Practical course: Air-quality measurement

Course script

Version: 1.1

Date: 16.5.2022

Author: Daniel Kühbacher

Content

[Prerequisites and course schedule 3](#_Toc102136811)

[Measurement station 3](#_Toc102136812)

[EC-Sense air-quality sensors 4](#_Toc102136813)

[Task 1 – Commissioning the sensing station. 5](#_Toc102136814)

[Measurements 6](#_Toc102136815)

[Task 2 – Measure the air quality 7](#_Toc102136816)

[Evaluation and assessment of the acquired data 8](#_Toc102136817)

[Task 3 – Analysis of the acquired data 8](#_Toc102136818)

[Task 4 – Analysis of LfU data (Landshuter Allee) 8](#_Toc102136819)

# Prerequisites and course schedule

Please watch our introduction video before joining the course, since it provides a comprehensive overview of the topic. Apart from that, no further preparation is necessary. The course is held on Monday and lasts from **10:00 to 13:00**.

We will cover the following topics in the practical course:

* Introduction to environmental sensing; Recap of the introduction video
* Commissioning of the sensor boxes
* Air quality measurement
* Evaluation and assessment of the collected data
* Evaluation and assessment of public air quality data

# Measurement station

A measurement station was developed specifically for this course. Figure 1 depicts the basic structure of the measurement station. It uses a Raspberry Pi as processing unit and implements a 7’’ touchscreen display on the front. It holds three sensors to measure NO2 (Nitrogen Dioxide), CO (Carbon Monoxide), and O3 (ozone). These gases are the main pollutants in cities next to PM10 (Particulate Matter) and are also officially measured by the LfU (Bayrisches Landesamt für Umwelt). As suggested by the manufacturer of the sensors, they are installed in a dedicated, ventilated measuring channel. The ventilator can also be controlled by the Raspberry and sucks fresh air into the channel on demand. During measurement, the ventilator is turned off to avoid fluctuations of the signal due to pressure fluctuations (was suggested in the datasheet) but also to save energy.

Ein Bild, das Text, Uhr, Gerät enthält.

Automatisch generierte Beschreibung

Figure 1 - Basic structure of the sensing station.

### EC-Sense air-quality sensors

“TB600B series modules (see Figure 2) combine various high-precision sensing technologies. The gas sensor is a small-in-size Solid Polymer Electrochemical Sensor from EC Sense, detecting very low concentrations of gases accurately and reliable. The module serves a UART digital output for ease of use and comes pre-calibrated eliminating the tedious work of calibration.

Solid polymer electrochemical technology is based on the principle of electrochemical catalytic reaction, detecting the output signals of the electrochemical reactions of different gases, and accurately measuring the gas concentration through the signal.

The sensor is composed of three electrodes in contact with the electrolyte. A typical electrode consists of a large surface area of noble metal and other materials. The electrode, electrolyte, and the surrounding air are in contact, and the gas diffuses into the working electrode. Here the gas will be oxidized, this causes a current, which is proportional to the gas concentration and can be measured.”[[1]](#footnote-1)

In Table 1 the main characteristics of the implemented sensors are listed. The rightmost column gives you a conversion factor between the measured value in ppm (parts per million) and the mass/ volume. This conversion depends on the molecular weight (g/mol) of the compared substances the surrounding temperature and the air pressure. Considering the ideal gas law at a temperature of 25°C at a pressure of 1 atm, we can derive a molar Volume . Using the molar mass of e.g., NO2 , we can derive the following equation:

Ein Bild, das Text, Elektronik enthält.

Automatisch generierte Beschreibung

Figure 2 - EC-Sense TB600B sensor modules

Table 1 - Overview of sensor characteristics

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sensor type | Range  [ppm] | Det-Limit  [ppm] | Resolution  [ppm] | FS-accuracy error | Conversion factor  @ 25 °C/ 1 atm | |
| **O3** | 0-5 | 0,01 | 0.001 | ± 5% FS | | 1ppb = 1,96 µg/m3 |
| **CO** | 0-10 | 0,1 | 0.01 | 0 – 5 ppm ± 2-5%  5 – 10 ppm ± 5% | | 1ppb = 1,15 µg/m3 |
| **NO2** | 0-5 | 0,05 | 0.001 | ± 5% FS | | 1ppb = 1,88 µg/m3 |
|  | Temperature error: ± 0.2 °C  Humidity error: ± 2% | | | | | | |

## Task 1 – Commissioning the sensing station.

In this task, you will set up the sensing station and test its functionalities. You should be able to read out the sensor values and control the ventilator with a simple python script. Follow the steps below:

1. Open *ecsense.py* and have a look at the already implemented class that controls the sensor. Since all sensors are from the same producer, they are identical to operate. When you run the file, the content of the function   
   if \_\_name\_\_ == "\_\_main\_\_" is executed.
   1. Have a look at the class and try to understand what each method does.
   2. In the constructor of the class one connection setting is missing. Have a look into the datasheet of the sensor (folder documents) and find the missing baudrate setting.
   3. The read command in the read method is missing. Look for the right command in the datasheet.
   4. Test the class by running it in the IDE. You should now get back some sensor values.
2. Now, since we can control and read one sensor, we want to apply this functionality to all three of them and control the ventilator.
   1. Open the file *sensor\_test.py* and implement the ventilator test as described in the code comments.
   2. Use the class EcSensor and implement a readout for each sensor connected to the Raspberry Pi. Print the sensor values in the console. Hint: Have a look to the file *ecsense.py*
3. Finally, have a look at the file *no-gui.py.* This simply runs what you did in step 3 in a loop and saves the data in a sqlite3 database. The image below shows a flow-chart of the main measurement cycle.

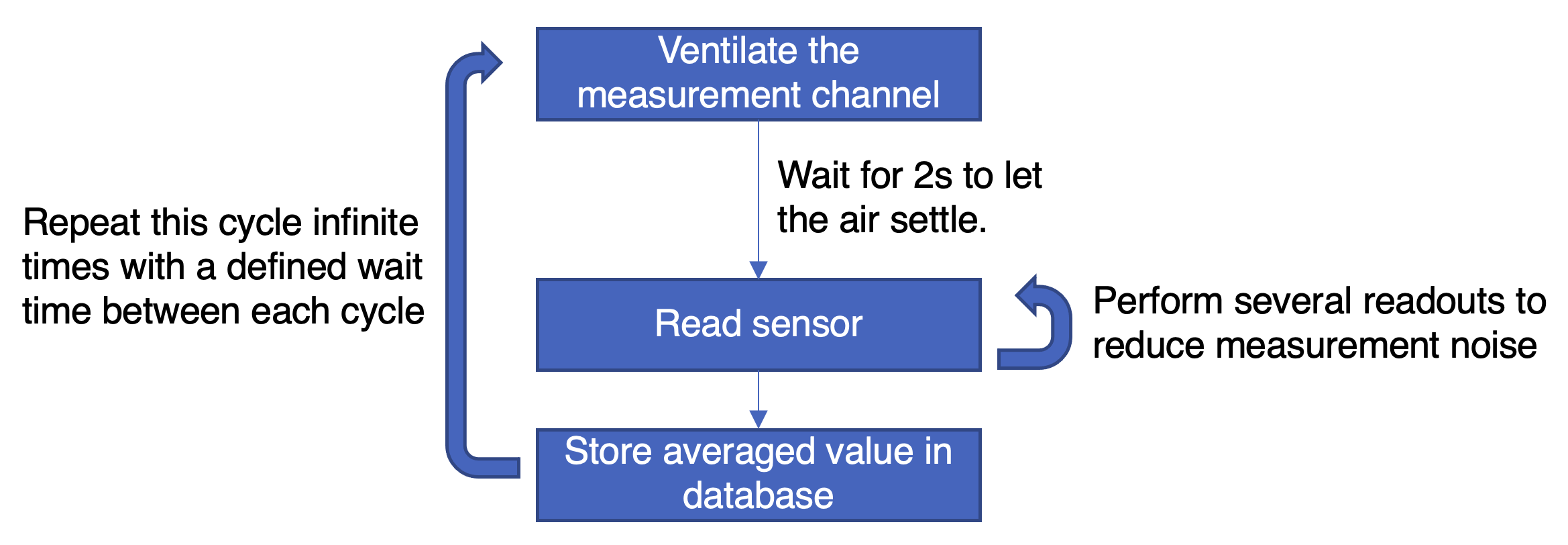


Figure 3 - Measurement cycle of the main program

# Influence of the placement on the measured air quality

Recently, the topic of nitrogen oxide in major German cities has been a constant theme in the media. The public interest in precise measurements with high temporal and spatial resolution has therefore increased considerably. So far, however, there is no optimal method that combines all these requirements.

In the following experiment, we will evaluate the performance of our low-cost station and compare the acquired values at different distances to the street. In Figure 4 you can find three different positions with the following characteristics:

* Position 1 is close to a crossing, where cars and trucks tend to wait and accelerate when the light turns green.
* Position 2 is like Position 1 but on a spot that complies with the official regulation “Anlage 3C of the BImSchV”: hight of about 1,5 – 4 m; > 25 m away from an intersection; < 10 m distance to the street
* Position 3 is far off the street.

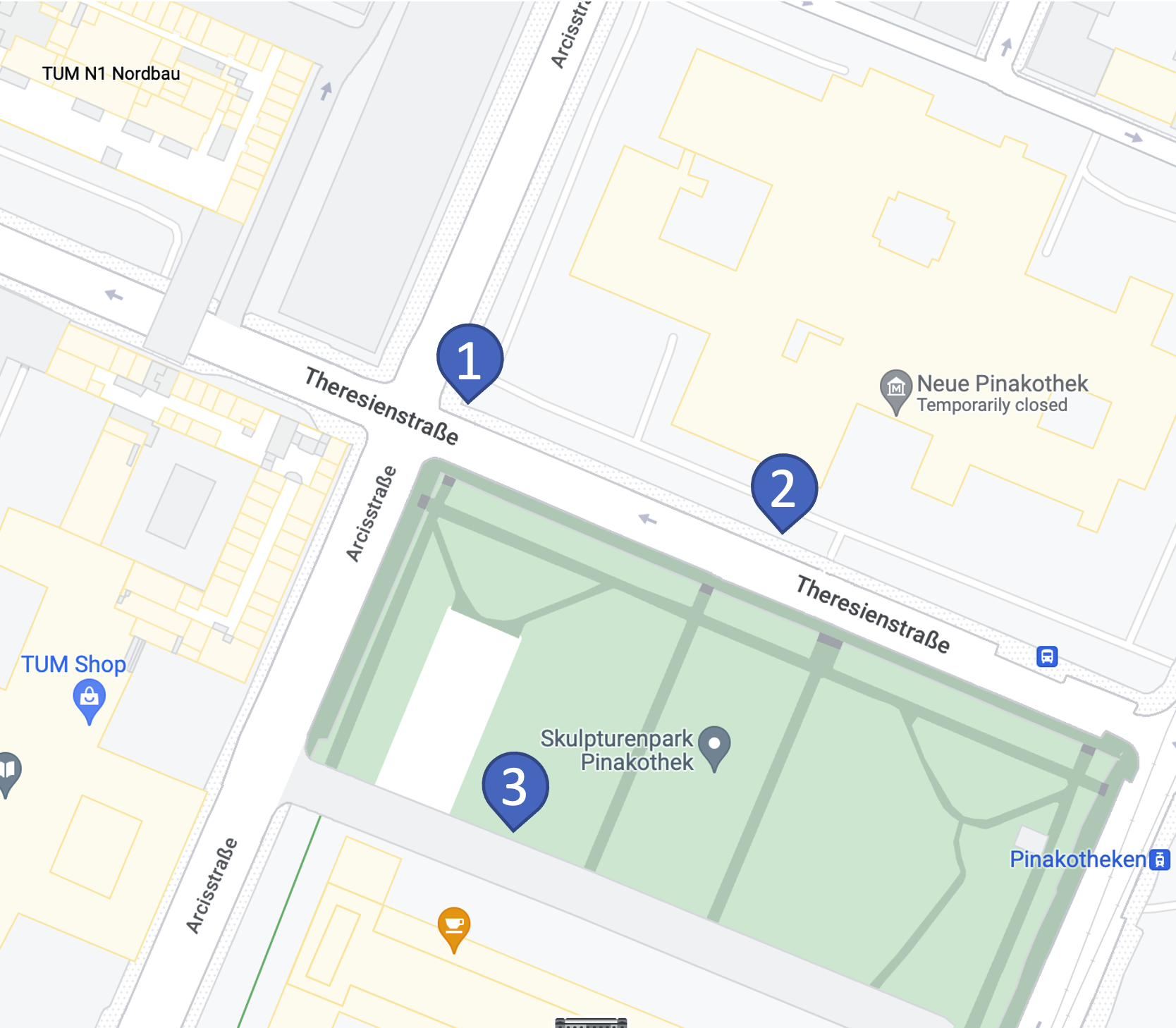


Figure 4 - Measurement positions for three sensors

## Task 2 – Measure the air quality

In this task we will measure the air-quality in different locations next to the university.

Follow the steps below:

1. Mount the sensor on the tripod and prepare it for the field measurement. Check if the sensor works by running your test script *sensor\_test.py.*
2. Choose a location and start the measurement. Measure for at least 30 minutes.
3. During measurement, discuss the following questions with your partner:
   1. Do you expect different values at different locations? Why and what conclusion would you draw from it?
   2. Are the sensors a good choice for the measurements conducted? Have a look at Table 1 (Detection limit, Error) and compare the values to typical expected values in urban environments
   3. Think about other use cases besides cities where air-quality measurements are necessary.

# Evaluation and assessment of the acquired data

Finally, it is time to evaluate the acquired data and calculate some representative statistical values to compare. You will use Jupyter Notebook for this. Jupyter is an IDE used by many data scientists and in many labs around the world. It can be used with a variety of programming languages (we use Python) and runs in your browser. To start the notebook, open a terminal window and navigate to the *device\_data* folder. Type in *jupyter notebook* to start the application. A browser window will open and you can start the notebook *analysis\_notebook.ipynb.*

## Task 3 – Analysis and comparison of the acquired data

Follow the steps described in the notebook.

## Extra task – Analysis of LfU data (Landshuter Allee)

Start jupyter notebook in the folder *lfu\_analysis.*

Follow the steps described in the notebook.

1. TB600B-O3-5 Technical Specification – EC Sense [↑](#footnote-ref-1)