of the installation, we recommend the LoRa module except advanced users of open hardware

Upload via LoRaWAN

It is possible to load sensor data via LoRaWAN ™ by the [TheThingsNetwork] (https://thethingsnetwork.org) (TTN) to the openSenseMap. LoRa is an increasingly popular radio standard, which is similar to WiFi. It allows digital data transmission in an IP network, but provides notable different features including:

- Data throughput: 300 - 3000 Bit/s
- Range: up to 15km

TTN is one of several projects that are related to the radio hardware Infrastructure implemented for the IP network. Whereby registered devices can be connected to the internet

Users can add Gateways as well as Nodes to the network.

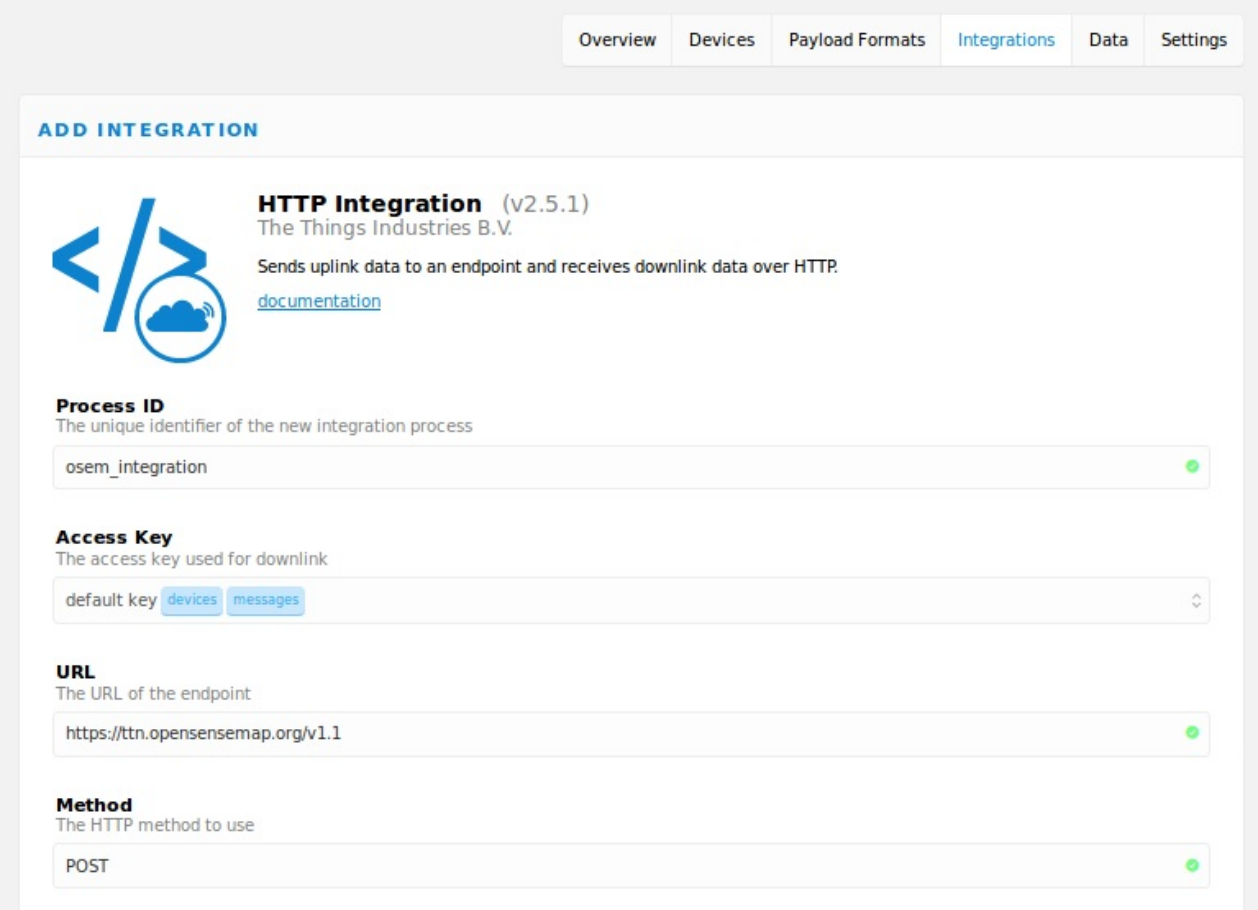
TTN openSenseMap Integration

The openSenseMap provides a direct integration into the TTN network, which simplifies the configuration. You therefor need to create an account on [TheThingsNetwork] (https://thethingsnetwork.org).

Registration in TTN Console

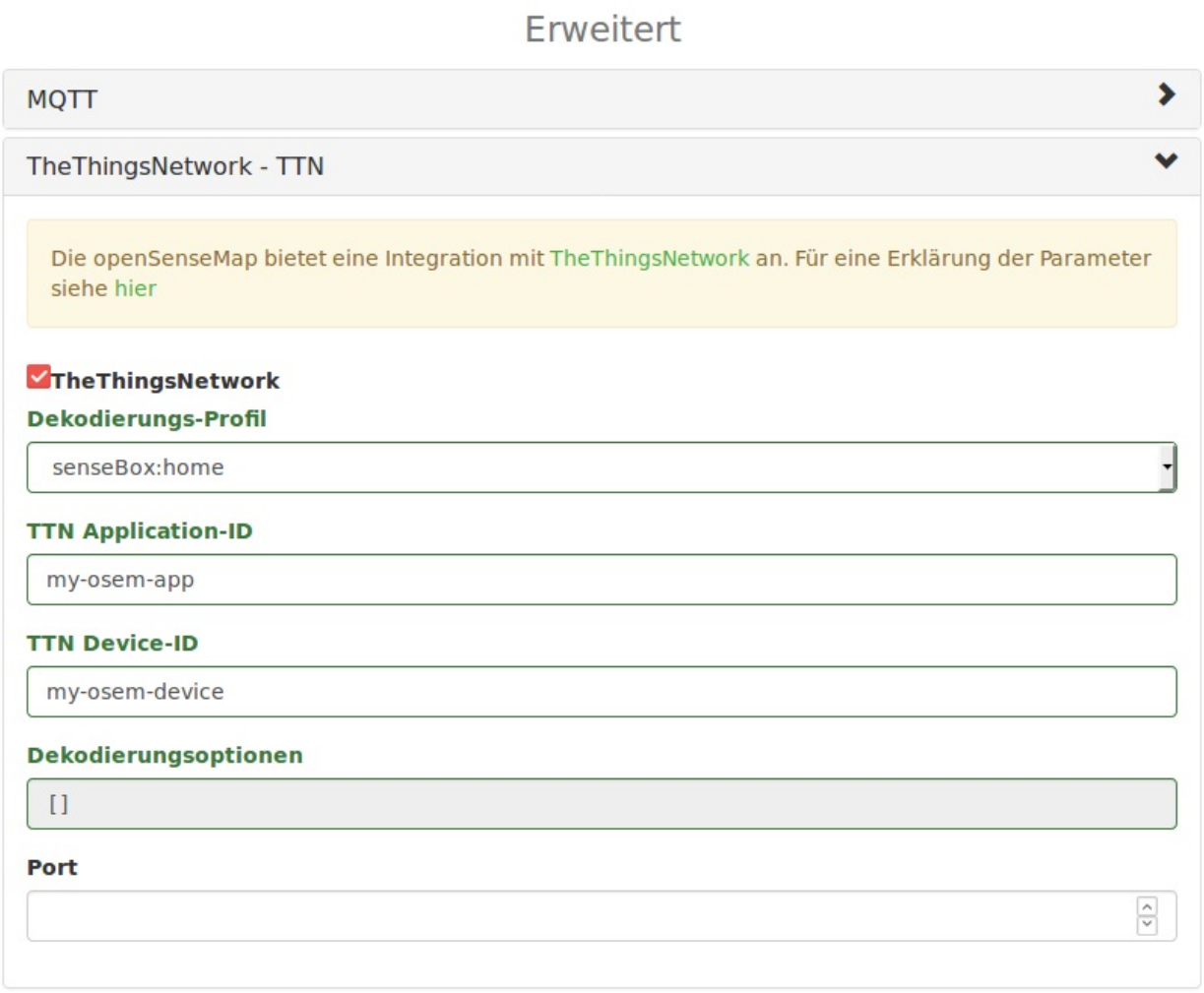
To integrate a device in to the TTN you have to first register an Application and a Device on the thethingsnetwork.org[1] Here you receive a `app_id` and a `dev_id`.

For the registered application, the HTTP integration must be activated under https://console.thethingsnetwork.org/applications/DEINE_APPID/integrations/create/http-ttn. To transmit messages from devices via `POST` to `https://ttn.opensensemap.org/v1.1`, you have to configurate this The authorization-field can stay empty.



ttnconsole

For the data transfer to openSenseMap, the `app_id` and `dev_id` must be included for the registration on openSenseMap in the TTN configuration. In addition, a suitable decoding profile must be configured. Which determines how the - because of the low bandwidth as raw bytes transmitted - data should be interpreted as measurements.



osemregister

Optionally you can indicate a port in the field `port`, on which the transmitter can send his data to the TTN. So you can use the same `app_id` and `dev_id` for multiple sensor stations.

Arduino Sketch

This could be an Arduino sketch that lets you send data to the openSenseMap over the TTN network.

> Important: You have to paste your recently crated Application-EUI, Device-EUI and the App-Key in the sketch. Please do this in the first line of the programmecode where 'INSERT YOUR ID HERE' is indicated.
>
> Mind that you have chosen the Device-EUI the Application-EUI the lsb-Format as well as the App-Key and the msb-Format on the TTN-Homepage.

Selected ID's and Keys

▼ **Arduino Sketch für senseBoxMCU**

```
/*******************************************************************************
 * Copyright (c) 2015 Thomas Telkamp and Matthijs Kooijman.
 * Edited by: senseBox
 *
 *******************************************************************************/
#include <LoraMessage.h>
#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>
#include <senseBoxIO.h>

#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <HDC100X.h>
#include <Makerblog_TSL45315.h>
#include <SDS011-select-serial.h>
#include <VEML6070.h>

// Number of serial port the SDS011 is connected to. Either Serial1 or Serial2
#define SDS_UART_PORT (Serial1)

//Load sensors / instances
Makerblog_TSL45315 TSL = Makerblog_TSL45315(TSL45315_TIME_M4);
HDC100X HDC(0x40);
Adafruit_BMP280 BMP;
VEML6070 VEML;
SDS011 SDS(SDS_UART_PORT);

bool hdc, bmp, veml, tsl = false;

//measurement variables
float temperature = 0;
float humidity = 0;
float pm10 = 0;
float pm25 = 0;
double tempBaro, pressure;
uint32_t lux;
uint16_t uv;

// This EUI must be in little-endian format, so least-significant-byte
// first. When copying an EUI from ttnctl output, this means to reverse
// the bytes. For TTN issued EUIs the last bytes should be 0xD5, 0xB3,
// 0x70.
static const u1_t PROGMEM APPEUI[8]={ 'Your APP ID Here' };
void os_getArtEui (u1_t* buf) { memcpy_P(buf, APPEUI, 8);}

// This should also be in little endian format, see above.
static const u1_t PROGMEM DEVEUI[8]={ 'YOUR DEVICE ID HERE '};
void os_getDevEui (u1_t* buf) { memcpy_P(buf, DEVEUI, 8);}

// This key should be in big endian format (or, since it is not really a
// number but a block of memory, endianness does not really apply). In
// practice, a key taken from ttnctl can be copied as-is.
// The key shown here is the semtech default key.
static const u1_t PROGMEM APPKEY[16] = { 'YOUR APP KEY HERE '};
void os_getDevKey (u1_t* buf) {  memcpy_P(buf, APPKEY, 16);}

static osjob_t sendjob;
```

```
// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).
const unsigned TX_INTERVAL = 300;

// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = PIN_XB1_CS,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = LMIC_UNUSED_PIN,
    .dio = {PIN_XB1_INT, PIN_XB1_INT, LMIC_UNUSED_PIN},
};

void checkI2CSensors() {
  byte error;
  int nDevices = 0;
  byte sensorAddr[] = {41, 56, 57, 64, 118};
  tsl = false; veml = false; hdc = false; bmp = false;
  Serial.println("\nScanning...");
  for (int i = 0; i < sizeof(sensorAddr); i++) {
    Wire.beginTransmission(sensorAddr[i]);
    error = Wire.endTransmission();
    if (error == 0) {
      nDevices++;
      switch (sensorAddr[i])
      {
        case 0x29:
          Serial.println("TSL45315 found.");
          tsl = true;
          break;
        case 0x38: // &0x39
          Serial.println("VEML6070 found.");
          veml = true;
          break;
        case 0x40:
          Serial.println("HDC1080 found.");
          hdc = true;
          break;
        case 0x76:
          Serial.println("BMP280 found.");
          bmp = true;
          break;
      }
    }
    else if (error == 4)
    {
      Serial.print("Unknown error at address 0x");
      if (sensorAddr[i] < 16)
        Serial.print("0");
      Serial.println(sensorAddr[i], HEX);
    }
  }
  if (nDevices == 0) {
    Serial.println("No I2C devices found.\nCheck cable connections and press Reset.");
    while(true);
  } else {
    Serial.print(nDevices);
    Serial.println(" sensors found.\n");
  }
  //return nDevices;
}

void onEvent (ev_t ev) {
    senseBoxIO.statusGreen();
    Serial.print(os_getTime());
    Serial.print(": ");
    switch(ev) {
        case EV_SCAN_TIMEOUT:
            Serial.println(F("EV_SCAN_TIMEOUT"));
            break;
        case EV_BEACON_FOUND:
            Serial.println(F("EV_BEACON_FOUND"));
            break;
```

```
        case EV_BEACON_MISSED:
            Serial.println(F("EV_BEACON_MISSED"));
            break;
        case EV_BEACON_TRACKED:
            Serial.println(F("EV_BEACON_TRACKED"));
            break;
        case EV_JOINING:
            Serial.println(F("EV_JOINING"));
            break;
        case EV_JOINED:
            Serial.println(F("EV_JOINED"));

            // Disable link check validation (automatically enabled
            // during join, but not supported by TTN at this time).
            LMIC_setLinkCheckMode(0);
            break;
        case EV_RFU1:
            Serial.println(F("EV_RFU1"));
            break;
        case EV_JOIN_FAILED:
            Serial.println(F("EV_JOIN_FAILED"));
            break;
        case EV_REJOIN_FAILED:
            Serial.println(F("EV_REJOIN_FAILED"));
            break;
            break;
        case EV_TXCOMPLETE:
            Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
            if (LMIC.txrxFlags & TXRX_ACK)
              Serial.println(F("Received ack"));
            if (LMIC.dataLen) {
              Serial.println(F("Received "));
              Serial.println(LMIC.dataLen);
              Serial.println(F(" bytes of payload"));
            }
            // Schedule next transmission
            os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
            break;
        case EV_LOST_TSYNC:
            Serial.println(F("EV_LOST_TSYNC"));
            break;
        case EV_RESET:
            Serial.println(F("EV_RESET"));
            break;
        case EV_RXCOMPLETE:
            // data received in ping slot
            Serial.println(F("EV_RXCOMPLETE"));
            break;
        case EV_LINK_DEAD:
            Serial.println(F("EV_LINK_DEAD"));
            break;
        case EV_LINK_ALIVE:
            Serial.println(F("EV_LINK_ALIVE"));
            break;
         default:
            Serial.println(F("Unknown event"));
            break;
    }
}

void do_send(osjob_t* j){
    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    } else {
        LoraMessage message;

        //-----Temperature-----//
        //-----Humidity-----//
        if (hdc) {
          Serial.print("Temperature: ");
          temperature = HDC.getTemp();
```

```
          Serial.println(temperature);
          message.addUint16((temperature + 18) * 771);
          delay(2000);

          Serial.print("Humidity: ");
          humidity = HDC.getHumi();
          Serial.println(humidity);
          message.addHumidity(humidity);
        }
        delay(2000);

        if (bmp) {
          float altitude;
          tempBaro = BMP.readTemperature();
          pressure = BMP.readPressure()/100;
          altitude = BMP.readAltitude(1013.25); //1013.25 = sea level pressure
          Serial.print("Pressure: ");
          Serial.println(pressure);
          message.addUint16((pressure - 300) * 81.9187);
          delay(2000);
        }

        if (tsl) {
          //-----Lux-----//
          Serial.print("Illuminance: ");
          lux = TSL.readLux();
          Serial.println(lux);
          message.addUint8(lux % 255);
          message.addUint16(lux / 255);
          delay(2000);
        }

        if (veml) {
          //-----UV intensity-----//
          Serial.print("UV: ");
          uv = VEML.getUV();
          Serial.println(uv);
          message.addUint8(uv % 255);
          message.addUint16(uv / 255);
          delay(2000);
        }

        uint8_t attempt = 0;

        while (attempt < 5) {
          bool error = SDS.read(&pm25, &pm10);
          if (!error) {
            Serial.print("PM10: ");
            Serial.println(pm10);
            message.addUint16(pm10 * 10);
            Serial.print("PM2.5: ");
            Serial.println(pm25);
            message.addUint16(pm25 * 10);
            break;
          }
          attempt++;
        }

        // Prepare upstream data transmission at the next possible time.
        LMIC_setTxData2(1, message.getBytes(), message.getLength(), 0);
        Serial.println(F("Packet queued"));
    }
    // Next TX is scheduled after TX_COMPLETE event.
}

void setup() {
    Serial.begin(9600);
    delay(10000);

    // RFM9X (LoRa-Bee) in XBEE1 Socket
    senseBoxIO.powerXB1(false); // power off to reset RFM9X
    delay(250);
```

```
    senseBoxIO.powerXB1(true);  // power on

    // init I2C/wire library
    Wire.begin();

    // Sensor initialization
    Serial.println(F("Initializing sensors..."));
    SDS_UART_PORT.begin(9600);
    checkI2CSensors();

    if (veml)
    {
      VEML.begin();
      delay(500);
    }
    if (hdc)
    {
      HDC.begin(HDC100X_TEMP_HUMI, HDC100X_14BIT, HDC100X_14BIT, DISABLE);
      HDC.getTemp();
    }
    if (tsl)
    {
      TSL.begin();
    }
    if (bmp)
    {
      BMP.begin(0x76);
    }
    Serial.println(F("Sensor initializing done!"));
    Serial.println(F("Starting loop in 3 seconds."));
    delay(3000);

    // LMIC init
    os_init();
    // Reset the MAC state. Session and pending data transfers will be discarded.
    LMIC_reset();

    // Start job (sending automatically starts OTAA too)
    do_send(&sendjob);
}

void loop() {
    os_runloop_once();
}
```

Decoding Profile

A decoding-profile fitting to the measuring data has to be selected and defined for a box. The decoding profile selection is based on the encoding of the messages on the Microcontroller. And whether in the TTN a payload function has been set dependent.

- For the senseBox: home (without extensions) the `senseBox: home` profile be used.
- If the measurements will be encoded on the LoRa node using the `lora-serialization` library , the `lora-serialization` profile should be used.
- The `json` profile supports any other encodings, if one Payload function in the TTN Console decodes the messages appropriately.

The following explains how to configure the supported profiles:

`sensebox/home`

This profile is tailored to the sensors supplied with the senseBox: home. Besides the specification `sensebox / home` under `profile` there is no further configuration necessary.
This works only without the fine dust sensors(PM2.5 und PM10)

In addition to the Arduino Sketch, you'll need to set up a decoder on the TTN homepage so that your metrics are sent to the openSenseMap in the correct format.

Navigate to Payload Formats in the Overview window

The decoder must now be inserted in the text box

▼ **Decoder für das TTN**

Important: Here you have to add your sensor ID's.

```
function Decoder(bytes, port) {
  // bytes is of type Buffer.
  'use strict';
  var TEMPSENSOR_ID = 'YOUR TEMPERATURE SENSOR ID HERE',
    HUMISENSOR_ID = 'YOUR HUMIDITY SENSOR ID HERE',
    PRESSURESENSOR_ID = 'YOUR PRESSURE SENSOR ID HERE ',
    LUXSENSOR_ID = 'YOUR LUXSENSOR ID HERE ',
    UVSENSOR_ID = 'YOUR UV SENSOR ID HERE';

  var bytesToInt = function (bytes) {
    var i = 0;
    for (var x = 0; x < bytes.length; x++) {
      i |= +(bytes[x] << (x * 8));
    }
    return i;
  };

  var uint8 = function (bytes) {
    if (bytes.length !== uint8.BYTES) {
      throw new Error('int must have exactly 1 byte');
    }
    return bytesToInt(bytes);
  };
  uint8.BYTES = 1;

  var uint16 = function (bytes) {
    if (bytes.length !== uint16.BYTES) {
      throw new Error('int must have exactly 2 bytes');
    }
    return bytesToInt(bytes);
  };
  uint16.BYTES = 2;

  var humidity = function (bytes) {
    if (bytes.length !== humidity.BYTES) {
      throw new Error('Humidity must have exactly 2 bytes');
    }

    var h = bytesToInt(bytes);
    return h / 1e2;
  };
  humidity.BYTES = 2;
```

```
var decode = function (bytes, mask, names) {

  var maskLength = mask.reduce(function (prev, cur) {
    return prev + cur.BYTES;
  }, 0);
  if (bytes.length < maskLength) {
    throw new Error('Mask length is ' + maskLength + ' whereas input is ' + bytes.length);
  }

  names = names || [];
  var offset = 0;
  return mask
    .map(function (decodeFn) {
      var current = bytes.slice(offset, offset += decodeFn.BYTES);
      return decodeFn(current);
    })
    .reduce(function (prev, cur, idx) {
      prev[names[idx] || idx] = cur;
      return prev;
    }, {});
};

var bytesToSenseBoxJson = function (bytes) {
  var json;

  try {
    json = decode(bytes,
      [
        uint16,
        humidity,
        uint16,
        uint8,
        uint16,
        uint8,
        uint16
      ],
      [
        TEMPSENSOR_ID,
        HUMISENSOR_ID,
        PRESSURESENSOR_ID,
        LUXSENSOR_ID + '_mod',
        LUXSENSOR_ID + '_times',
        UVSENSOR_ID + '_mod',
        UVSENSOR_ID + '_times'
      ]);

    //temp
    json[TEMPSENSOR_ID] = parseFloat(((json[TEMPSENSOR_ID] / 771) - 18).toFixed(1));

    //hum
    json[HUMISENSOR_ID] = parseFloat(json[HUMISENSOR_ID].toFixed(1));

    // pressure
    if (json[PRESSURESENSOR_ID] !== '0') {
      json[PRESSURESENSOR_ID] = parseFloat(((json[PRESSURESENSOR_ID] / 81.9187) + 300).toFixed(1));
    } else {
      delete json[PRESSURESENSOR_ID];
    }

    // lux
    json[LUXSENSOR_ID] = (json[LUXSENSOR_ID + '_times'] * 255) + json[LUXSENSOR_ID + '_mod'];
    delete json[LUXSENSOR_ID + '_times'];
    delete json[LUXSENSOR_ID + '_mod'];

    // uv
    json[UVSENSOR_ID] = (json[UVSENSOR_ID + '_times'] * 255) + json[UVSENSOR_ID + '_mod'];
    delete json[UVSENSOR_ID + '_times'];
    delete json[UVSENSOR_ID + '_mod'];

  } catch (e) {
    json = { payload: bytes };
```

```
    }

    return json;
  };

  return bytesToSenseBoxJson(bytes);
}
```

`lora-serialization`

The `lora-serialization` profile can accept almost any data, even sensor stations, which have a special sensor configuration. For this we use the [`lora-serialization`] (https://github.com/thesolarnomad/lora-serialization) Library, which provides a unified encoding on the microcontroller, and Decoding on the other end of the line.

The encodings `temperature` , `humidity` , `unixtime` , `uint8` and `uint16` are supported, which need to be indicated per sensor under decoding options . The assignment of the sensor can be made via one of the properties `sensor_id` , `sensor_title` , `sensor_unit` , `sensor_type` .

An example of two sensors looks like this:

```
[
  { "decoder": "temperature", "sensor_title": "Temperatur" },
  { "decoder": "humidity", "sensor_unit": "%" }
]
```

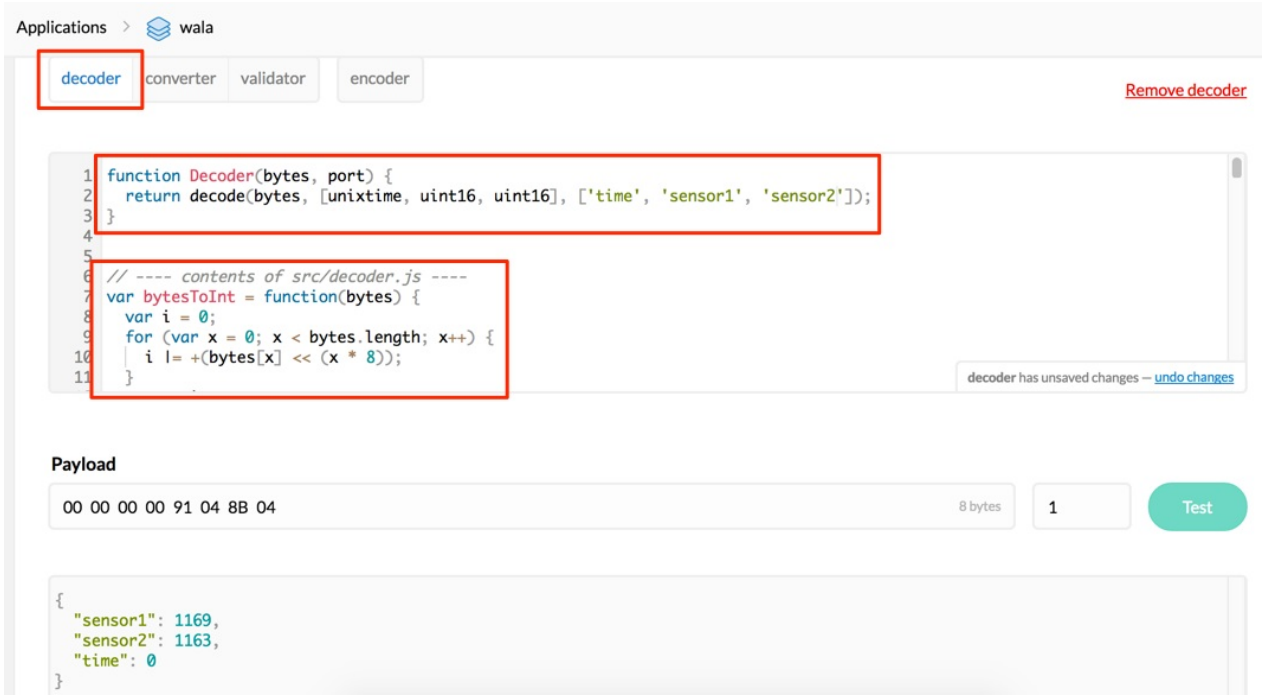
> Information: The order of the sensors must be the same here as well as on the Arduino and the >openSenseMap!

If a `unixtime` decoder is specified, its timestamp will be used for all of the following measurements. Otherwise, the moment is used in which the first gateway receives the message. Example:

```
[
  { "decoder": "unixtime" },
  { "decoder": "temperature", "sensor_title": "Temperatur" }
]
```

`json` - Decoding mit TTN Payload Function

If the `lora-serialization` library is not available, measurements can still get decoded on the TTN side by means of a Payload Function , so that any data formats are supported here.



In the TTN Console, a payload function must be defined

The resulting JSON must be compatible with that of theopenSenseMap-API verstandenen Measurement Formaten sein. A simple example:

```
{ "sensor_id1": "value1, "sensor_id2: "value2" }
```

An example of this is indicated for you you above[1].

On the side of the openSenseMap no configuration is necessary.

---

1. https://console.thethingsnetwork.org/ ↩

1. See *3.1.2.4 LoRa-Bee* > *decoder* ↩

Sensors

With sensors you can measure different environmental phenomena and thus recognize, observe and analyze.

In our senseBox-Shop[1] you will find a list of sensors we work with and therefore have written small instructions. For these sensors we offer you here information, but also help with connecting to the senseBox.

Here you will find information about the following sensors, just click on the name of the sensor and you will be redirected to a separate page

- Temperature & air humidity (HDC1080)[1]
- Air pressure & temperature[2]
- Exposure and UV[3]
- Fine dust[4]

Of course, you can also connect any other sensor you know to the senseBox. However then you are asked as a tinkerer to figure out the wiring and programming ;) We would love to see you sharing information about your sensor and how you connected it to your senseBox so we can add it to our instruction.

---

1. https://sensebox.kaufen/ ↩

1. See *3.1.3.1 Temperature- and Air Humidity Sensor (HDC1080)* ↩

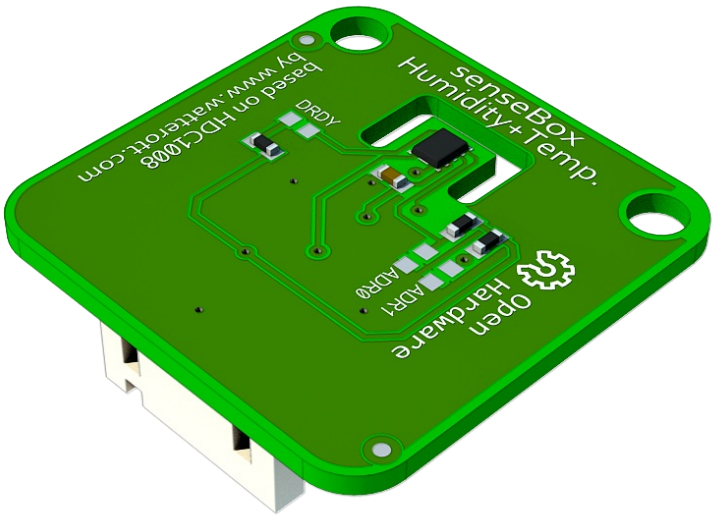
2. See *3.1.3.2 Airpressure- and Temperature Sensor* ↩

3. See *3.1.3.3 Illumination and UV-Radiation Sensor* ↩

4. See *3.1.3.4 Fine Dust Sensor* ↩

Temperature- and Air Humidity Sensor (HDC1080)

The HDC1080 is a digital humidity and temperature sensor. The sensor has a high accuracy and a very low power consumption and thus fits perfectly with the senseBox. The sensors are factory calibrated and can be used directly.



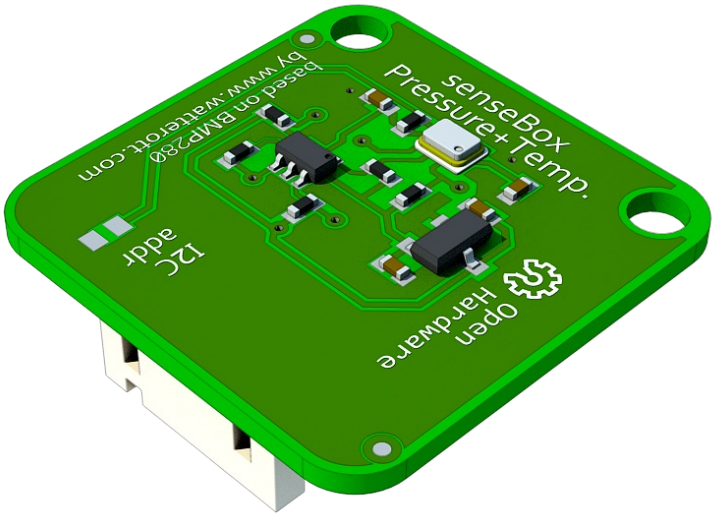
Temperature and Air Humidity Sensor

Technical Details

- Relative Humidity (RH) Operating range 0% to 100%
- 14-bit Measurement Resolution
- Relative humidity accuracy ± 4%
- Temperature accuracy ± 0.2 ° C
- 2100nA Sleep Mode Current
- Operating voltage 2.7 V to 5.5 V
- I2C interface
- "Plug-in-and-Go" senseBox compatible
- Average Supply Current: 710nA @ 1sps, 11bit RH Measurement 1.3µA @ 1sps, 11bit RH and Temperature Measurement

Airpressure- and Temperature Sensor

This sensor measures air pressure and temperature and is based on the BMP280 sensor from Bosch.



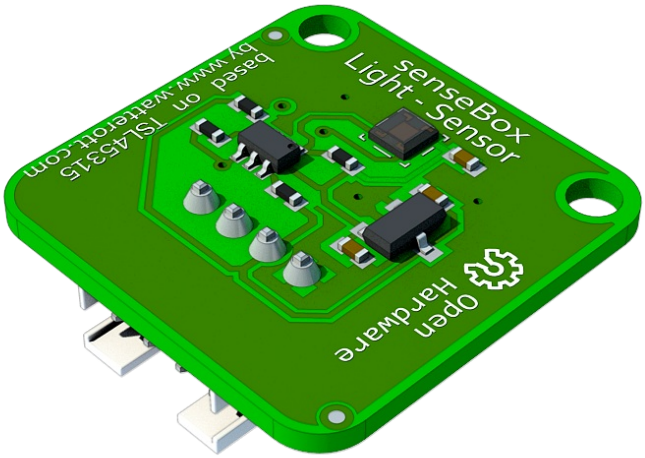
Airpressure- and Temperature Sensor

Technical Detail

- Dimensions: 25mm x 25mm x 9mm
- Weight: 2.4 g
- "Plug-in-and-Go" senseBox compatible
- Operating pressure 300 to 1100 hPa
- Relative precision ± 0.12 hPa
- Absolute precision ± 1 hPa
- Operating power 2.7µA at 1Hz sampling frequency

Illumination and UV-Radiation Sensor

Two sensors are put together on this senseBox component. The light intensity is measured with the TSL45315 sensor from AMS-TAOS. This sensor detects the light conditions similar to the human eye and outputs the brightness values directly in lux, with a large dynamic range (3 lux to 220k lux). The second sensor is a Vishay VEML6070 Ultraviolet (UV) light sensor. This converts the intensity of the UV light of the sun into digital data. The sensor has excellent UV sensitivity and linearity via Filtron ™ technology. It has a good UV radiation measurement even with long solar UV exposure and can compensate for excellent temperature fluctuations.



Illumination and UV-Radiation

Technical Details

Exposure Sensor

- 3,3V - 5V tolerant I2C/TWI Interface
- Input voltage range: 3,3V - 5V
- On-board 2,5V voltage regulator
- On-board levelconverter

UV-Sensor

- Operating voltage: 2,7V - 5,5V I2C Interface
- Supports confirmation function (Active Acknowledge-Function)
- Temperature compensation: -40°C to +85°C
- Software-switching of control for immunity with flickering fluorescent lamps

Measurement

- 25mm x 25mm x 9mm
- Weight: 2,5 g

Fine Dust Sensor

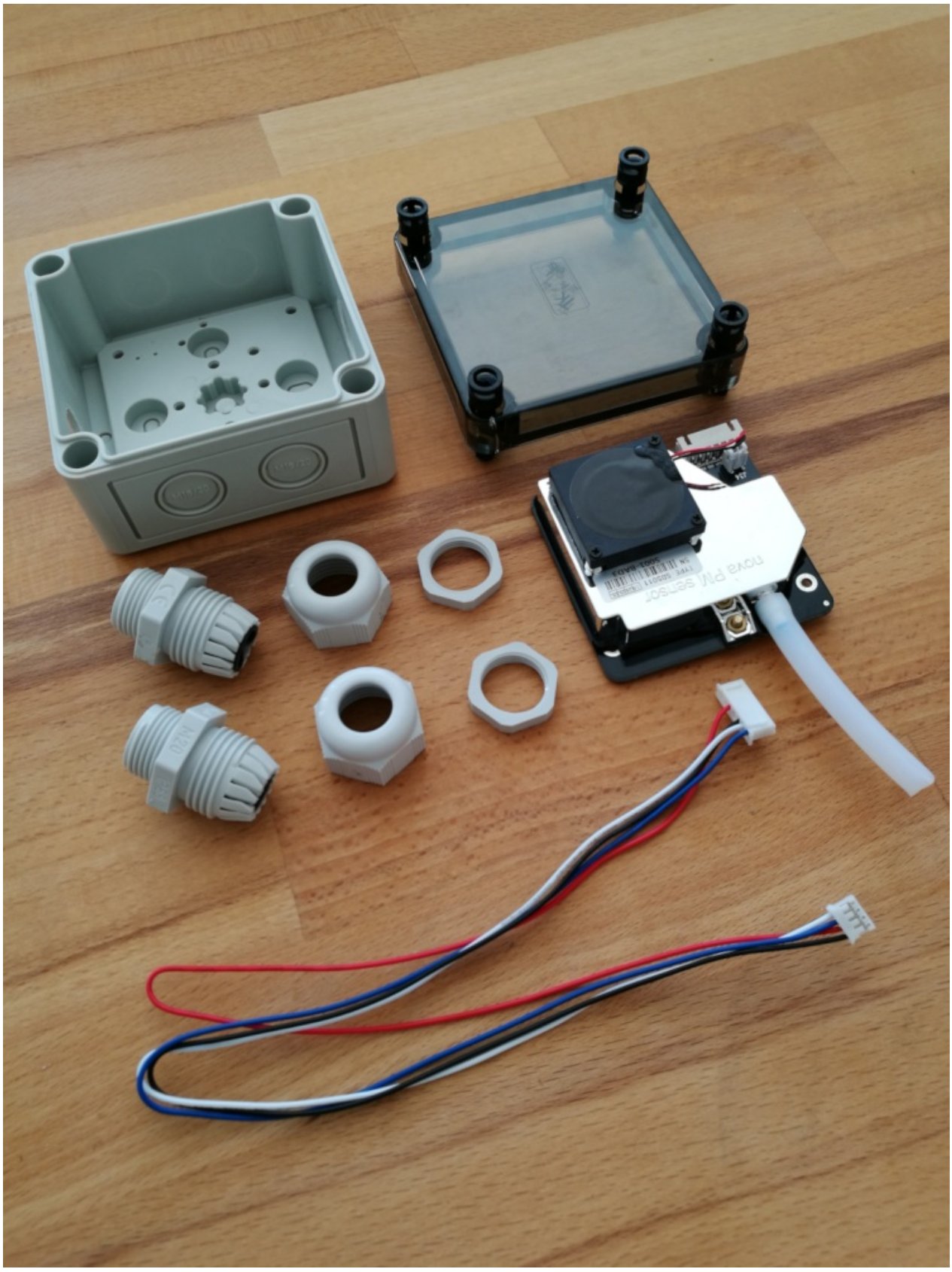
With this sensor SDS011 it is possible to determine the fine dust concentration in the air. The sensor outputs two values: the concentration of PM2.5 (particle < 2.5 µm) and PM10 (particle < 10 µm). This sensor is equipped with a small fan to suck in air. Inside is a laser that measures the number of particles together with a photodiode. The results of the measurements are given in µg / m³ (micrograms per cubic meter)

Fine dust sensor for PM10 und PM2.5

Technical Details

- Fast reaction time less than 10 seconds
- "Plug-in-and-Go" senseBox compatibel
- High resolution up to 0.3µg/m3
- Multiple scientific verification of data accuracy

Parts required for the Set-Up



Required (provided) Parts

- SDS011 fine dust sensor
- connecting wire

- an unit of teflon tube Ø = 6mm inside and Ø = 8mm outside
- Case
- Cable gland 16mm

Connection and Programming

With the supplied connection cable you can connect your fine dust sensor with the "UART / Serial" port of the senseBoxMCU. Once this is done, we can now initialize the sensor in the program code and have the first measured values output

> Make sure you have the latest board support package installed because you need the correct software libraries. How this works was shown in Step 2.

First, create an instance of the sensor. For this we create 2 variables in which we save our two readings for PM10 and PM2.5.

```
#include "SenseBoxMCU.h"
SDS011 my_sds(Serial1) // Serial1 indicates the serial port where you connected the sensor
float p10,p25
```

▼ setup() Funktion

The sensor should now start in the `setup()` -Function:

```
void setup(){
    // Initialise normal seriel Port
    Serial.begin(9600);
    while(!Serial);
    // Initialize the serial port where our sensor is connected
    Serial1.begin(9600);
    delay(5000);

}
```

▼ loop() Funktion

In the `loop ()` function we can use the command 'getPm10 ()' and 'getPm25 () to retrieve the currently measured fine dust values:

```
void loop(){
    // Assign variables to measured particulate matter values
    p10 = my_sds.getPm10();
    p25 = my_sds.getPm25();
    //Print values in the console
    Serial.println("P2.5: "+String(p25));
    Serial.println("P10:  "+String(p10));
    delay(1000);
}
```

Accessories

Here you can find all the additional accessories-parts which cannot be assigned to sensors or Bees. Including parts, such as a JST expander, which is not included in the senseBox by default. Nevertheless the expander is offered and supported by us. So do not worry if you do not find something in your senseBox right now, look into the inventory.

Overview

There are the following components offered and tested for you with the senseBox:

- Radiation protection[1]
- Housing[2]
- AC adapter and USB cable[3]
- LED display[4]
- HUB[5]
- Micro SD card[6]
- GPS[7]

---

[1]. See *3.1.4.1 Radiation Protection* ↩

[2]. See *3.1.4.2 Housing* ↩

[3]. See *3.1.4.3 Power Supply and USB-Cable* ↩

[4]. See *3.1.4.4 LED-Display* ↩

[5]. See *3.1.4.5 Expander* ↩

[6]. See *3.1.4.6 Micro-SD Karte* ↩

[7]. See *3.1.4.7 GPS* ↩

Radiation Protection

The radiation protection protects the outdoor transmitter for temperature + humidity (BMP280) from the weather. It serves as protection against precipitation and direct sunlight. It is easy to mount and has an opening for the sensor cables.

Advice for the Location of the Radiation Protection

- Choose a shady spot outdoors for radiation protection . (Direct sunshine falsifies the measured values).
- Please check wheter a transmission from the transmitter to the desired installation location is possible (in massive walls, especially with metal parts, the transmission range can get considerably reduced).

Mounting of the Protective Cover

- Turn off the protective cover clockwise from the bottom plate .
- The protective cover can be be screwed on a suitable surface for a secure fit.
- Choose a smooth, horizontal position.
- Pass a screw through the opening in the middle and screw the floorplate firmly.

Attachment to a Wall or to a Pole

- Remove the wall bracket from the case by sliding it down.
- Attach the wall bracket to the wall with the screws and dowels. Please mind the marking "UP"
- If you want to attach the wall bracket to a pole, you can also use the cable ties.
- Note: When mounting, please make sure that the protective cover can be easily inserted into the Wall bracket
- Put the protective cover with the base plate and the attached transmitter in the wall bracket from above.

Attachment of the Transmitter

- To secure the transmitter to the base plate, attach the Velcro strip with the double-sided tape to the base on the stand and on the back of the transmitter and fix the transmitter. Alternatively, you can also use the cable ties or work with hot glue.
- Guide the cables through the provided opening in the bottom plate.
- Place the protective cover on the base plate with the attached transmitter and turn it counterclockwise.

Care and Maintenance

- Clean the protective cover with a soft, slightly damp cloth. Do not use scouring or solvents.

Technical Data

Interior dimensions: Höhe 160 mm Interior diameter: 60 mm Case measurements: 95 x 102 (108) x 180 mm Weight: 163 g

Housing

The housing protects the microcontroller and some sensors from the weather.

The Housing

The transparent polycarbonate housing protects the senseBox MCU and the belonging sensors from the weather.

Measurements: 82mm x 162mm x 85mm Weight: 300 g

Power Supply and USB-Cable

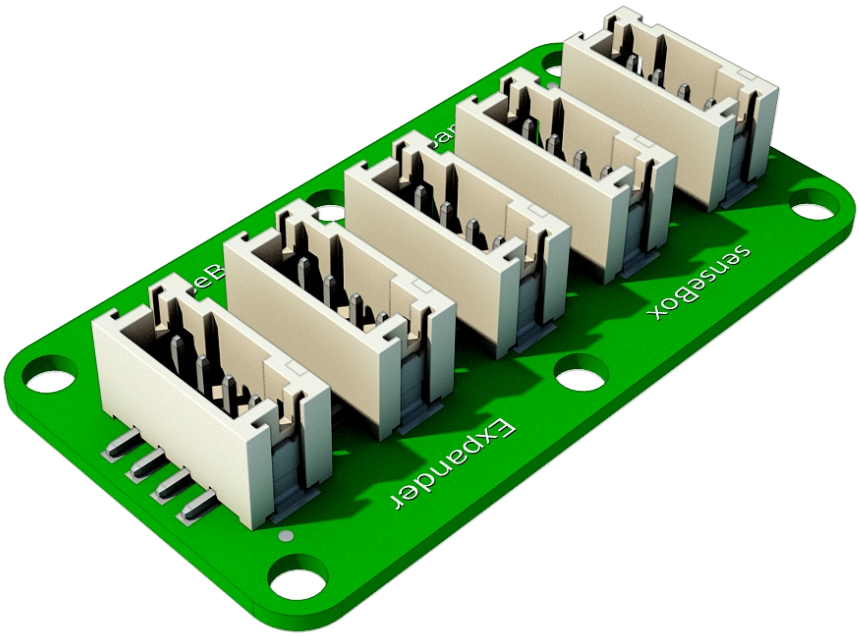
The senseBox is connected to a conventional micro USB cable and connected to the computer or to a power source

USB-cable

With the enclosed USB cable, the senseBox can be configured and connected to the computer. The power supply ensures the continuous power supply. The enclosed cable has a length of 1.5 meters.

LED-Display

Expander

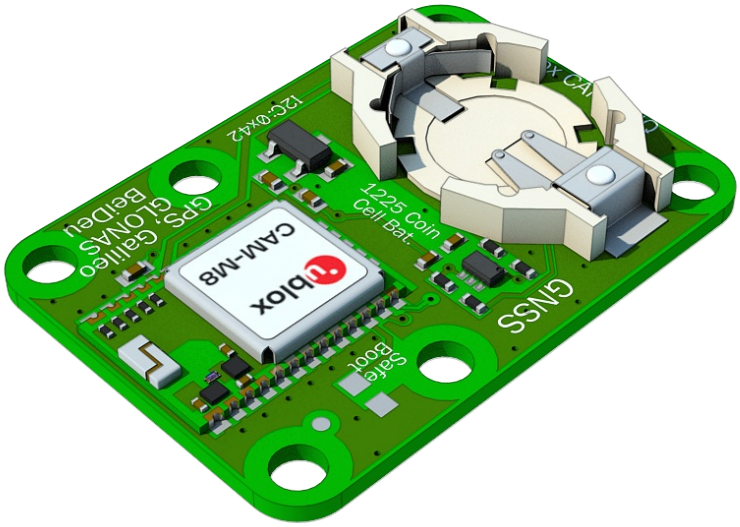


Expander

Micro-SD-Karte

GPS

Sorry, unfortunately we have not yet managed to document the GPS sensor. We hope to implement this as soon as possible



The GPS-sensor

Help

Here you can find frequently asked questions and answers. You can first take a look here and see if your question can be answered. If not have a look in our Forum and ask a question.

Error compiling or transferring code in Arduino

Errors that occur by compiling or transferring codes are often caused by simple mistakes which are easy to fix. Often a `;` `(Semikolon)` a `} (geschweifte Klammer)` or other small mistakes appear. But the hardest part of troubleshooting is not fixing the error, but finding the error or source of it. To help find the bug, the Arduino IDE gives you an error message. These are, however, without previous experience with programming, often more cryptic than the actual error.

You will therefore find some common errors along with the error message and instructions for correcting the error. If you can not solve your problem with these, look in the forum and create there if necessary a contribution with the description of your error. For this you should provide the best error message and your sketch.

> Of course, if you publicly present your sketch (for example, in our forum), you should always make sure that you do not disclose personal information to the public. Therefore, you should first look in the code, if you have your Wifi name (SSID) and the password built into the sketch. If this is the case, please delete it from the sketch before publishing it on the internet.

▼ I get an error when I connect the fine dust sensor. But I did not change the sketch, what is wrong here?

First, look at the error message. The message can be found in the Arduino IDE below your sketch, in a separate area. Here is your error message. You can scroll through this message with your mouse or drag the area around to see the entire error message.

Even if you have the sketch directly from the openSenseMap without changig it, it can happen that an error with a similar or identical error message like the following occurs:

```
'undefined' was not declared in this scope

#define SDS_UART_PORT (undefined)
                       ^
 /Users/user-name/Documents/Arduino/libraries/sketch_jun27a/example_sketch123.ino:77:12: note: in expansion of macro
'SDS_UART_PORT'
   SDS011 SDS(SDS_UART_PORT);
              ^
 /Users/user-name/Documents/Arduino/libraries/example_sketch123/example_sketch123.ino: In function 'void setup()':
example_sketch123:247: error: 'undefined' was not declared in this scope
     undefined.begin(9600);
     ^
exit status 1
'undefined' was not declared in this scope
```

The error is caused here by

```
#define SDS_UART_PORT (undefined)
```

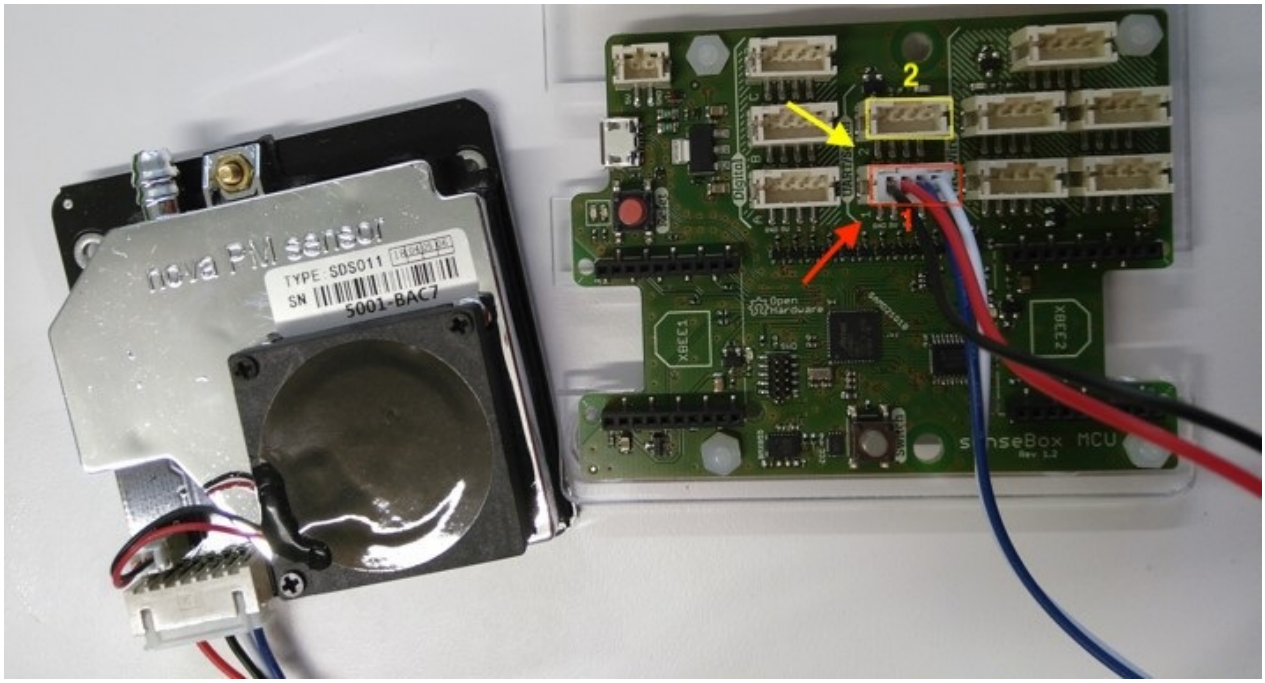
and

```
undefined.begin(9600);
```

An error in the configuration has crept in here, because where `undefined` is displayed should stand `serial1` or `serial2`. The senseBox MCU gets here from your sketch the information to which port the fine dust sensor was connected. However, the senseBox MCU can not handle the information `undefined` and thus does not know which port the fine dust sensor sends the measured values to.

Please follow the steps to solve the problem:

- Look where the cable from the fine dust sensor is connected at your senseBox MCU.



Look at which of the two color-coded ports you've connected your fine dust sensor.

- If you have connected the fine dust sensor like in the picture to the serial port 1 (red marked port), you can exchange the `undefined` in the sketch with `Serial1`.
- If you have connected your fine dust sensor besides to serial port 2 (yellow marked in the picture), you can exchange the `undefined` in the sketch with `Serial1`
- Please conduct the change in both lines. So, at the point where `#define SDS_UART_PORT (undefined)` is in the text and where `undefined.begin (9600);` stands.
- Then compile the sketch again.

Your problem should be solved - Have fun with your fine dust sensor.

Please visit the Forum[1] if you have any problems with this instruction.

Questions about Programming

▼ I would like to include an extern library. Is it possible?

Yes it is. The senseBox is not limited to the included sensors. You can extend it with any sensors. However externl libraries from the suppliers are often needed.

This help website[1]! explains how to integrate them manually in the Arduino IDE.

Questions about the Data transmission

▼ I have problems with the transmission of data via wifi. What can I do?

It is possible with some of our Wifi Bees of the type WINC1500 that an old firmware (Version 19.4.4) is installed. This can lead to problems with the data transmission. If this is the case, please visit this help website[2] to refresh the firmware.

Questions to the Connection between the senseBox and the Computer

▼ I have a problem with the connection of my sensBox MCU and my windows-Computer. What can I do?

On some of our Windows computer, it may happen that the USB bootloader drivers are not installed correctly. This can lead to the computer not recognizing the senseBox MCU as a USB device and is therefore unable to transfer files. If you encounter these problems, check this help page[3] to see if your drivers work and update them if necessary.

Questions about the senseBox-Project

▼ Where can I buy the new senseBox version 2

That is easy, visit sensebox.kaufen and configurate the senseBox fitting to your needs.

▼ How can I ensure to be at the latest status considering the senseBox?

Just sign up to our newsletter and never miss any news.

---

1. https://forum.sensebox.de ↩

1. See *4.3 Manual Integration of Libraries* ↩

2. See *4.4 Firmware Update Wifi-Bee* ↩

3. See *4.5 Update Windows USB Bootloader Driver* ↩

Downloads

In this section you will find various downloads that help you to use the senseBox.

Documentation as PDF

This book is also available as PDF for printing.

PDF-Download: senseBox:home-Documentation[1]

- More useful downloads coming soon;) *

[1]. https://github.com/sensebox/books-v2/raw/gh-pages/senseBox:home.pdf ↩

Manual Integration of Libraries

To be able to connect own sensors to the senseBox, the manufacturers of many sensors provide suitable libraries. Here we show you how to download libraries from a Github repository and integrate them manually. You can use this guide for any external libraries you want to include in Arduino.

ATTENTION: The libraries required for the senseBox are already in the board support package, which is downloaded in Step 2. You should not longer integrate them manually like it was suggested in the previous books. This creates duplications that can lead to errors.

Download and add Libraries

▼ 'Library' - What is it and why do I need it?

As the name suggests, a library is a collection of something - a collection of methods to be more specific. Methods are programming smaller sections of code that can be applied to an object. For example, with the senseBox, a method can be invoked to turn the LEDs on and off on the MCU. There are a lot of such standard methods that are used by a variety of programs. In order not to have to transfer these methods individually into the program code, they can be stored in libraries. So a library is a file that stores many methods. You can include libraries in your code. For this it is enough if they are stored in the Arduino folder for libraries and then they are integrated with a single line at the beginning of the program code. This is how it looks like in Arduino for the library named "senseBoxIO":

```
#include <senseBoxMCU.h>;
```

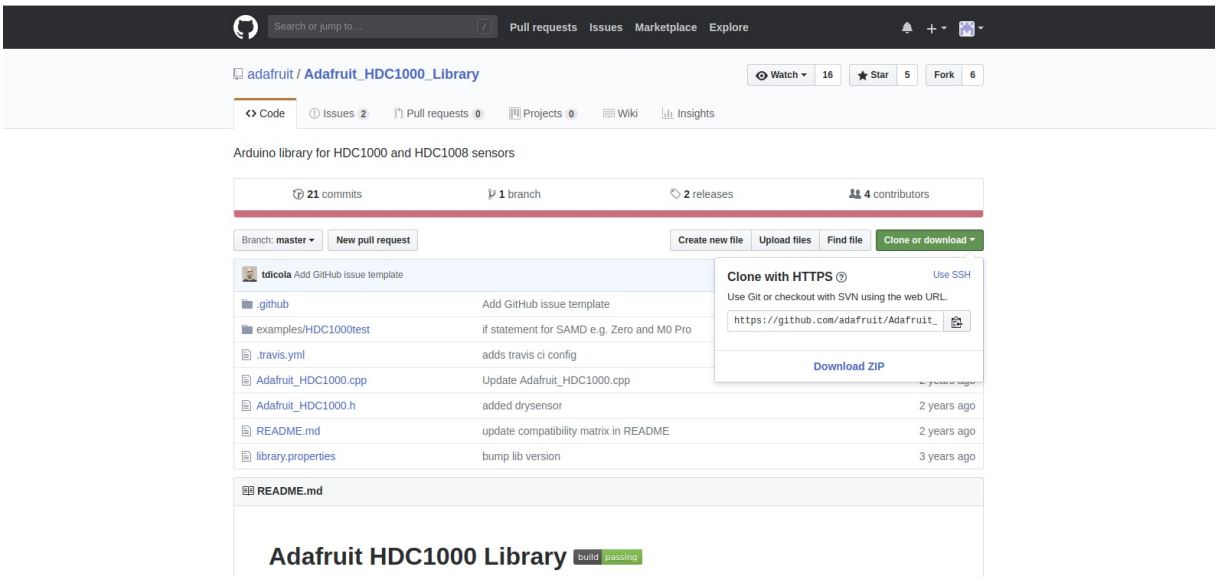
If the library is included, all methods contained in it can be used in the code.

The manual installation of the libraries can cause errors very quickly, so be sure to pay close attention to each step. To help you with the installation as well as possible, we have written a separate manual for each operating system. Choose the right system for your computer and follow the steps given.

- Libraries einfügen Windows
- Libraries einfügen Mac
- Libraries einfügen Linux

Insert Libraries Windows

1. Most external libraries can be found in Github repositories. To download them, you have to click the green button `Clone or download` and click `Download ZIP` afterwards.

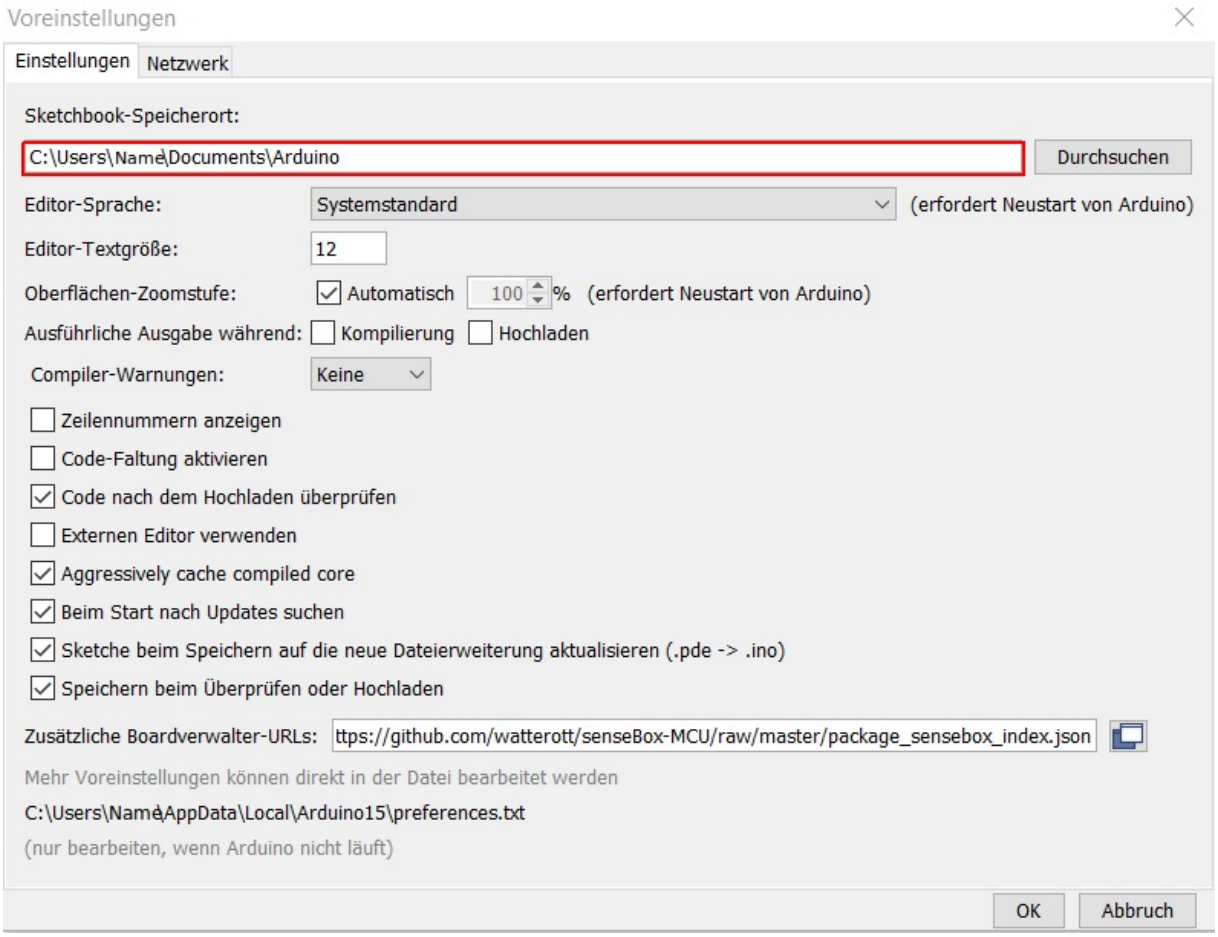


*Example download of the *

2. If the download does not start on its own, a window opens in which you have to select the field `Save file` and place the folder anywhere on your computer (this is the Downloads folder by default).
3. The downloaded file is a `.zip` archive, which is a compressed version of the library. The next step is to unpack this `.zip` archive. To do this, open the archive location and right-click it and select 'Extract All ...' in the menu that appears. Select the same folder as the download location (for example, the Downloads folder).
4. Now open the Arduino IDE. Go to `File` -> `Preferences` :

Click `File` and then` Preferences`

and look in the box under ` Sketchbook location 'for the location of the Sketchbook folder.



Look in the red-marked box to see where your sketchbook location is

Remember the path to this folder, ie the location where it is stored.

> You need to move the library folder you have already downloaded and unpacked to the sketchbook location in the next step. It is therefore very important that you remember exactly the corresponding location from point 4, to avoid errors that could occur later.

5.Now navigate to the sketchbook location in your file explorer (see 4.). Note that the destination folder is named 'Arduino' in the File Explorer at the sketchbook location. Double-click on the folder to see its contents. The folder contains another folder named "libraries".

▶ **What do I do if there is no 'libraries' folder?**

Now copy or drag the downloaded and unzipped folder into the `libraries` -folder.

1. Close the Arduino program completely and start it again to complete the installation of the respective libraries.

> Unfortunately, a typical error is that the senseBox library is not placed in the correct folder. Please check again if you put the file in the right folder from 4..

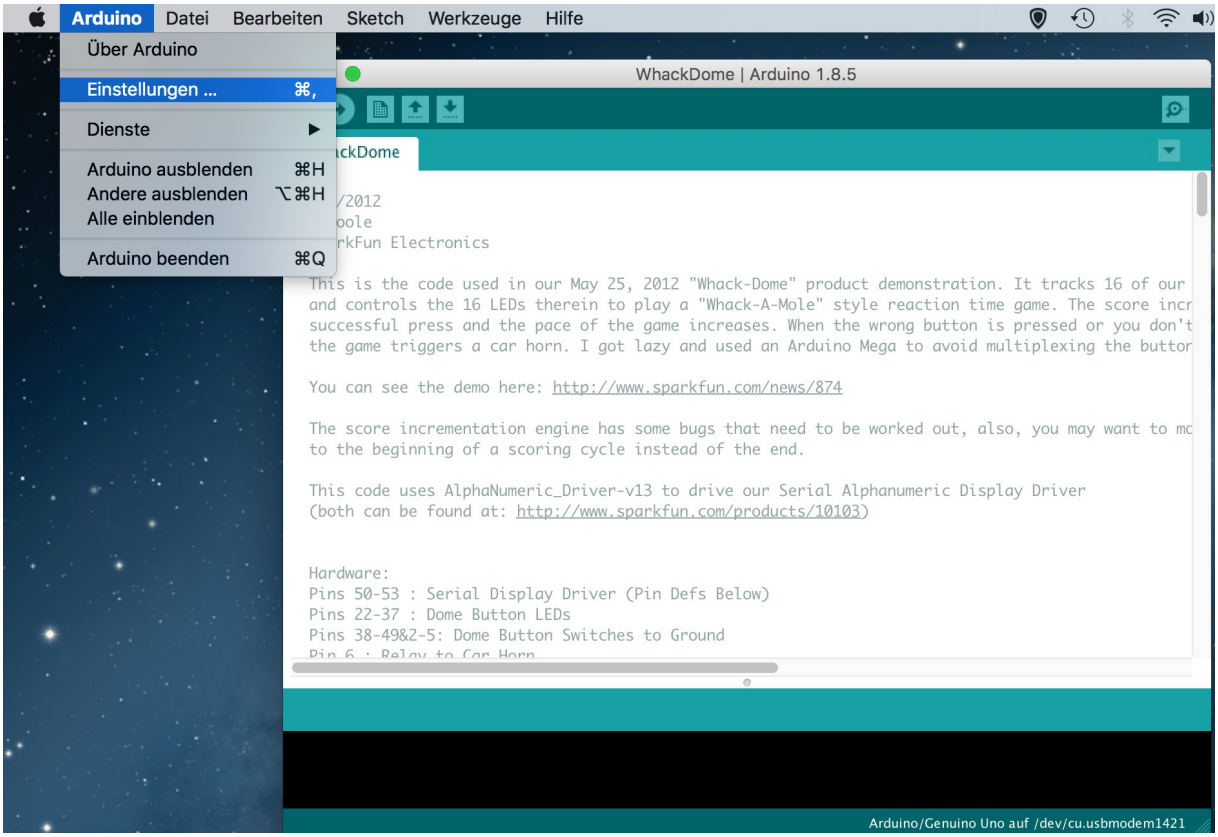
Insert Libraries Mac

1. Most external libraries can be found in Github repositories. To download them, you have to click the green button `Clone or download` and then in the window `Download ZIP` which opens.

   ![Example download of the "Adafruit_HDC_Library" from github.com] (pictures/libraries/github_download.png)

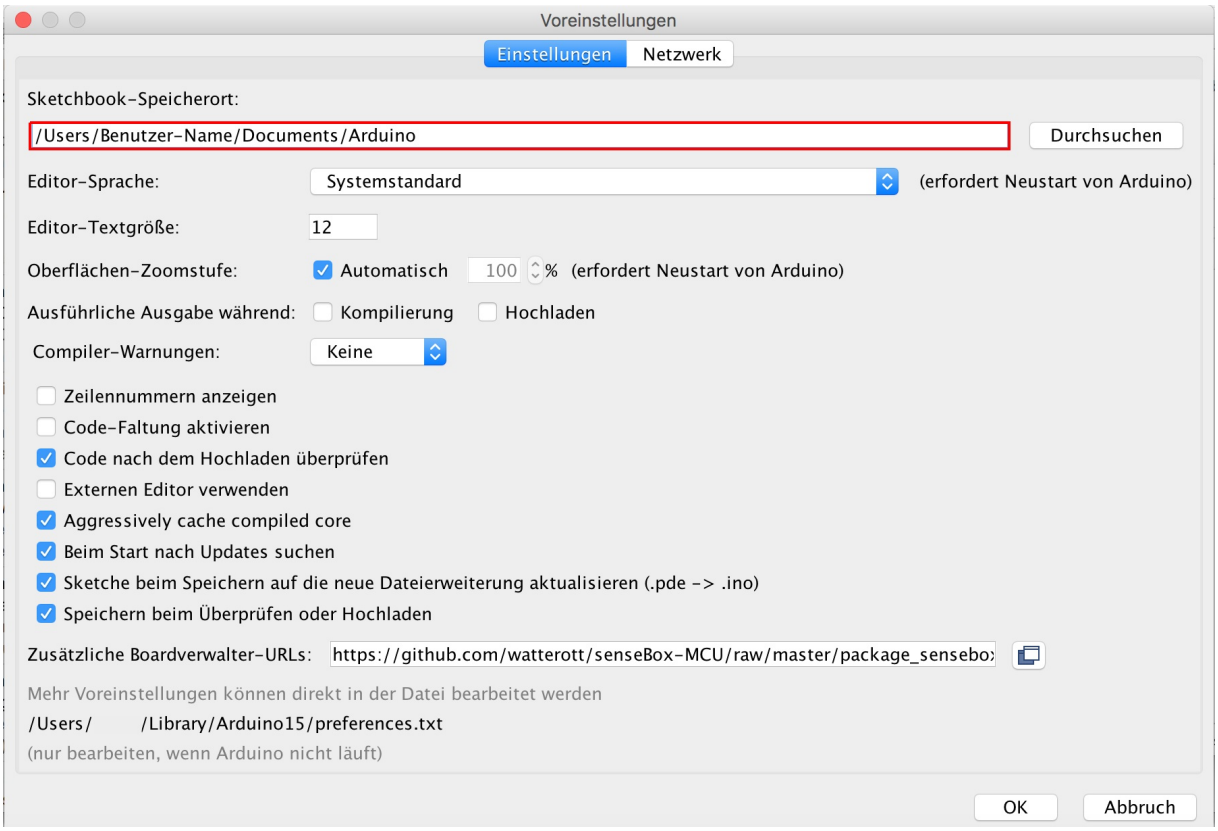
2. The download should start automatically and the file should be automatically unzipped and placed in your "Downloads" folder. Open the Downloads folder and see if the downloaded folder exists there. If there is only one . Zip file instead of one folder, double-click to unpack it.

3. Now open the Arduino IDE. Go to the top of `Arduino` -> `Settings ...` :



Click `Arduino` and then` Settings ... `

and look in the box under ` Sketchbook location 'for the location of the Sketchbook folder.



Look in the red-marked box to see where your sketchbook location is

Remember the path to this folder, ie the location where it is stored.

> You need to move the libraries you have already downloaded to the sketchbook location in the next step. It is therefore very important that you remember exactly the corresponding location from point 3, to avoid errors that could occur later.

4. Now you navigate to the sketchbook location in your Finder. Note that the destination folder in the Finder is named 'Arduino' at the sketchbook location. Double-click on the folder to see its contents. The folder contains another folder named "libraries".

   ▶ **What do I do if there is no 'libraries' folder?**

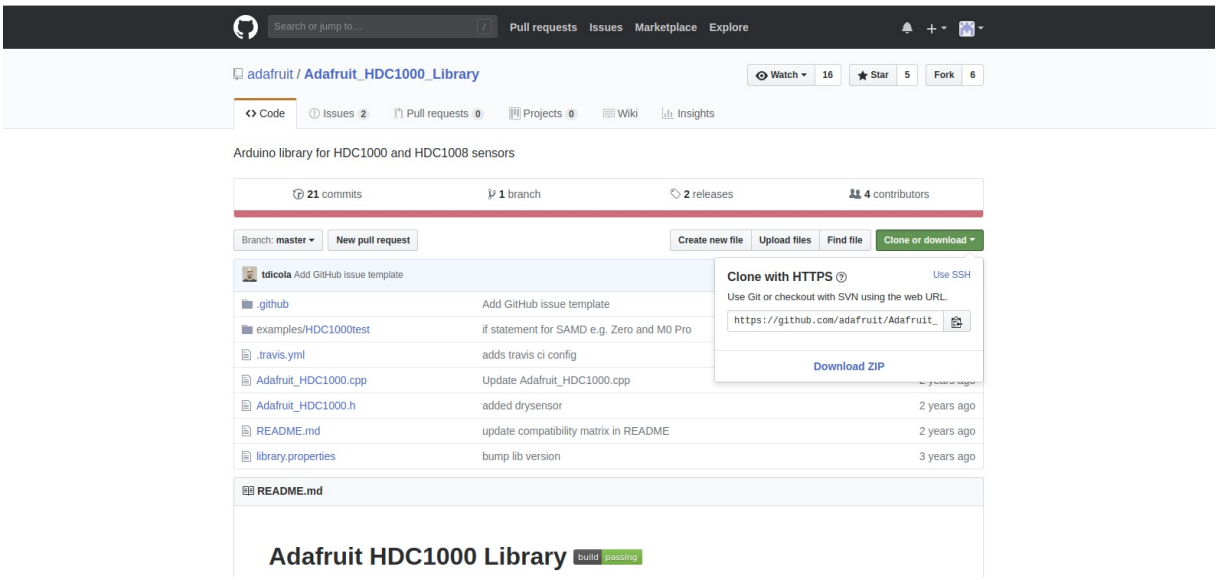
   Copy or drag now the downloaded (unzipped) folder into the `libraries` -folder.

5. Close the Arduino program completely and start it again to complete the installation of the respective libraries.

> Unfortunately, a typical error is that the senseBox library is not placed in the correct folder. Please check again if you have placed the file in the correct folder from 3.

Insert Libraries Linux

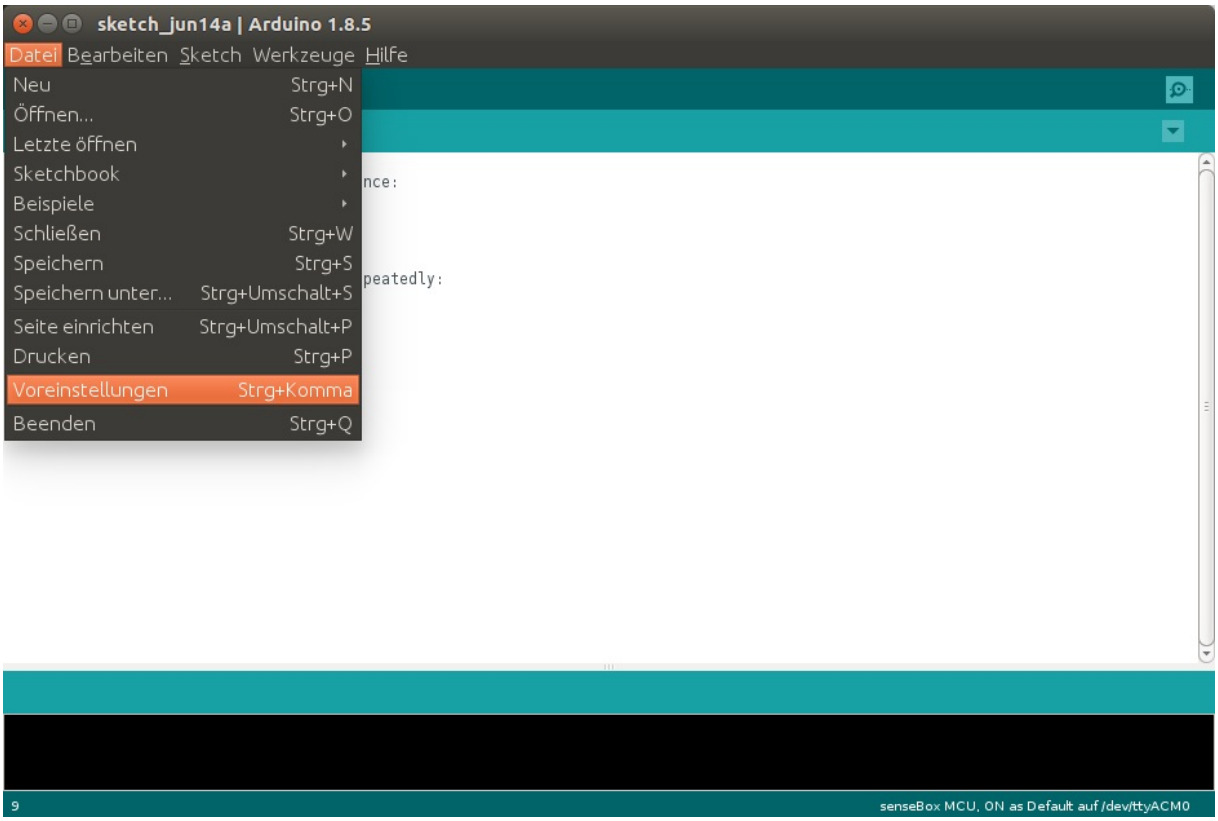
1. Most external libraries can be found in Github repositories. To download them, you have to click the green button `Clone or download` and then in the window `Download ZIP` which opens.



*Example download of the *

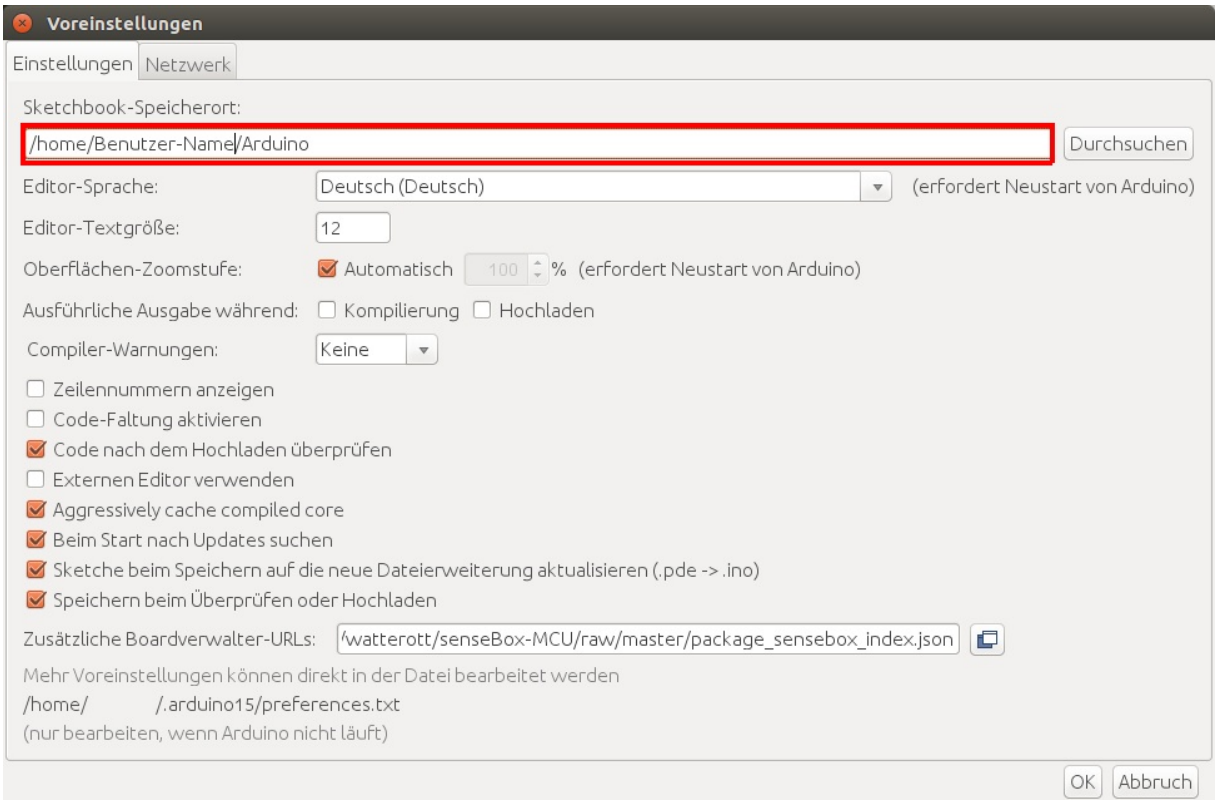
2.The download starts automatically and saves a `.zip` archive in your Downloads folder. Open the Downloads folder and unpack the `.zip` file with right-click -> `Extract Here` (*`Extract Here`*).

1. Now open the Arduino IDE. Go to `File` -> `Preferences` :



Click `File` and then` Preferences`

and look in the box under `Sketchbook location 'for the location of the Sketchbook folder.



Look in the red-marked box to see where your sketchbook location is

Remember the path to this folder, ie the location where it is stored.

> You need to move the libraries you have already downloaded and unzipped to the sketchbook location in the next step. It is therefore very important that you remember exactly the corresponding location from point 3, to avoid errors that could occur later.

2. Now navigate to the sketchbook location in your file explorer (see 3.). Note that the destination folder is named 'Arduino' in the File Explorer at the sketchbook location. Double-click on the folder to see its contents. The folder contains another folder named "libraries".

   ▶ **What do I do if there is no 'libraries' folder?**

   Copy or drag now the downloaded (unzipped) folder into the `libraries` -folder.

3. Close the Arduino program completely and start it again to complete the installation of the respective libraries.

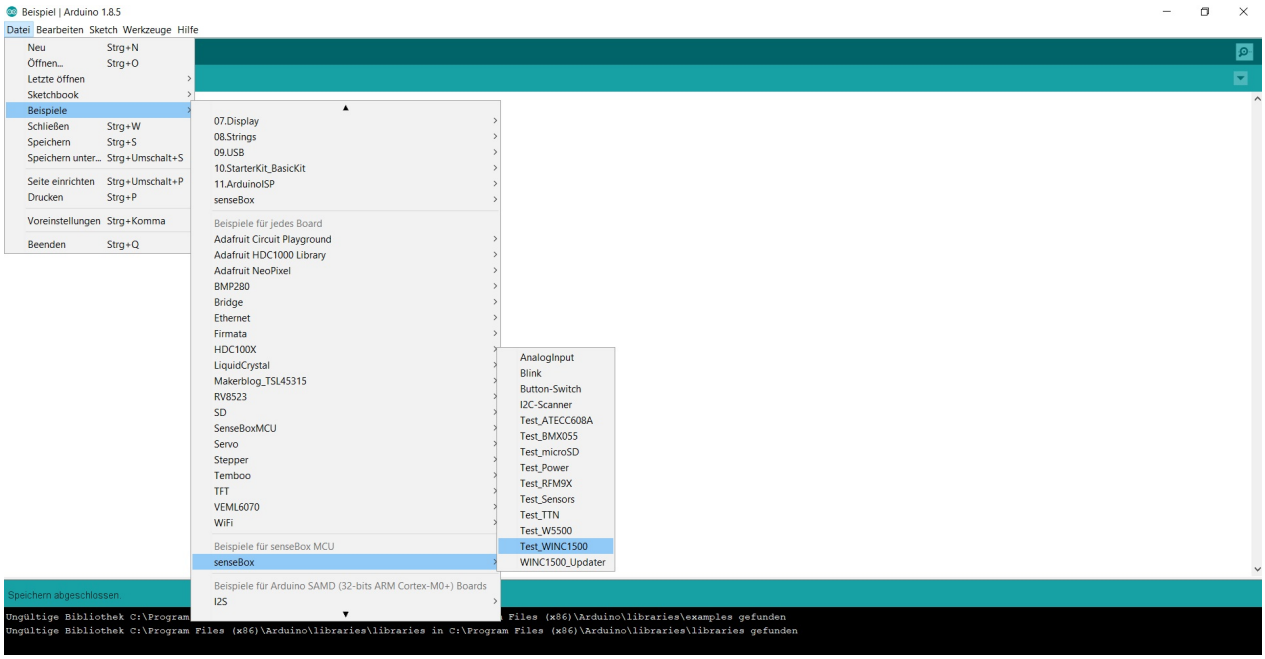
> Unfortunately, a typical error is that the senseBox library is not placed in the correct folder. Please check again if you have placed the file in the correct folder from 3..

Firmware Update Wifi-Bee

Unfortunately, some of our WINC1500 WiFi Bees have outdated firmware (version 19.4.4) installed. Unfortunately, there is no other way to update this firmware than to do it manually. The following chapter explains how to find out which firmware you are using and (if you have an outdated version) how to update it.

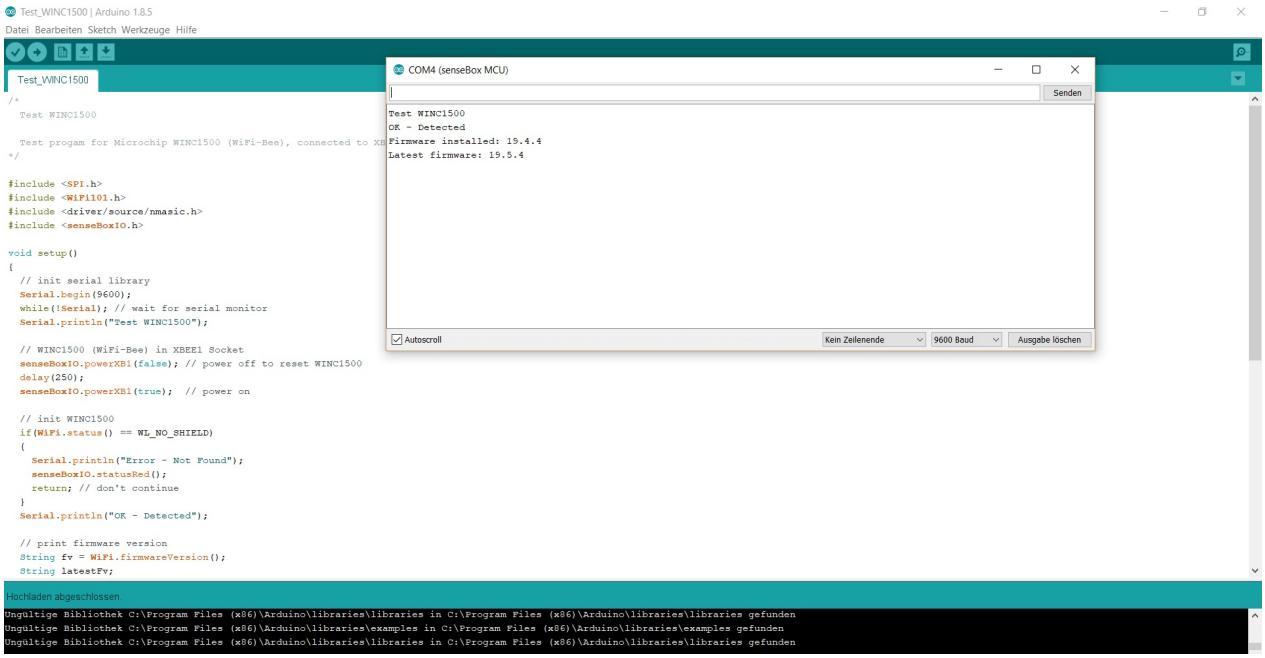
Test the Version

First, it must be found out which version the supplied WiFi Bee has. Go to"file" " Examples " and under "Examples for senseBox MCU" on"Test_WINC1500"



Open the WiFi-Test

Now upload the sketch to your board (by clicking the arrow icon). Note that the WiFi Bee must be plugged into your senseBox board (please plug it in to XBEE1). Then click on the serial monitor (by clicking the magnifying glass icon) and it will be checked if your Wifi-Bee is functional and shows which firmware is installed on it.



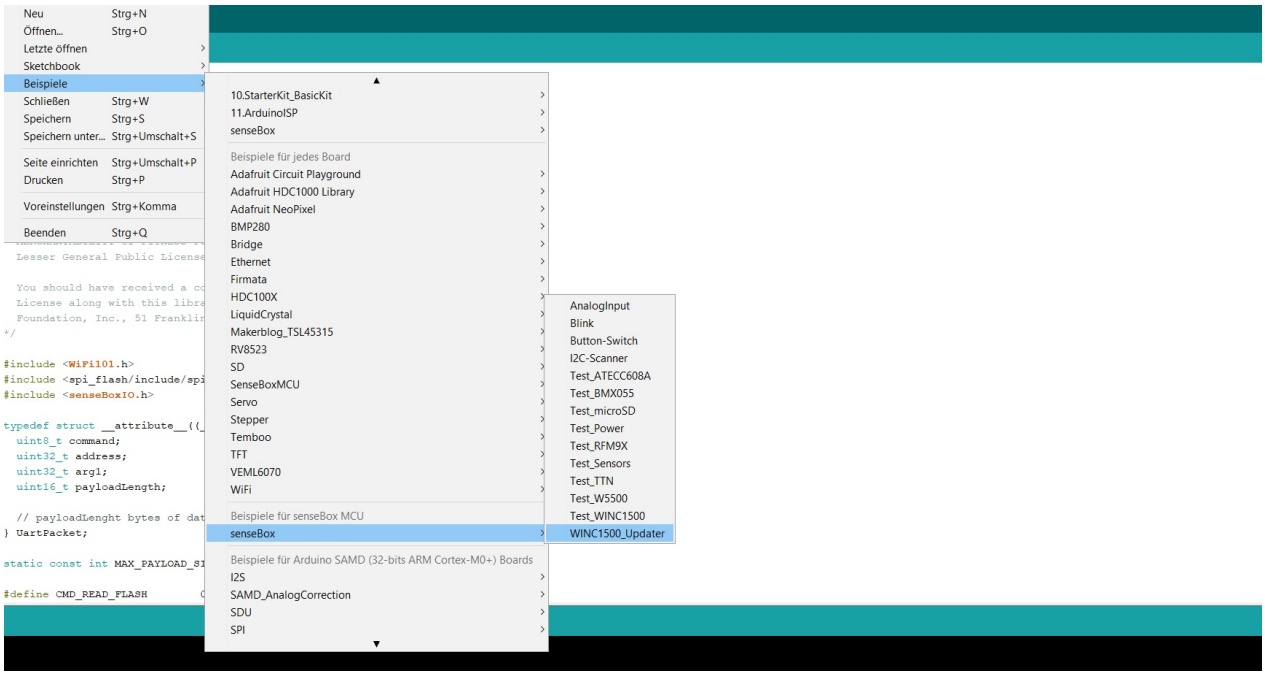
Test results with a non-current firmware

If you have a firmware ** 19.5.2, or higher ** you can cancel here. Your WiFi Bee works perfectly.

If you have a firmware lower than 19.5.2 you unfortunately have to update the firmware. You will find out how this works in the next step.

WiFi-Bee Firmware Update

To update the firmware, follow the path from the top:"file"; " Examples " and under "Examples for senseBox MCU" on"WINC1500_Updater"

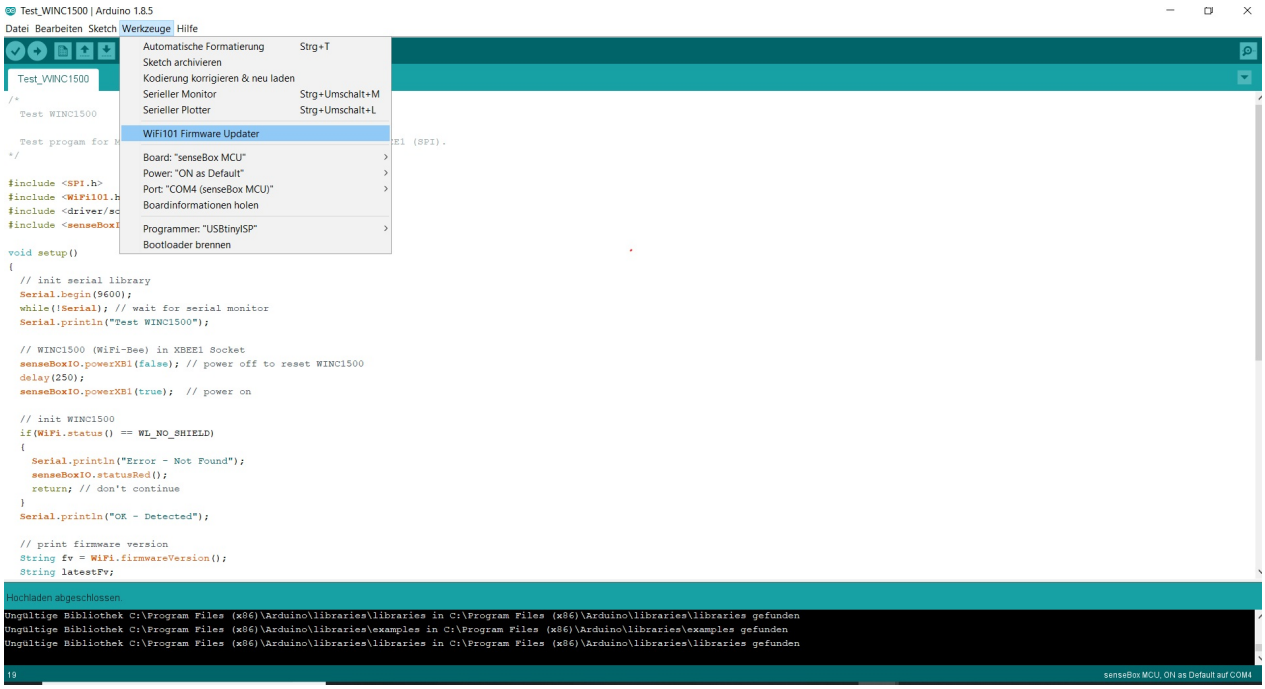


Open the WINC1500_Updater

Now load the sketch on your board (by clicking the arrow icon), on which the WiFi Bee is plugged (please plug it on XBEE1)

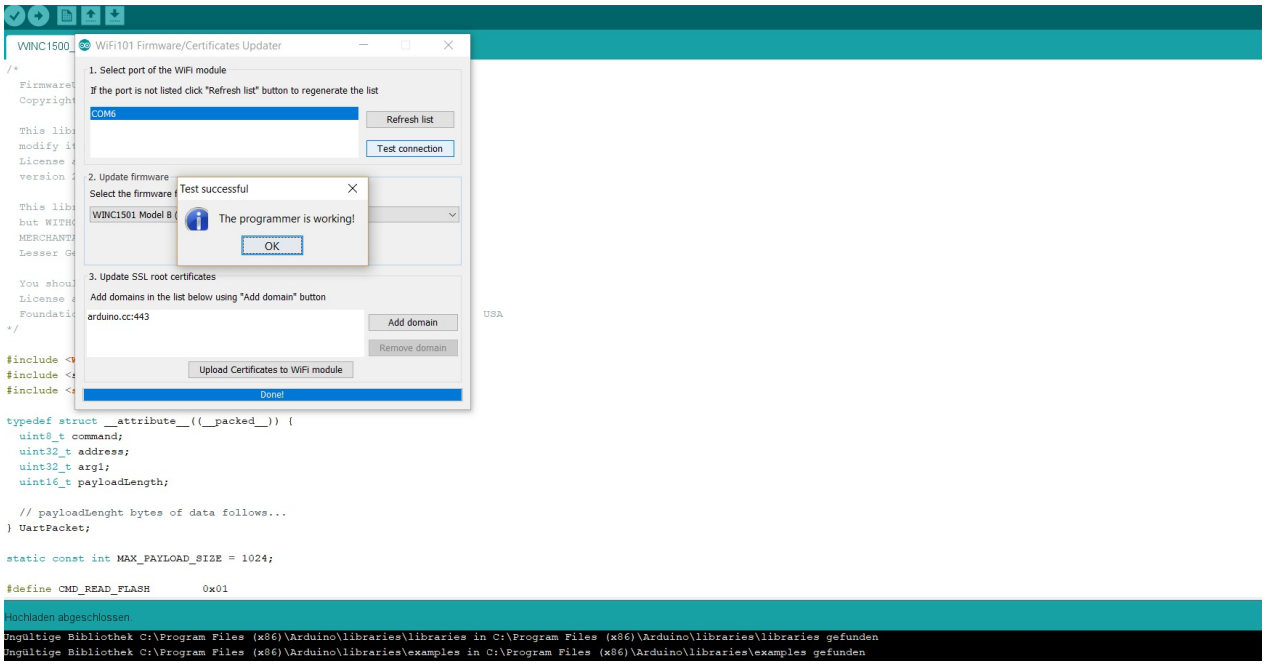
> Do not open the serial monitor this time (do not click on the magnifying glass icon)

Now click `Tools` and select `Wifi101 Firmware Updater` .



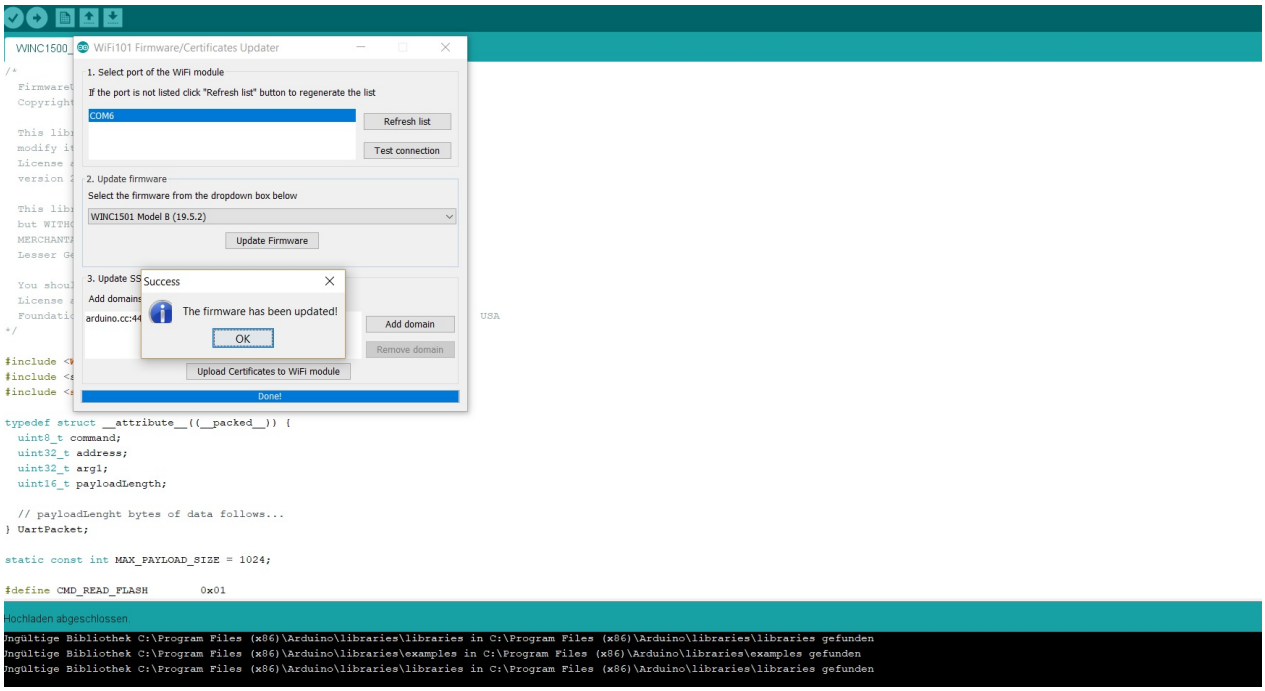
Choose Wifi 101 Firmware Updater

First, you can test the connection by clicking on the `COM Port` and then on `Test connection`. The following information should come back: "The programmer is working!"



Test connection of the Wifi-Bee

Almost done, now just click on `Update Firmware` and the upload starts. Then a success message should come "The firmware has been updated!".



Finally Update Firmware

> Do not let yourself be confused that there is also a version of the firmware 19.5.4, but this is not found in Arduino if you want to update the firmware. This firmware will be integrated with the Arduino Release 1.8.6 and will be available from then on. Nicht davon irritieren lassen, dass es auch eine Version der Firmware 19.5.4 gibt, diese aber nicht in Arduino zu finden ist wenn man die Firmware updaten möchte.

We apologize for the detour and wish you lots of fun with the senseBox.

If this article has not helped you, you can try on www.forum.sensebox.de to look for a solution, or if necessary submit yourself a contribution.

Update Windows USB Bootloader Driver

In some rare cases Windows can cause problems with the USB-Bootloader drivers. What you can do to check if your drivers are installed correctly and how you can install them if you find out here.

Check Driver Status

To check if your USB bootloader drivers work, follow these steps:

- Connect your senseBox MCU board via USB cable to your Windows computer
- Open the Device Manager by searching for `Device Manager` in the Windows bar and open it with a click.
- Activate the bootloader mode of the senseBox MCU by pressing the `Reset` button (red mechanical button on the senseBox MCU) twice in quick succession
- The device manager should now show a point `Connections (COM & LPT)`
- Click on the item to open a list of connected devices, where the senseBox MCU should be listed
- If the senseBox MCU is not listed, disconnect the USB cable from the computer and reconnect it - leave the device manager open and look what happens

If there is no corresponding and no new device displayed on reconnect, your USB bootloader drivers are not installed correctly. Download the current drivers here with one click:

Download senseBox MCU Driver[1]

Now go to Device Manager and select `Update Driver` -> `Find on Computer` and select the drivers you just downloaded.

Restart the computer and check the driver status again, as stated above. The senseBox MCU should now be recognized.

> If this article did not help you, you can try to search for a solution in our forum or create a post there yourself.

[1]. https://github.com/watterott/senseBox-MCU/raw/master/arduino/driver.zip ↩

Update of Board Support Package and Libraries

We have changed the installation steps, to make the installation and updates of the senseBox libraries more user-friendly. On this page, we'll show you the steps you need to take to update your board support package and your senseBox libraries. This guide only applies to you if you have completed the first steps of this book before June 23, 2018 .

What is new?

We have developed a new board support package that combines the old board support package with the senseBox libraries. This bypasses the error-prone manual installation of the senseBox libraries. At the same time, Ardunio IDE's built-in update support for board support packages can be used to bring the libraries up to date. So updates can be recorded in the future with much less effort.

Instructions for updating

The update consists of 2 steps:

1. Deleting the senseBox libraries from the Sketchbook folder to avoid duplication of libraries and the use of old versions.
2. The installation of the new board support package to integrate the libraries in Arduino. Wähle dein Betriebssystem, um die passende Anleitung zu sehen:

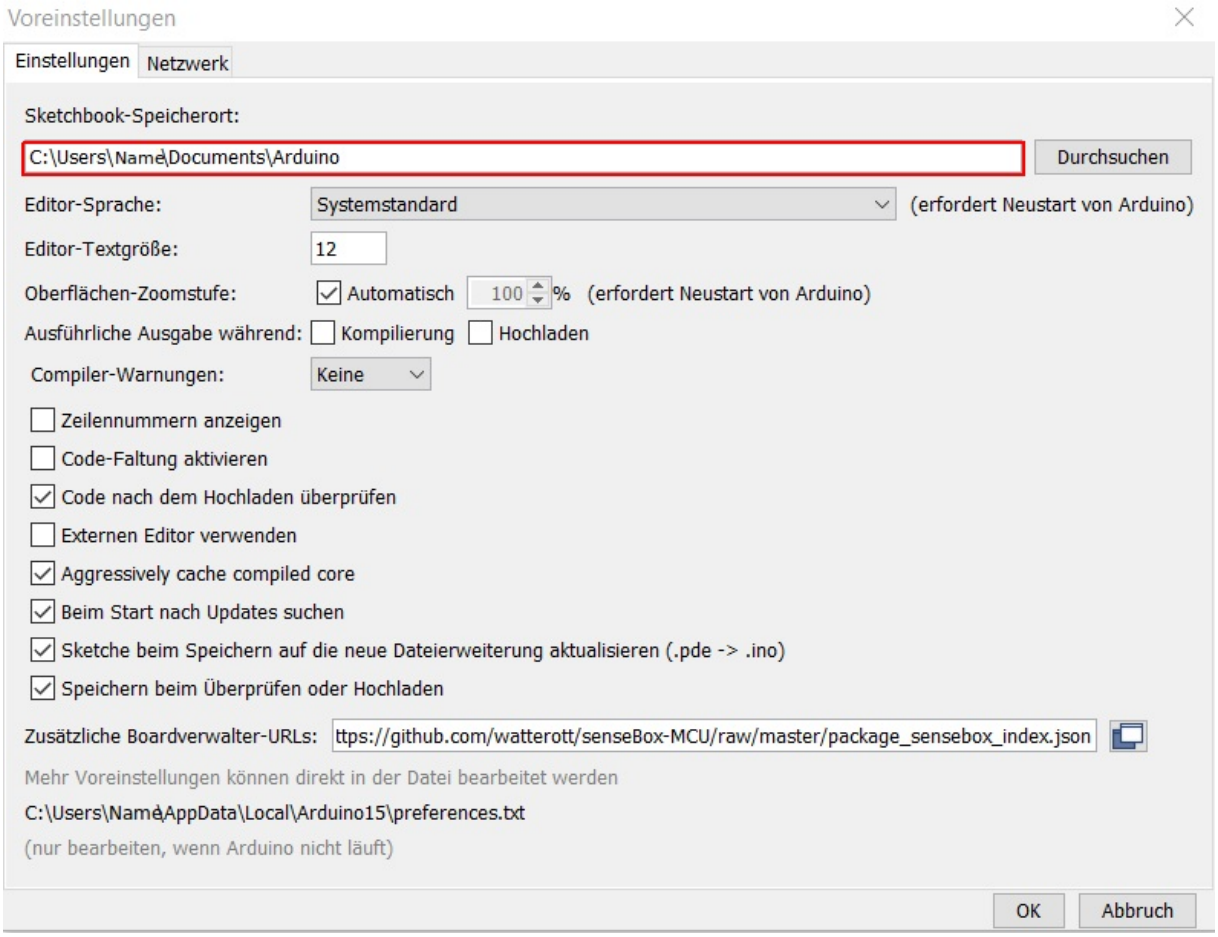
- Windows
- Mac(OSX)
- Linux

Step 1: Delete the senseBox libraries from the sketchbook folder

1. Now open the Arduino IDE. Go to `File -> Preferences` :

Click `File` and then` Preferences`

and look in the box under `Sketchbook location 'to see where the Sketchbook folder is stored.



Look in the red-marked box to see where your sketchbook location is

Remember the path to this folder, ie the location where it is stored.

1. Now navigate to the sketchbook location in your file explorer (see 1.). Note that the destination folder is named 'Arduino' in the File Explorer at the sketchbook location. Open this folder. Inside the `Arduino` folder is a folder called `libraries`. Within this folder are the senseBox libraries. Delete the `libraries` folder to remove them.

> If you are an experienced Arduino user and have in the past included other external libraries that are not part of the senseBox libraries. Go to the `libraries' folder and delete all the libraries that were not external to you, rather than deleting the entire folder.

2. Now completely close the program Arduino and start it again to complete the deletion of the old senseBox libraries.

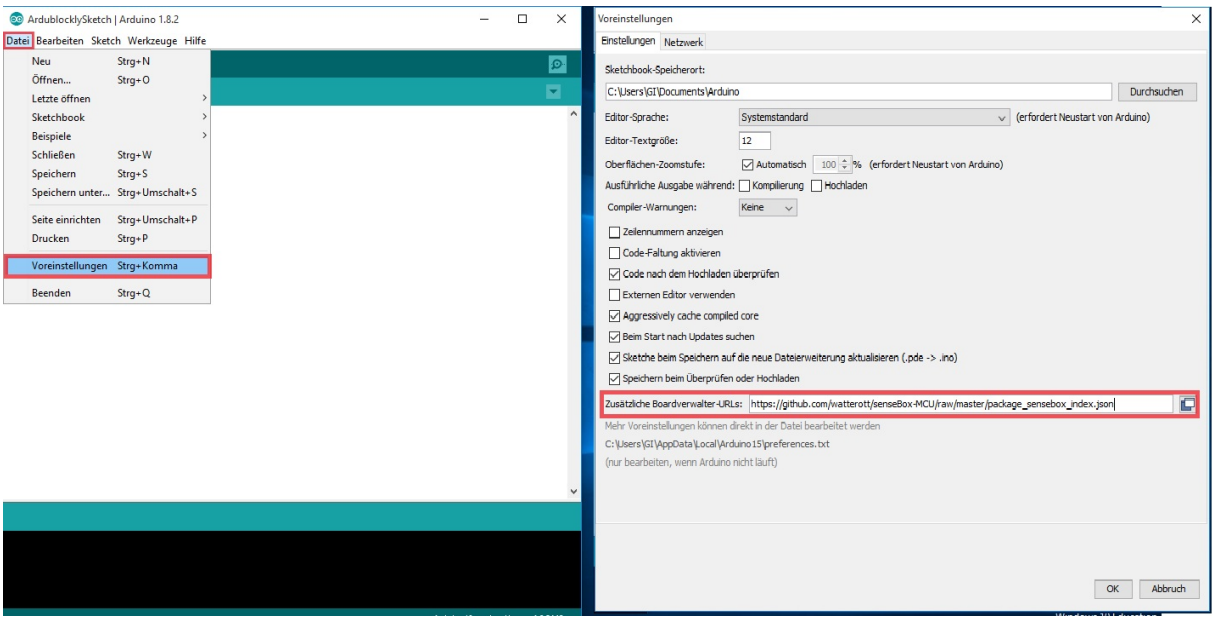
Step 2: Integrate new board support package

To incorporate the new board support package, please proceed similar, as in the first steps. There will just be a few small changes.

1. Paste the following URL in your Arduino IDE under File -> Preferences into the field Additional Board Administrator URLs :

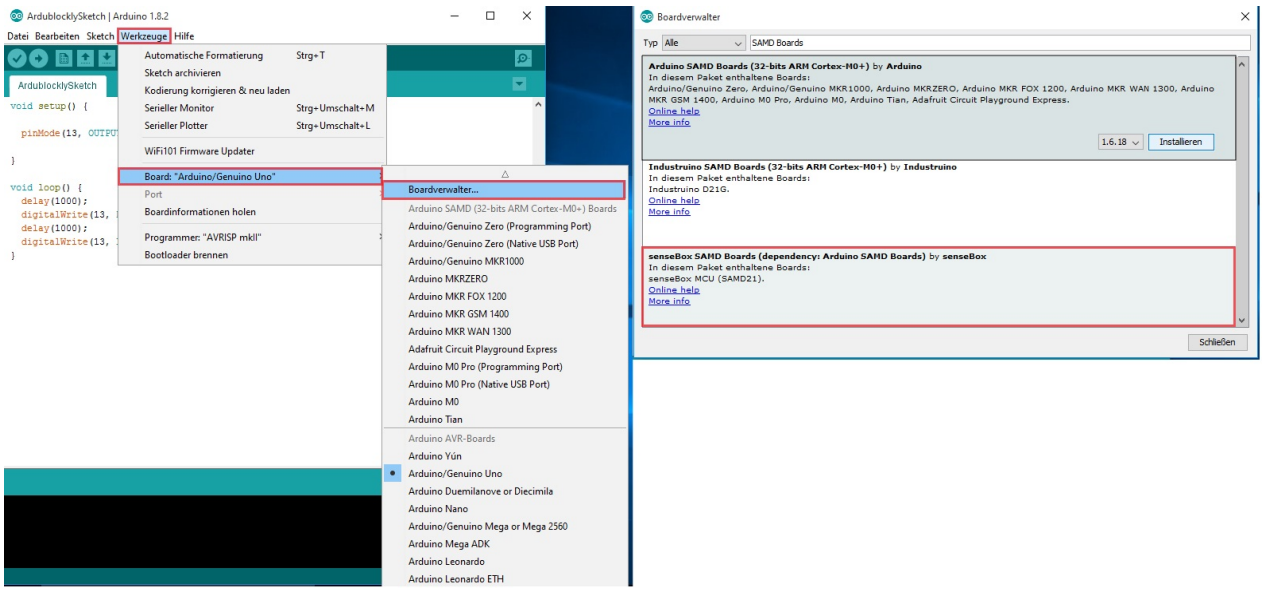
```
https://github.com/sensebox/senseBoxMCU-core/raw/master/package_sensebox_index.json
```

> An der Stelle steht im Normalfall vorher schon folgende URL: https://github.com/watterott/senseBox-MCU/raw/master/package_sensebox_index.json diese sieht der obigen sehr ähnlich, ist aber nicht die gleiche URL. Sie muss aber unbedingt durch die oben stehende URL ausgetauscht werden.



Open the preferences and paste the URL

2.Now open the board administrator under Tools -> Board: "..." -> Board Administrator and search there for the senseBox SAMD Boards - Package.

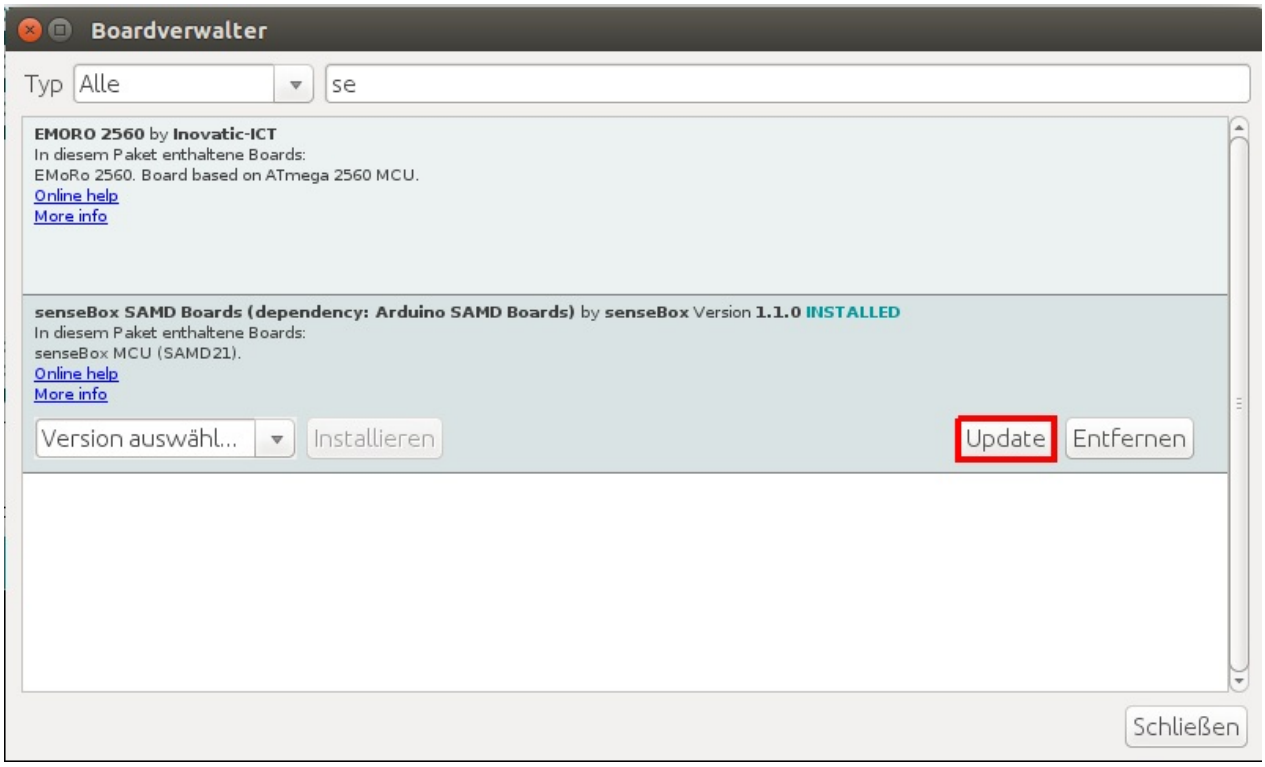


Search for the red highlighted package

If you click on the entry in the list, there appears an update button.

> It is important to click on the entry first. Otherwise, the update button is not displayed, even if there is already a new version.

1. Click on this button and make sure that the installed version is higher than 1.1.0.

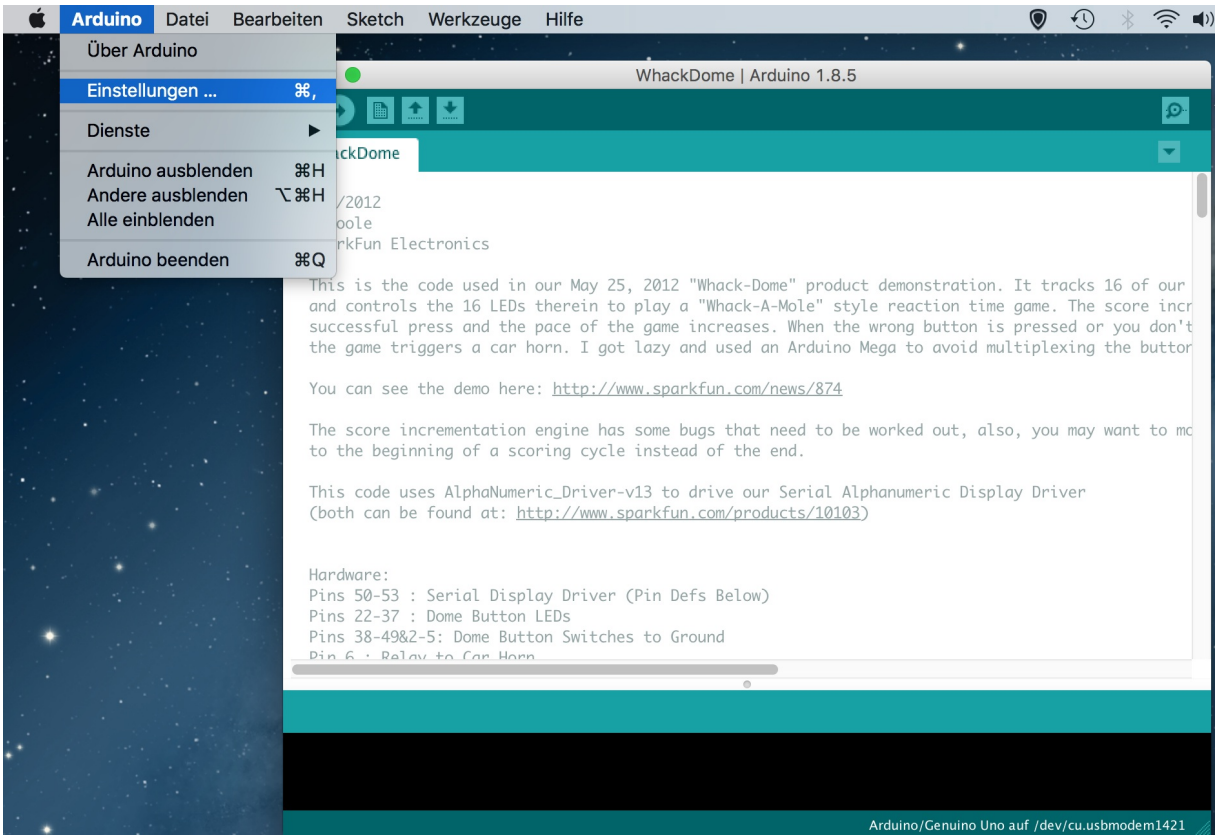


Click 'Update' to update the board support package

Since we regularly update the senseBox SAMD boards package for you, you should always go back to the board administrator and see if the senseBox SAMD boards package is still up to date. Open the board administrator as described above, look for senseBox SAMD boards and click there on 'update'.

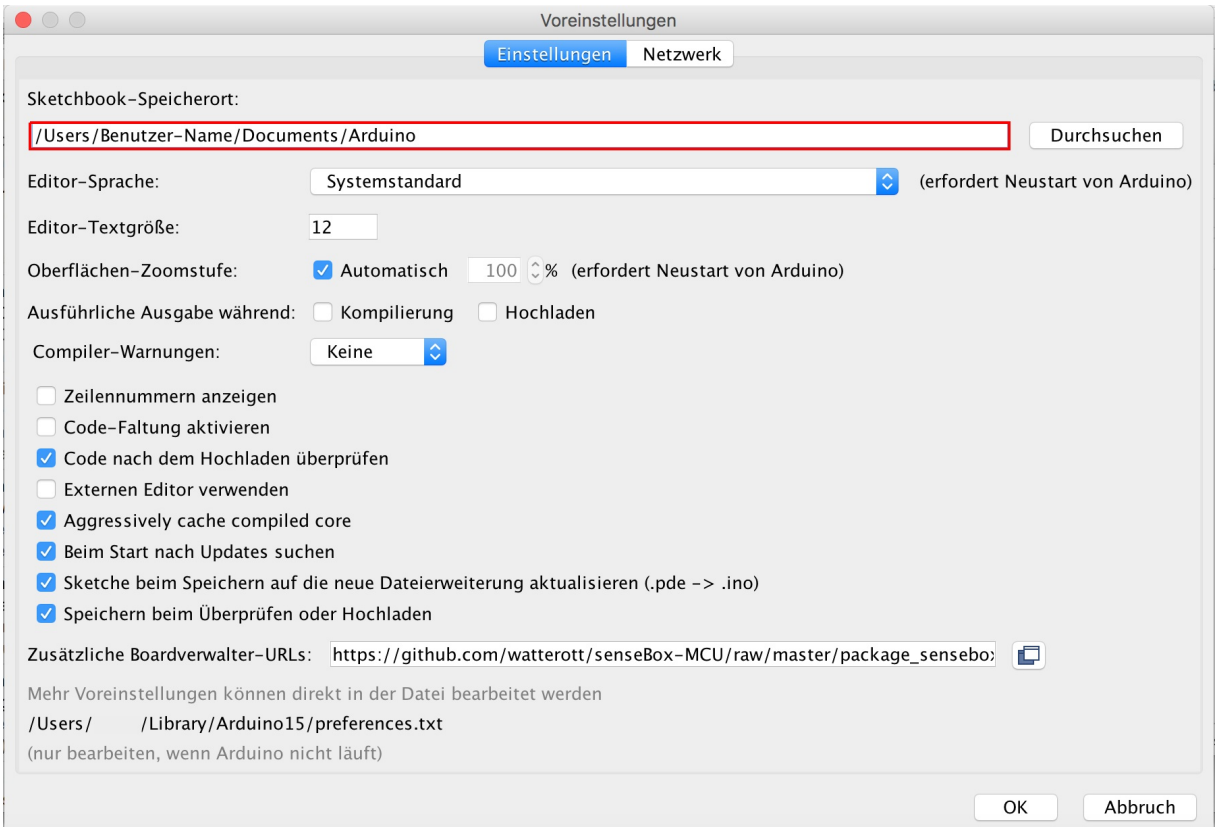
Delete the senseBox Libraries from the Sketchbook Folder

1. Now open the Arduino IDE. Go to `Arduino` -> `Settings ...` :



Click `Arduino` -> ` Settings ... `

and look in the box under ` Sketchbook location 'to see where the Sketchbook folder is stored.



Look in the red-marked box where your sketchbook location is

Remember the path to this folder, ie the location where it is stored.

1. Now navigate to the sketchbook location in your file explorer (see 1.). Note that the destination folder is named 'Arduino' in the File Explorer at the sketchbook location. Open this folder. Inside the `Arduino` folder is a folder called `libraries`. Within this folder are the senseBox libraries. Delete the `libraries` folder to remove them.

> aria-hidden = "true" style = "color: # f0ad4e"> If you are an experienced Arduino user and have in the past included other external libraries that are not part of the senseBox libraries. Go to the `libraries` folder and delete all the libraries that were not external to you, rather than the delete entire folder.

2. Now completely close the Arduino program and restart it to finish deleting the old senseBox libraries.

Step 2: Integrate new board support package

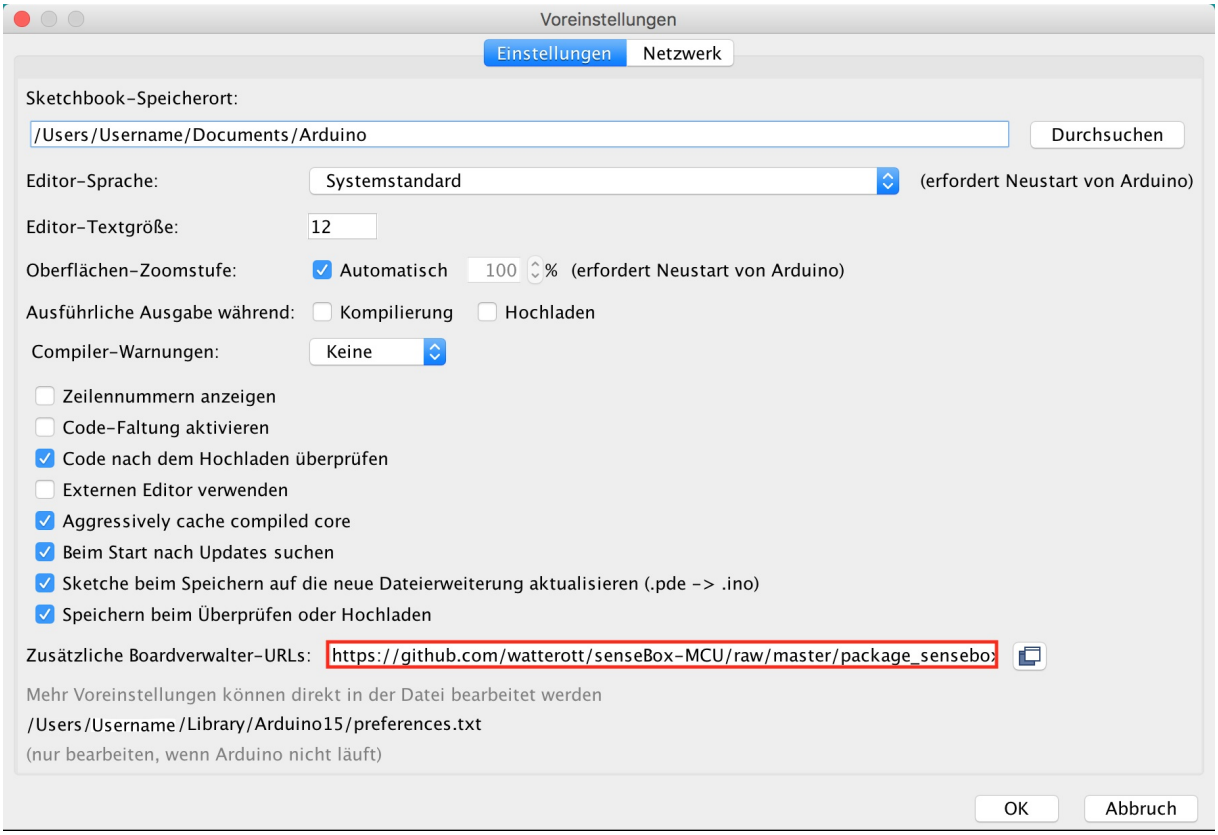
To incorporate the new board support package, it's similar, as in the first steps, with a few small changes.

1. Paste the following URL in your Arduino IDE under 'Arduino `->` Settings ... ` in the field Additional Board Administrator URLs :

```
https://github.com/sensebox/senseBoxMCU-core/raw/master/package_sensebox_index.json
```

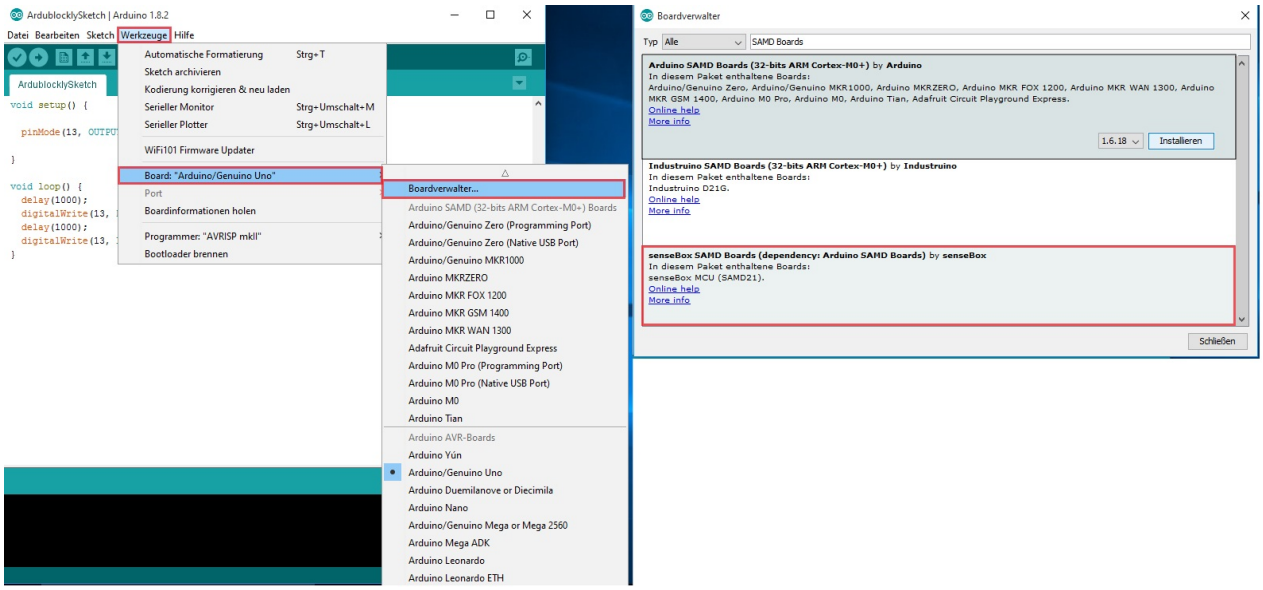
> An der Stelle steht im Normalfall vorher schon folgende URL: https://github.com/watterott/senseBox-

> MCU/raw/master/package_sensebox_index.json diese sieht der obigen sehr ähnlich, ist aber nicht die gleiche URL. Sie muss aber unbedingt durch die oben stehende URL ausgetauscht werden.



Open the preferences and paste the URL there

2. Now open the board administrator under Tools -> Board: "..." -> Board Administrator and search there for the senseBox SAMD Boards - Package.

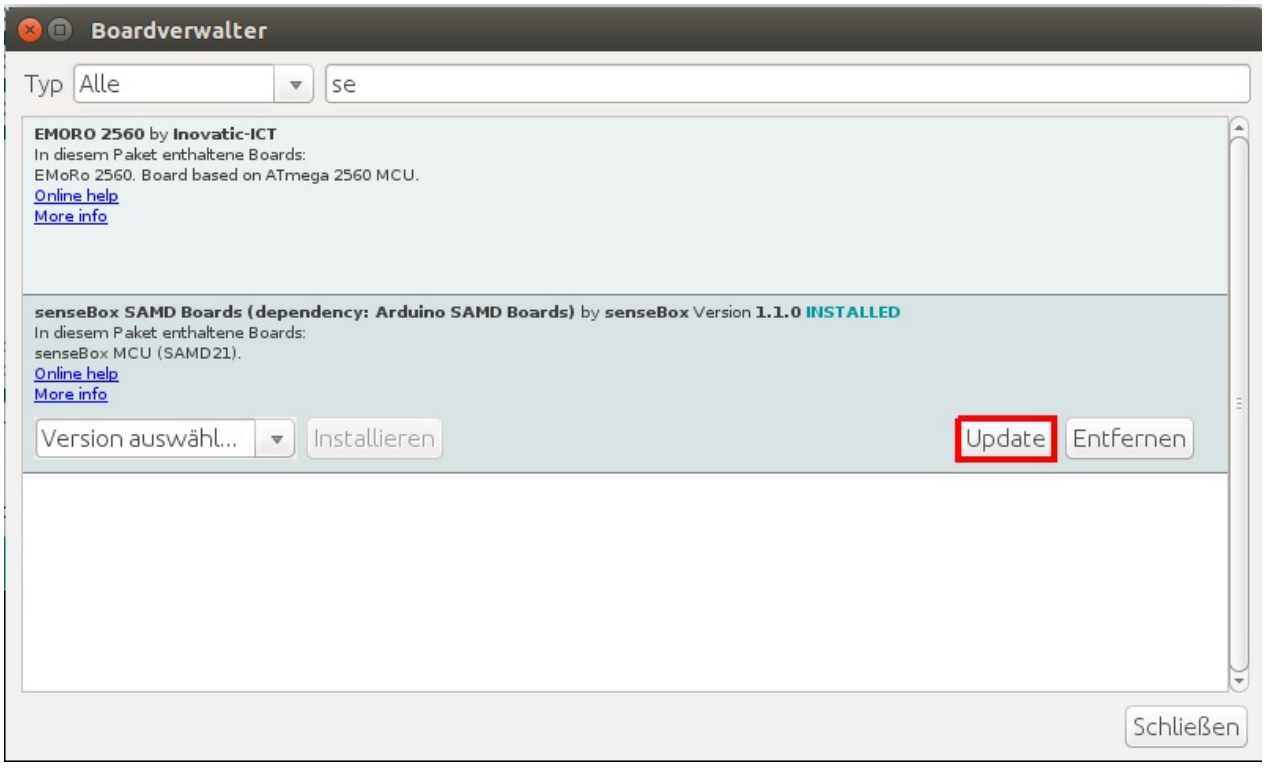


Search for the red highlighted package

If you click on the entry in the list, there appears an update button.

> It is important to click on the entry first. Otherwise, the update button is not displayed, even if there is already a new version.

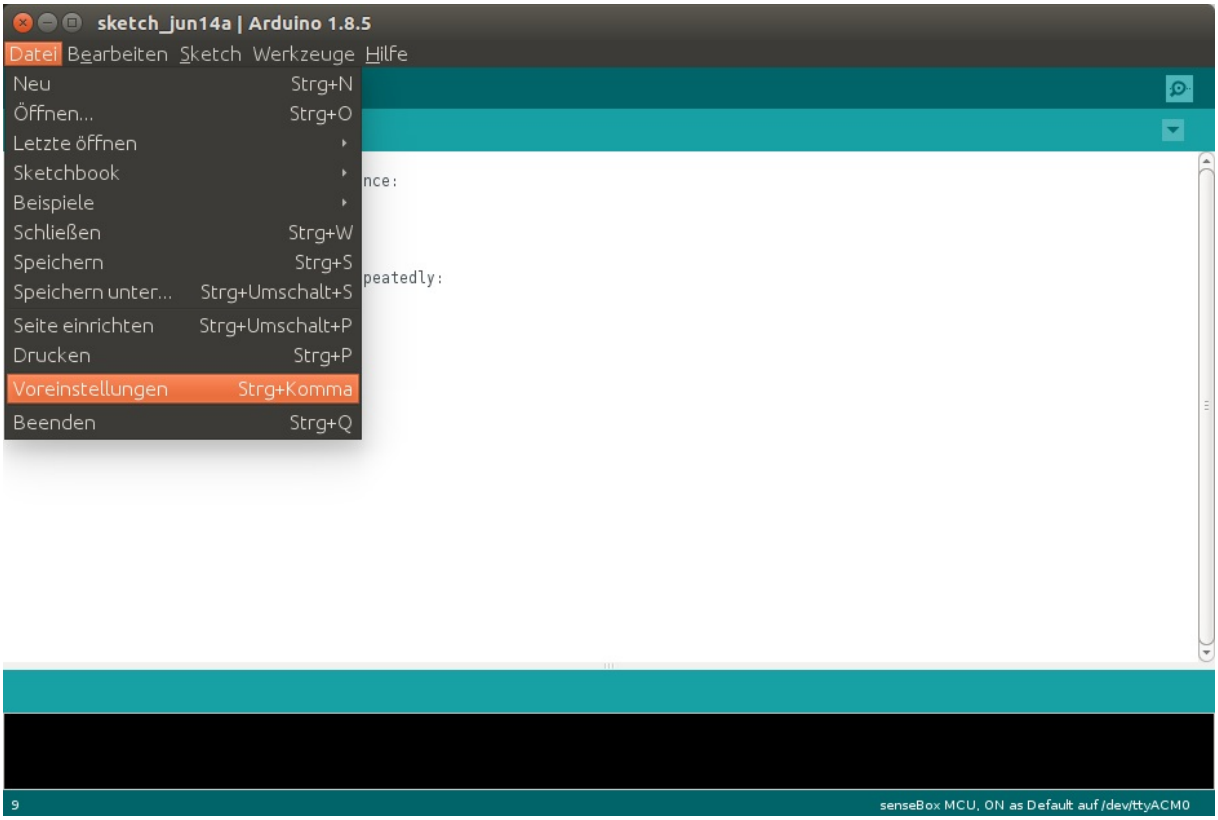
3.Click on this button and make sure that the installed version is higher than 1.1.0.



Since we regularly update the senseBox SAMD boards package for you, you should always go back to the board administrator and see if the senseBox SAMD boards package is still up to date. Open the board administrator as described above, look for senseBox SAMD boards and click there on 'update'.

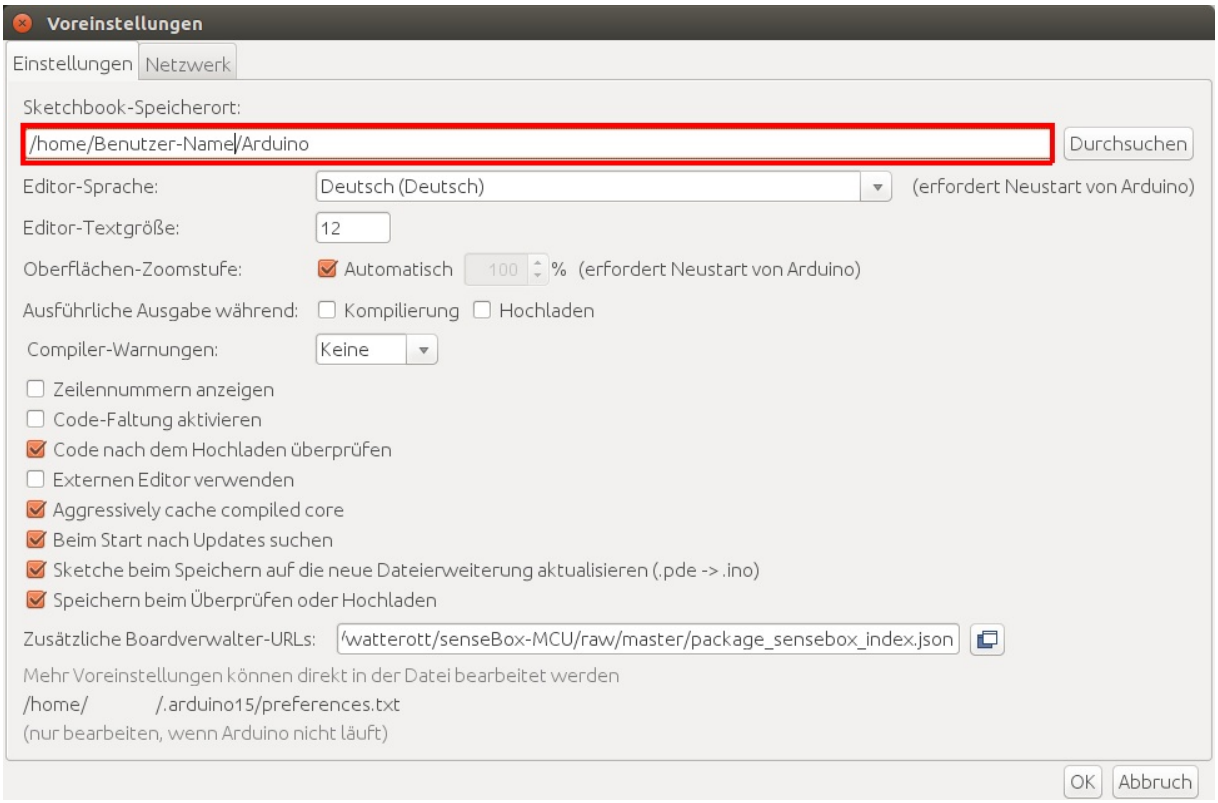
Step 1: Delete the senseBox libraries from the sketchbook folder

1. Now open the Arduino IDE. Go to `File` -> `Preferences` :



Click `File` -> ` Preferences`

and look in the box under ` Sketchbook location 'to see where the Sketchbook folder is stored.



Look in the red-marked box to see where your sketchbook location is

Remember the path to this folder, ie the location where it is stored.

1. Now navigate to the sketchbook location in your file explorer (see 1.). Note that the destination folder is named 'Arduino' in the File Explorer at the sketchbook location. Open this folder. Inside the `Arduino` folder is a folder called `libraries`. Within this folder are the senseBox libraries. Delete the `libraries` folder to remove them.

> aria-hidden = "true" style = "color: # f0ad4e"> If you are an experienced Arduino user and have in the past included other external libraries that are not part of the senseBox libraries. Go to the `libraries` folder and delete all the libraries that were not external to you, rather than the delete entire folder

2. Now completely close the program Arduino and start it again to complete the deletion of the old senseBox libraries.

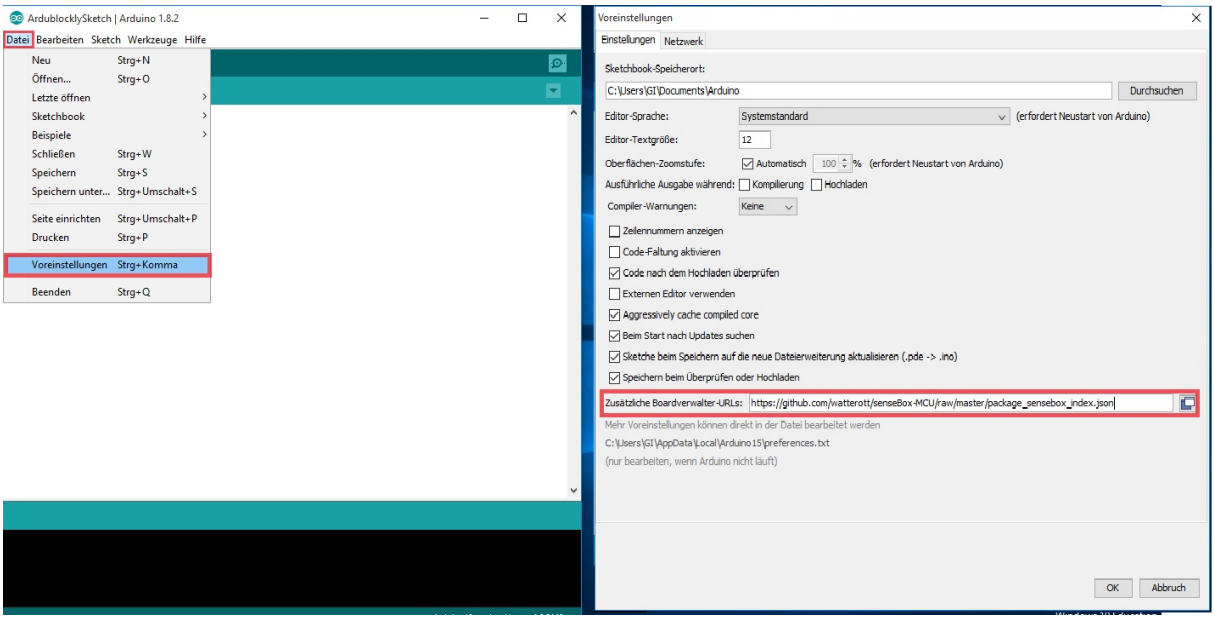
Step 2: Integrate new board support package

To incorporate the new board support package, it's similar, as in the first steps, with a few small changes.

1. Paste the following URL in your Arduino IDE under File -> Preferences into the field Additional Board Administrator URLs :

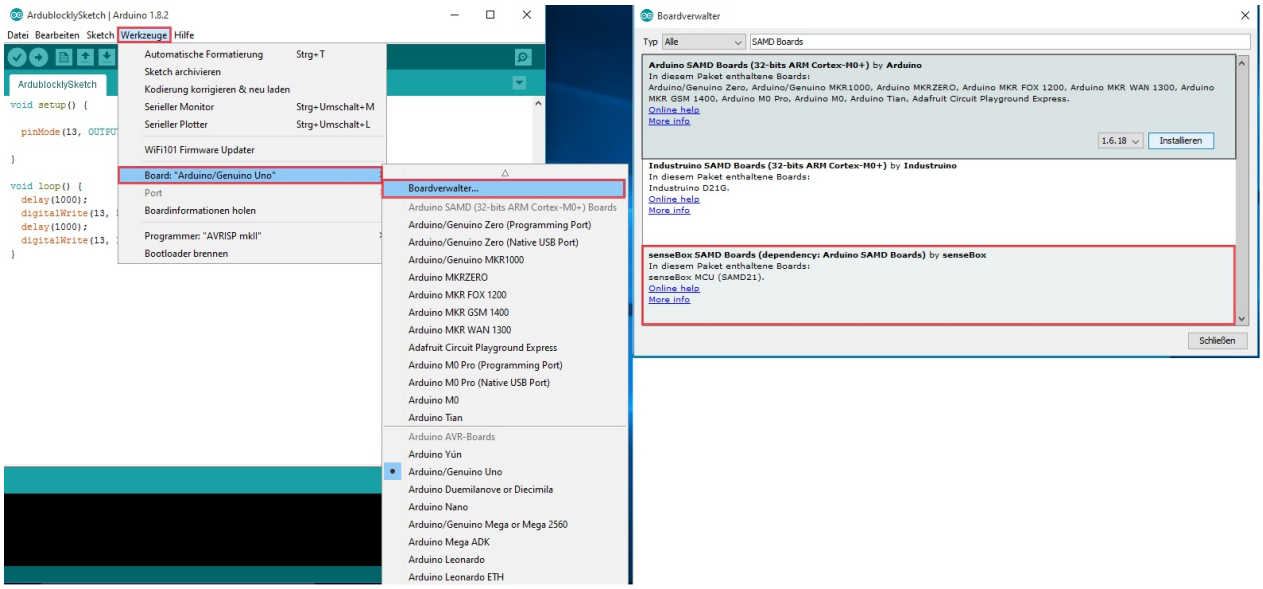
```
https://github.com/sensebox/senseBoxMCU-core/raw/master/package_sensebox_index.json
```

> Normally, the following URL already exists at this point: https://github.com/watterott/senseBox-MCU/raw/master/package_sensebox_index.json this looks very similar to the above, but is not the same URL. However, it must necessarily be replaced by the above URL.



Open the preferences and paste the URL

2.Now open the board administrator under Tools -> Board: "..." -> Board Administrator and search there for the senseBox SAMD Boards - Package.



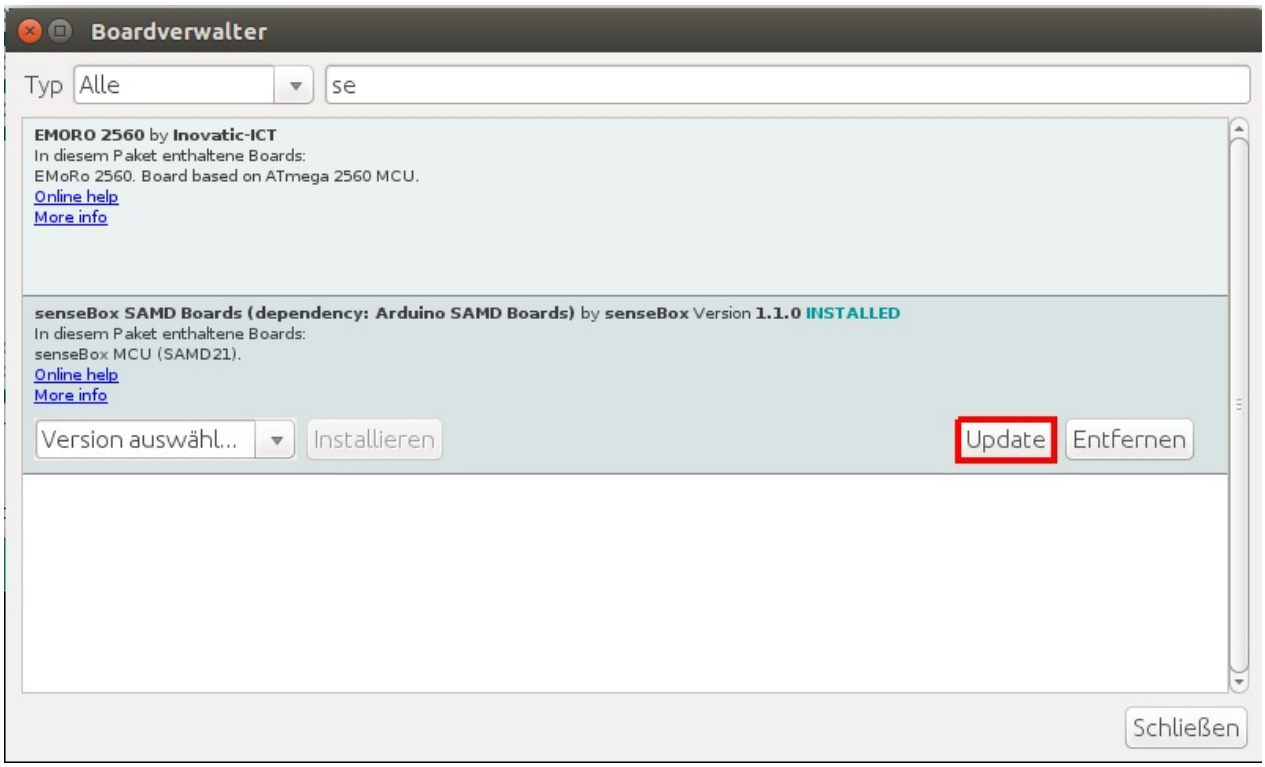
Search for the red highlighted package

If you click on the entry in the list, there appears an update button.

> It is important to click on the entry first. Otherwise, the update button is not displayed, even if there is already a new

version.

1. Click on this button and make sure that the installed version is higher than 1.1.0.



Click 'Update' to update the board support package

Since we regularly update the senseBox SAMD boards package for you, you should always go back to the board administrator and see if the senseBox SAMD boards package is still up to date. Open the board administrator as described above, look for senseBox SAMD boards and click there on 'update'.