

# Geosensors with senseBox – Lab 01



Source: <https://sensebox.de/>

## Overall Objectives

The geosensor lab is structured in three parts. In this lab, you should achieve the competence to build and run a technology that enables ambient sensing. Next lab, you make experiments with different sensors in gathering data from outdoors. The final lab, you will merge your data with further open context data and visualize them in the web by using a Javascript 3d library.

## 1 This lab: Setting up the senseBox and developing with Arduino

### 1.1 Goals

- Learn to setup and test a geosensor device (senseBox example)
- Learn to develop with a geosensor IDE (Arduino IDE)

## 1.2 Tasks

### 1.2.1 Setup Arduino IDE

🕒 15min 👤 Individual Work

Follow the instructions at <https://sensebox.github.io/books-v2/edu/en/erste-schritte/software-installation.html> and <https://sensebox.github.io/books-v2/edu/en/erste-schritte/board-support-packages-installieren.html> to install the Arduino IDE and to learn about the necessary Arduino libraries to work with the sensebox:edu.

### 1.2.2 Getting to know of Arduino IDE and programming

🕒 30min 👤 Individual Work

Visit <https://sensebox.github.io/books-v2/edu/en/grundlagen/der-arduino-sketch.html> and learn on how you write code in Arduino IDE. If necessary, search for additional tutorials.

### 1.2.3 Setup, installation and testing of the sensebox:edu components

🕒 75min 🧑‍🤝‍🧑 Group Work

The following instructions explain on how to setup, program and test the different sensebox:edu components

- <https://sensebox.github.io/books-v2/edu/en/erste-schritte/anschluss-und-verkabelung.html>
    - Wifi and mSD-Bee to connect to the WiFi and to access the SD card
    - **Note:** ignore the parts on the fine dust sensor
  - <https://sensebox.github.io/books-v2/edu/en/projekte/Umweltstation/>
    - Learn about the different sensebox:edu sensors by creating your own environmental measuring station, which includes:
      - Temperature and Humidity
      - Experiments with light
      - UV-Sensor
      - Air pressure
      - **Note:** ignore the parts on the file upload to openSenseMap
- Next step is to test the installations of the simple sensors. Run them each a couple of minutes.
- Hints:
    - WiFi setup: Wifi is not active by default. Please activate it in the «mcu\_component\_test». You have to add the ssid and password as well. Note: We suggest you use your mobile phone to create a hotspot. The ETH Wireless Network is error prone.
    - The connection to senseBox is often broken. A workaround is to change the port in the editor, push the reset button at the senseBox device and change back to the

---

correct port.

- If you do not know the port, you can check the “Device Manager”. The system is shown as USB Serial Device.
- Note: Should your senseBox for some reason not be recognized by your PC anymore, please follow Jan’s suggestions here: <https://forum.sensebox.de/t/sensebox-edu-wird-nicht-erkannt/334/4>

### 1.2.4 Store the sensing data on the SD card

🕒 15min 🧑‍🎓 Group Work

The next step is to store the sensing data on the SD card. Please follow the old instruction guide (for Arduino Board) at [https://edu.books.sensebox.de/en/community\\_projects/Tutorial\\_ArduinoWeatherStation.html](https://edu.books.sensebox.de/en/community_projects/Tutorial_ArduinoWeatherStation.html) - step: “write the results to a file on a SD card”.

Hints:

- Also, look at the example file "Test\_microSD". You can use the constant "PIN\_XB\_CS" for determining the "chipSelect"-parameter.
- There is also an older instruction guide at <https://github.com/TodWell/books/blob/patch-1/en/basics/datenlogger.md>

### 1.2.5 Homework

Since your senseBox is able to sense and store ambient data in the classroom, you should now apply the learning in a real-world environment until next lab.

Use the given sensors of your senseBox to log the data at some random location where you live. Do not forget to bring back the box and the stored data!

# Geosensors with senseBox – Lab 02



Source: <https://sensebox.de/>

## Overall Objectives

You experiment with different sensors in gathering data from outdoors.

## 1 This lab: Setup and testing of advanced sensors and evaluation of the location for gathering the geodata

### 1.1 Goals

- Learn to setup and test more advanced geosensors (e.g., fine dust sensor)
- Decision making of the measuring position / area based on OpenSenseMap analytics and requirements (internet connection etc.)
- Ambient sensing / Data collection for one week

### 1.2 Tasks

#### 1.2.1 Recap of last week

 15min  Group Work

Extract and analyse your data (datalog.txt) and verify within peers type and values and check if they are plausible based on the sensing context (weather, ambient parameters, etc.)

## 1.2.2 Connect and test other sensors

🕒 75min 👥 Group Work

Now, we dive in the more complex components of senseBox and connect und test other sensors. Follow **the instructions** and connect further sensors such as:

- Fine dust: <https://sensebox.github.io/books-v2/edu/en/komponenten/sensoren/feinstaub.html>
- Ultrasonic distance: <https://sensebox.github.io/books-v2/edu/en/projekte/verkehrszaehlerv2.html>
- GPS:
  - Example code from old documentation: [https://github.com/sensebox/SenseBoxMCU-Lib/tree/master/examples/mcu\\_gps](https://github.com/sensebox/SenseBoxMCU-Lib/tree/master/examples/mcu_gps)
  - Examples from new documentation: [https://sensebox.github.io/books-v2/edu/en/projekte/Mobile\\_Station.html](https://sensebox.github.io/books-v2/edu/en/projekte/Mobile_Station.html)
  - **Note:** connect to I2C
- Display (in **German** only): <https://sensebox.github.io/books-v2/edu/de/komponenten/zubehoer/led-display.html>

Test the sensors and evaluate if they are displayed correctly at the LED-Display.

## 1.2.3 Research on sensor data providers

🕒 30min 👥 Group Work

Visit the website of <https://api.opensensemap.org/> or <https://docs.opensensemap.org/> and learn how other senseBox users are sensing or requesting ambient data:

Which parameters are mostly available? Which are missing?

More weather station data may be found with other providers. Check out the following website, APIs etc. and try to find out if some data is available for public (OpenData, SOS):

- NOAA land-based station data <https://www.ncdc.noaa.gov/data-access/land-based-station-data>  
Use the map tool of <https://www.ncdc.noaa.gov/isd/data-access> and get an overview of your living location's neighbourhood
- IBM Weather Company Data  
[https://console.bluemix.net/docs/services/Weather/weather\\_rest\\_apis.html#rest\\_apis](https://console.bluemix.net/docs/services/Weather/weather_rest_apis.html#rest_apis)
- Weather underground  
<https://www.wunderground.com/weatherstation/overview.asp>
- Luftdaten.info data from <https://luftdaten.info/> (archive: <http://archive.luftdaten.info/>)

You will use available datasets next week in the last lab!

## 1.2.4 Decide the measuring position

🕒 15min 👥 Group Work

Analyse different measuring positions / area based on requirements (internet connection, security etc.) at your living location together with OpenSenseMap analytics and decide for one location.

---

### 1.2.5 Homework

Since your senseBox is able to sense and store the advanced ambient data now, you should collect data outdoors until next lab.

Put your senseBox at your living location and sense and store data over several days. Observe these days manually the weather and other ambient observation and describe them qualitatively in a separate document.

Don't forget to bring back the box and the stored data!

# Geosensors with senseBox – Lab 03



Source: <https://sensebox.de/>

## Overall Objectives

In this lab, you merge your data with further open context data and visualize them in the web by using a 3d library.

## 1 This lab: Embed context data and visualize your project in the web

### 1.1 Goals

- Learn to download meaningful context data of web providers
- Learn to harmonize different datasets into an equal structure and an exchangeable format
- Learn to create a web map service and render geosensor data as a web application using in 3d js components

### 1.2 Tasks

#### 1.2.1 Recap of last week

 15min  Group Work

Extract and analyse your data (datalog.txt), and verify within peers, the types and values and check if they are plausible.

## 1.2.2 Search for third party data sources, download and merge them with yours

🕒 45min 🧑‍🤝‍🧑 Group Work

In this lab, you will build an ambient sensing application, similar to Opensensemap. Such a Volunteered Geographic Information (VGI) system, gains credibility if the data is supported by (scientific) evidence. Therefore, we want to enrich your data with data from different origins (sensors). The aim is not only to validate your single measurements, but the quality of the application you will develop improves through the additional information about the situational conditions of the surrounding area.

First, have another look at the sensor data providers from last week's lab.

Then, define a meaningful extent for which you want to acquire the information based on the geographical range, the density of measuring stations and the terrain characteristics.

Next, download appropriate data from Opensensemap as well as from other data providers.

Afterwards, clean and harmonize the data structure, that you can later easily compare the content.

Finally, format the data sets in exchangeable open data format. The result should be one aggregated or several individual datasets formatted as CSV.

## 1.2.3 Upload and publish data on ArcGIS Online

🕒 15min 🧑 Individual Work

As a next step, we publish the CSV data sets on a web server. We use Esri's cloud infrastructure "ArcGIS Online" to do so. You need an Esri Developer Account for that. Please create one at <https://developers.arcgis.com> .

Then, you need to upload your datasets to <https://maps.arcgis.com/home/webmap/viewer.html>, and publish your dataset as vectorised service layer (FeatureLayer). Feature Layers support querying, visualizing, and analysing the data.

Learn more about the procedures here: <https://doc.arcgis.com/en/arcgis-online/manage-data/publish-features.htm#GUID-DD47C642-38B4-4645-801D-2866FE90BF73> .

Give your service layer an individual style in a web map, which provides you a rich library of classifiers and symbologies. Do not forget to save the style changes in the layers' properties. Finally save the web map for later work.

Learn more about the procedures here: <https://doc.arcgis.com/en/arcgis-online/create-maps/change-style.htm>

If you want to be independent of locating your data on the web, you can keep the CSV on your own web server. But be aware that if CSV files are not on the same domain as your website, a CORS enabled server or a proxy is required.

## 1.2.4 Create a web-based map client

🕒 45min 🧑 Individual Work

Programming with ArcGIS:

With ArcGIS Online you can create a web app with a map using a configurable app, the Web AppBuilder, or the Operations Dashboard. These offer you various functionalities, such as layouts and colour schemes, editing and identification tools, social media feeds, side-by-side map viewers, and so on.



Learn more about the procedures here: <https://doc.arcgis.com/en/arcgis-online/create-maps/change-style.htm>

In this lab, the goal is to create a custom solution with an expressive visualization. Therefore, you will develop an application using HTML5, CSS and JavaScript (with the **ArcGIS Javascript API 4.x**).

Feel free to use any online code editor and open-source learning environment of your choice (e.g., <https://codepen.io>, <https://codesandbox.io/>, <http://collabedit.com>). They allow you to create code snippets, test them, and store them under a URL.

For ArcGIS Javascript beginners, follow the sandbox example and further step-by-step online tutorials on <https://developers.arcgis.com/javascript/latest/sample-code/intro-layers/index.html> (there you learn to create a Map, a SceneView and a checkbox input HTML element.)

#### Map Client Requirements:

Your web app should consist of at least **two sensor data layer** (yours and a third party) which one of them **switchable** (visible / none visible) through a map control (visibility toggles).

Specify the **layer properties** (renderers, opacity, maxScale, etc.), and give data insights in **popups** and **further functionality** of your choice.

#### Further Hints:

##### MapView (2d) and SceneView (3d)

- Quick Overview

<https://developers.arcgis.com/javascript/latest/guide/index.html#3d-support>

- Sample:

<https://developers.arcgis.com/javascript/latest/sample-code/introsceneview/index.html>

##### Layers

- Introduction to Vector Layers (Feature Layers)

<https://developers.arcgis.com/javascript/latest/sample-code/layers-featurelayer/index.html>

- Introduction to CSV Layer

<https://developers.arcgis.com/javascript/latest/api-reference/esri-layers-CSVLayer.html>

- Sample with a WMS Layer

<https://developers.arcgis.com/javascript/latest/sample-code/layers-wms/index.html>

- Openstreetmap

<https://developers.arcgis.com/javascript/latest/sample-code/sandbox/index.html?sample=layers-osm-3d>

##### PopUps

- Introduction to PopUps

<https://developers.arcgis.com/javascript/latest/sample-code/intro-popup/index.html>

---

### 3d Rendering (3d Symbols)


- Creating visualizations based on numeric values, attributes types with extrusion, opacity, rotation etc.  
<https://developers.arcgis.com/javascript/latest/guide/creating-visualizations-manually/index.html>
- Visualizing points with 3D symbols:  
<https://developers.arcgis.com/javascript/latest/guide/visualizing-points-3d/index.html>
- API:  
<https://developers.arcgis.com/javascript/latest/api-reference/esri-symbols/IconSymbol3DLayer.html>

### Ready to use Widgets

- <https://developers.arcgis.com/javascript/latest/sample-code/intro-widgets/index.html>

Finally, post your map (i.e., the URL to your application) on Moodle.

### **1.2.5 *Resumé on the project***

 15min

Review and reflect on the project with your peers. The results are collected in a storymap, which is available at Moodle or at <https://arcg.is/HqGCn>