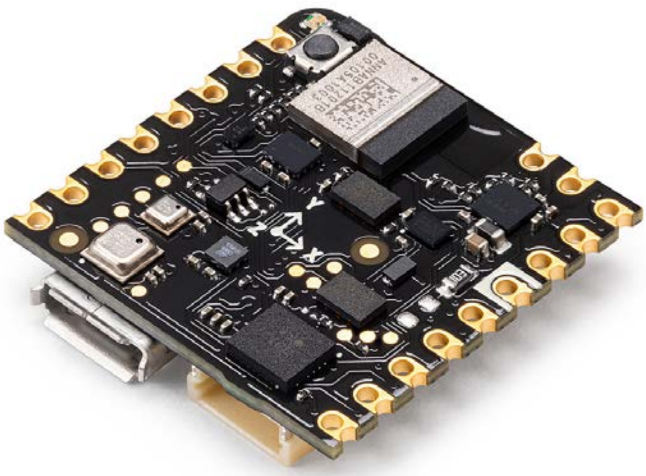
Sensors Lab Course

Gyroscope, accelerometer, pressure sensor, magnetometer, gas sensor

Dr. Jochen Kieninger, Prof. Dr. Stefan J. Rupitsch

Winter term 2022/23



Laboratory for Electrical Instrumentation and Embedded Systems

IMTEK – Department of Microsystems Engineering

University of Freiburg

Contact: kieninger@imtek.uni-freiburg.de

Introduction

This lab course is a mandatory part of the lectures *Sensors* and *Sensorik* from Master of Science programs in *Microsystems Engineering*, *Mikrosystemtechnik*, and *Embedded Systems Engineering* at the Faculty of Engineering. Through the three modules of the lab course, we provide you with hands-on experience to deepen the knowledge from the lectures. We like you to spark your interest in sensor applications playfully but simultaneously expect a university-level performance during the experimentation and while writing your reports.

All experiments use the Arduino Nicla Sense ME platform (Figure 1), which comprises four state-of-art integrated sensors from Bosch Sensortec, an Arm Cortex M4 processor, and Bluetooth connectivity. The different sensors are an inertial measurement unit (IMU) measuring acceleration and rotation (BHI260AP), a pressure sensor (BMP390), a magnetometer (BMM150), and a gas sensor (BME688) providing deduced parameters like equivalent CO_2 and volatile organic compounds (VOC) concentrations together with temperature and humidity.

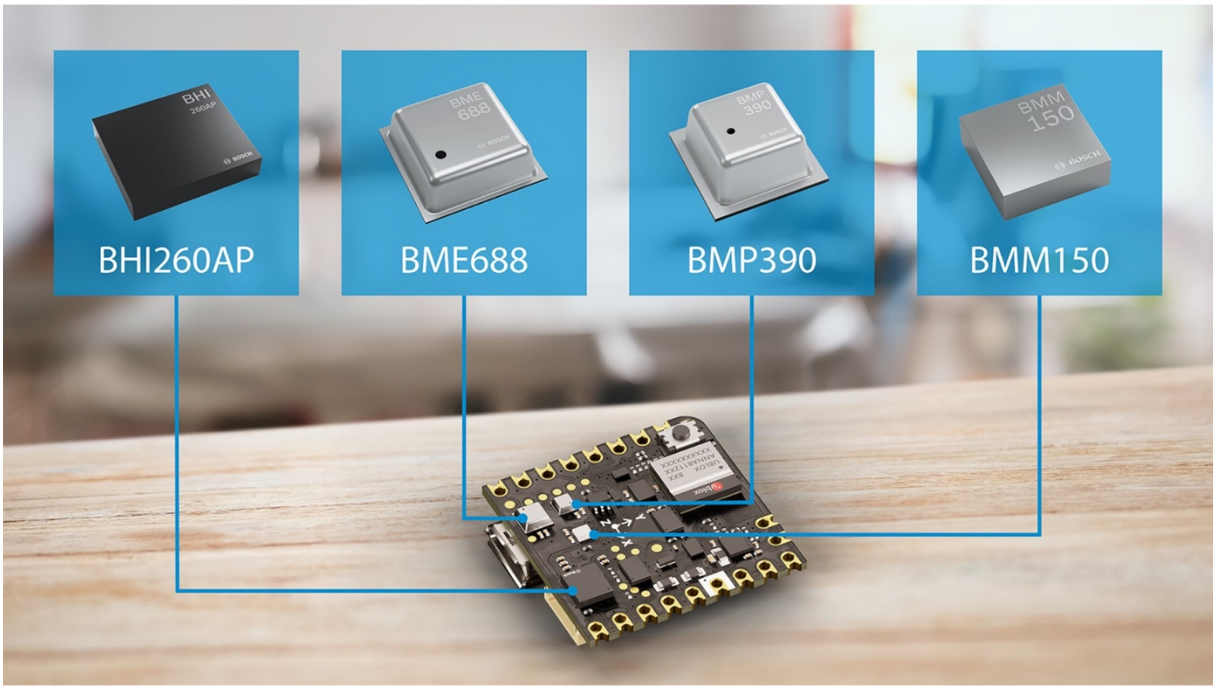


Figure 1: The Arduino Nicla Sense ME board comprises different sensors and connectivity modules. The board allows programming with the Arduino IDE using a USB connection. Image: Bosch Sensortec.

The lab course consists of self-learning modules for which each student borrows one Nicla Sense ME board from the faculty library[1]. Nothing else is needed besides a computer with a USB port, preferably a notebook. Within the given schedule (i.e., deadlines for report submission), you can work on the experiments at your own pace.

While there will be no physical meetings, please stay connected with your colleagues through the University's online learning platform[2] (ILIAS). **You can ask questions at any time in the forum in the ILIAS group of the lab course.** We will help you there promptly, and other students are welcome to suggest solutions. Please support each other – in the lab course, you

[1] Address and opening hours: https://www.ub.uni-freiburg.de/?id=3281
[2] https://ilias.uni-freiburg.de

are not in competition with each other but should all explore the exciting world of sensors together. If you are new to the University, you can also use the forum to find mates to work together.

Learning Objectives

The learning objectives are the goals we want you to reach until the end of the lab course.

1. You have practical experience with different state-of-art sensors (accelerometer, gyroscope, pressure sensor, magnetometer, gas sensor, humidity sensor, temperature sensor) and an embedded sensor platform.
2. You can program an embedded system to interface with different sensors and provide the data to a connected computer.
3. You know how to perform sensor measurements according to scientific standards.
4. You can analyze sensor data (filtering, integration, differentiation).
5. You can document and appropriately discuss your measurements in a report.
6. You understand the working principles of the different sensors and relate your measurements to the limitations of the sensor principle.

Formalities

The lab course accounts to 2 ECTS translating into a workload of 60 hours over the semester, which is, on average, 4 hours per week during lecture time.

To pass the course, you need all three reports to get acknowledged. There is a deadline for each report. The reports must be submitted as a PDF file. The used microcontroller code needs to be submitted as a zip file. You will find an upload option for these files in the ILIAS course of this lab course.

In case a report is insufficient, you have the option to rework it and **resubmit within one week**. In total, you have three options to rework. Please be aware: if you need all three rework options to pass the first, you must succeed with the other two reports in the first submission.

We encourage you to do the practical experiments with your colleagues (small groups of up to 4 students). However, writing the report and discussing the results need to be done by each student individually. **When you share code or perform experiments together resulting in the same data, you must indicate in your report the code or data and declare with whom you collaborated.**

Before the three modules, we reserved additional time for an onboarding phase to familiarize you with the software and hardware platform. Here no report from your side is required; instead, we provide a sample report reflecting this experiment and illustrating our expectations.

Disclosure

The Arduino Nicla Sense ME boards were kindly provided by Bosch Sensortec.

Schedule

We suggest you a schedule how to work on the lab course. However, the lab course is self-paced; only the red marked dates (hardware distribution/return and submission deadlines for the report) are binding.

Date	Task
Week 1 (17.-21.10.22)	Getting started, collect the hardware from the faculty library
Week 2 (24.-28.10.22)	Module 0: Gyroscope and Acceleration Sensors ("onboarding")
Week 3 (31.10.-04.11.22)	Module 0: Gyroscope and Acceleration Sensors ("onboarding")
Week 4 (07.-11.11.22)	Module 1: Acceleration and Pressure Sensors
Week 5 (14.-18.11.22)	Module 1: Acceleration and Pressure Sensors
Week 6 (21.-25.11.22)	Module 1: Acceleration and Pressure Sensors
Week 7 (28.11.-02.12.22)	Module 1: Acceleration and Pressure Sensors
Friday, 02.12.22	Submission of the report to Module 1
Week 8 (05.-09.12.22)	Module 2: Magnetic Sensors
Week 9 (12.-16.12.22)	Module 2: Magnetic Sensors
Week 10 (19.-22.12.22)	Module 2: Magnetic Sensors
Week 11 (09.-13.01.23)	Module 2: Magnetic Sensors
Friday, 13.01.23	Submission of the report to Module 2
Week 12 (16.-20.01.23)	Module 3: Gas Sensors
Week 13 (23.-27.01.23)	Module 3: Gas Sensors
Week 14 (30.01.-03.02.23)	Module 3: Gas Sensors
Week 15 (06.-10.02.23)	Module 3: Gas Sensors
Friday, 10.02.23	Submission of the report to Module 3
Until 15.02.23	Return the hardware to the faculty library

Submission

The report must be submitted as a PDF file. The used microcontroller code needs to be submitted as a zip file. You will find an upload option for these files in the ILIAS course of the lab course. You must strictly comply with following naming convention:

`SensLab_M`*`<Module>`*`_`*`<Family name>`*`_`*`<Matriculation no>`*`[_rev`*`<revision>`*`].pdf` or `.zip`

For example, Isaac Newton would have filed his report as `SensLab_M1_Newton_3141592.pdf` and the third revision as `SensLab_M1_Newton_3141592_rev3.pdf` (for sure he would have struggled with today's way of writing his laws ...).

Arduino Nicla Sense ME platform

The *Arduino Nicla Sense ME* is a versatile development board with many different sensors allowing simple programming using the Arduino Integrated Development Environment (IDE). At the same time, it is a platform enabling complex *artificial intelligence* (AI) applications with substantial computational demands. The platform allows for advanced *Internet of Things* (IoT) applications with data processing at the sensors (*edge computing*) before a transmission unit (e.g., an external WiFi module) transfers the processed information to the cloud. A built-in module features connectivity over Bluetooth Low Energy (BLE); a Universal Serial Bus (USB) allows cable-bound serial connection to a computer. The board design focuses on low power consumption with a supply from a battery but also allows powering from the USB.

Within the lab course, we focus on the standalone operation of the board and programming with the Arduino IDE. We use the USB port for the power supply, programming, and data retrieval. You are encouraged to explore the other functionalities to deepen your skills beyond the mandatory tasks.

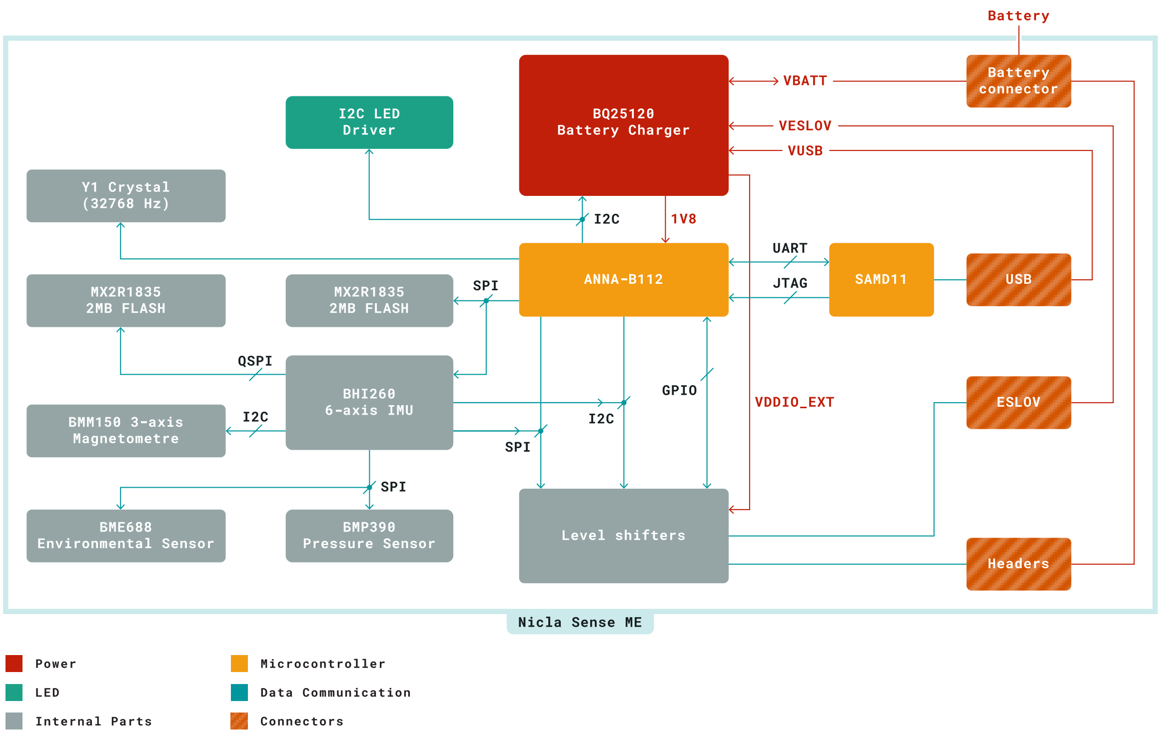


Figure 2: Nicla Sense ME block diagram [1]. The ANNA-B112 module hosts the main microcontroller. The magnetometer (BMM150), the gas sensor (BME688), and the pressure sensor (BMP390) connect to the inertial measurement unit (BHI260), which not only comprises the accelerometers and gyroscopes but also acts as a smart sensor hub with its processor enabling AI applications.

The ANNA-B112 Bluetooth module comprises the board's main microcontroller with an ARM Cortex M4 architecture and 512 kB flash memory for the program code. The block diagram (Figure 2) illustrates the other system components and reveals the sensor's connections. The different sensors connect to the BHI260 inertial measurement unit (IMU), which also contains a microcontroller acting as a sensor hub. The board's data sheet [1] provides a more detailed description of the system architecture and the different components.

We use the pre-installed firmware in the BHI260 for the lab course. The firmware provides interfaces called *virtual sensors*. A virtual sensor may represent the raw values of the sensor, calculate values based on calibration data, or consider other parameters (e. g., temperature) to provide compensated values. We focus on raw values in the lab course as much as possible, while real applications prefer compensated data. Other virtual sensors provide fused data, e.g., for orientation or advanced features like gesture recognition. In the description of each module, we provide which virtual sensor you should use. Once more experienced, you can run an example program[3] that generates a list of all available virtual sensors.

Tools

A simple way is **programming the microcontroller** of the Nicla Sense ME with the *Arduino IDE*. An integrated development environment (IDE) combines an editor with a compiler and, if necessary, other tools. The Arduino IDE is available for Windows, Linux, and macOS and includes the *gcc* (GNU C compiler). Apart from the editor, the IDE contains some basic libraries and allows the installation of additional ones easily. Besides compiling, the IDE can transfer the program to the microcontroller's flash storage.

Furthermore, the IDE contains a console for the serial interface and a simple plot program for data received via the serial interface. The free Windows software *PuTTY*[4] or *HTerm*[5] are alternatives to the Arduino IDE's serial console and facilitates saving the gathered data. If you are already familiar with programming, you may also use your preferred IDE, such as Microsoft Visual Studio Code which provides Arduino support through extensions. In the forum, we provide only support for the Arduino IDE, PuTTY, and HTerm.

Module 0 guides you through the necessary online resources to get the Arduino IDE to run for programming the Nicla Sense ME.

[3] https://github.com/arduino/nicla-sense-me-fw/tree/main/Arduino_BHY2/examples/ShowSensorList, also available in the Arduino IDE under examples once the Arduino_BHY2 library is installed
[4] https://www.putty.org
[5] https://www.der-hammer.info/pages/terminal.html

Writing a Scientific Lab Report

This section introduces you to how to write a scientific lab report. The goal should be that another scientist with a similar background and the same equipment can reproduce your results. That means you must balance over-educating and skipping important details hiding how you did the experiments. Length is not a quality criterion. An excellent report is concise, which means as short as possible while mentioning everything of importance in the above-mentioned sense. A good guideline is to imagine you, in three years, should be able to reproduce everything without starting the investigations from scratch. For Module 0 (the "onboarding"), we provide a sample report illustrating the essential aspects.

Before Writing

Writing an excellent scientific lab report already starts before the experiment. Most importantly, carefully consider which parameters might influence your measurements and adjust accordingly. Try to figure out as much information as possible about your equipment, e.g., which principles the sensors rely on or which value range is possible. It is also helpful to look over the learning objectives of the student's lab to understand what we expect from you. Discussing with your colleagues helps you better understand the topic and shows yourself if you lack in some parts.

While doing the experiments, you should document everything! Do not underestimate how fast one forgets the details. Don't count on being an exception to that rule. Reconstructing information later is a colossal waste of time. You may use your smartphone's camera to document the setup.

Content

While there are no internationally binding guidelines for writing a lab report, a de facto standard has evolved following the structure of research papers. The report should be structured as follows:

- The **title page** must indicate what kind of report it is, who wrote it (from which institution), and the date of the last update. Here, you need to state the lab course's name, the module, your name, your matriculation number, and the date. If you use data generated in a team, state your colleagues' names and matriculation numbers.
- The **Introduction** (*Einführung*) provides short background information and an overview of the experiment. A precise formulation of the experiments' goals helps both the writer and the reader.
- You can keep the **Theory** (*Theorie*) section concise, assuming a generally educated engineer as the reader. Never recapitulate Newton's Laws in all detail, while more specific relationships like the Barometric Formula are worth mentioning. For the lab course, use this section to briefly explain the theory of the sensors' working principles.
- In the **Methods** (*Methoden*), you should describe the setup and procedures so that you could repeat the experiment in three years. Also, consider listing software versions if you suspect influence on the measurement results. Any environmental conditions, such as temperature, exceptional weather conditions, etc., are essential.

- **Results and Discussion** (*Ergebnisse und Diskussion*) can be separated or combined into a single section, as it has become common in many scientific journals. Also, if combined, make it clear while writing what the result is (what did you observe) and what the discussion is (your interpretation). The results (your observation) are always valid, while the discussion might get outdated when newer research results improve our understanding. That is why researchers in fundamental sciences prefer separate sections; for a lab report, a combination is the better choice.
- The **Summary** (*Zusammenfassung*) wraps up the results and major findings. Keep it short and easy to read. 3-5 sentences are an ideal guideline for the modules in this lab course.
- The **References** (*Quellenangaben*) contain all your sources (see below for details).

Language

You can choose between German or English (either British or American spelling). Be consistent in your choice. That includes using the correct decimal separator ("." vs. ",") and using the same language for labeling your plots. Scientific language is simple. Focus on short and precise sentences. Avoid bloomy formulation or literary style that excludes non-native speakers. Please remember that you are at a university, so write complete sentences and use a spell checker!

Quantities, Numbers, and Units

You must use SI units, i.e., no imperial units. SI stand for the *Système international d'unités*, the International System of Units:

- There are seven *base units*: m, kg, s, A, K, mol, and cd
- For convenience many *derived units* are in use: rad, sr, Hz, N, Pa, J, W, C, V, F, Ohm (Ω), S, Wb, T, H, °C, lm, lx, Bq, Gy, Sv, and kat
- Expression of small or large quantities benefit from *prefixes*: Y (10^{24}), Z (10^{21}), E (10^{18}), P (10^{15}), T (10^{12}), G (10^{9}), M (10^{6}), k (10^{3}), h (10^{2}), da (10^{1}), d (10^{-1}), c (10^{-2}), m (10^{-3}), μ (10^{-6}), n (10^{-9}), p (10^{-12}), f (10^{-15}), a (10^{-18}), z (10^{-21}), and y (10^{-24})
- Possible *exceptions*, which are also accepted in this lab course, account for traditions in the different subjects: min, h, d, y, l (liter), g (gram), t (ton), dB, eV, Da, Å (10^{-10}), M (molar), ppm, and ppb

Take care to report only **significant digits**. You should calculate with full precision but finally round the results to meaningful digits. Never pretend to have higher precision in your results than your measurement system has!

Example: When the 13th person entered the elevator, there was an alert, and the display said, "Overweight: 973 kg". What was the average mass m of the people in the elevator?

$m = \frac{973\ \text{kg}}{13} = 74.8\cancel{4615}$ kg … round to the significant digits (three in this case)

You can use the standard deviation for a measurement series with normally distributed data to judge the significant digits. In case you know about the sensor's accuracy, consider this information in your reporting of results.

Example: If the above-mentioned elevator's balance has an accuracy of ±1 kg the average weight is $m = 75$ kg.

Typesetting of Equations and Units

Consistent typesetting of equations and units helps the reader to oversee your report and shows the professionality of your work. Use the italic font for your variables and the upright font for labels and units. Do not forget the blank between numbers and units (preferably a "protected space" in case your text processor supports it to avoid line breaks in between). Use the correct decimal separator ("." vs. ",") depending on your language.

Example: The length l of a marathon (indicated by the label "m") is $l_\mathrm{m} = 42.195$ km. Refer to different lengths by l_i $(i = 1, 2, 3)$.

Introduce each variable at its first occurrence. Sometimes a list with all variables, especially in an interdisciplinary context, helps the reader.

Never use old-fashioned pseudo-code to typeset equations in a report.

Example: Use $P = I^2 R$ instead of $P = I\hat{}2 \ * R$.

Citations and References

Statements in the text, values, data, and images from another source require appropriate citation. You can cite another work within the text or in the caption of a figure by introducing a numeric or alphanumeric reference.

Examples:

- Citing a book is as easy as traveling through the galaxy [1].
- Most microsystems engineering students have read the famous lecture of Richard Feynman. Its transcript can also be found as journal publication [2].
- In case you cite a web page like our faculty's [3] be sure to mention the date of last access.

In a section at the end of your report, you provide the source of your references. Footnotes are also sometimes used for the sources but are less common in the engineering disciplines.

Examples:

[1] D. Adams, The hitchhiker's guide to the galaxy. New York, NY, USA: Del Rey/Ballantine, 2005.

[2] R. Feynman, "There's plenty of room at the bottom", Journal of Microelectromechanical Systems, vol. 1, no. 1, pp. 60-66, 1992.

[3] "The Faculty of Engineering – Solutions for the future — Faculty of Engineering", Tf.uni-freiburg.de, 2022. [Online]. Available: https://www.tf.uni-freiburg.de/en. [Accessed: 01-Aug-2022].

There exist many citation styles; the above examples used the *IEEE style*. You are free to use any style typical in the engineering disciplines but ensure consistency. Citing whole sentences or paragraphs, common in arts and humanities, is unusual in the engineering disciplines or natural sciences and should be avoided.

Do not confuse the scientific correct referencing with copyright aspects. The latter is not an issue in an unpublished lab report, but copyright-free material also requires proper referencing. If you publish your lab report, ensure to not violate any copyright.

Plots

Scientific plotting is essential to report data accurately and display measurement results in a report. Unfortunately, students often pay less attention to correct and meaningful plots. Sometimes even the axes have no labels, or units are missing. Please pay utmost attention to the graphical representation of your data while writing your report.

Figure 3 shows an example plot for two-dimensional data. Some rules and hints help you produce correct and meaningful diagrams:

- Each axis needs a label comprising the unit, and the scaling gets clear, which means you need at least two ticks to relate the data points to their numerical correspondence.
- Avoid the unit in rectangular brackets ("Length [m]") as in some engineering disciplines, the rectangular brackets are also used as an operator or, in analytical chemistry, represent the concentration.
- If a quantity is dimensionless, you can use "1" instead of the unit, or if the unit is device-specific, indicating an arbitrary unit by "a. u." is possible.
- When of importance, scale to include the zero in your diagram but avoid the data sticking only to a small fraction of the diagram's area. Indicate the usage of nonlinear scaling (e.g., logarithmic scaling).
- Avoid connected lines if only a few discrete points are available. For a high amount of data points, you may use lines instead.
- Use legends to clarify your plots and labels (letters, arrows) to highlight specific sections you want to refer to in your discussion.
- Take care of color-blind people. A rough approach could be to check if the diagram would also work as a greyscale image. You may use different markers or line styles.

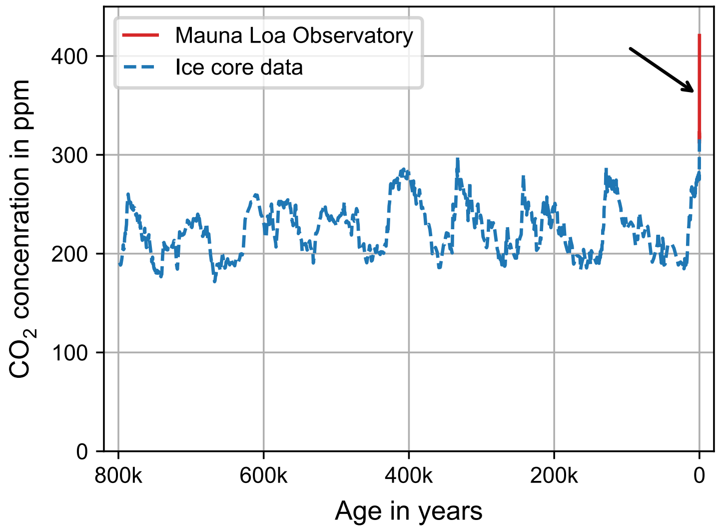


Figure 3: CO_2 concentration in the atmosphere: the Keeling curve (marked with the arrow), named after Charles David Keeling from the Scripps Institution of Oceanography, is measured since 1958 at the Mauna Loa Observatory [2]. The curve is much more alarming in a larger context combined with paleoclimatic ice core data [3].

Plagiarism and Cheating

Plagiarism and cheating are the most critical frauds in science. Worst, reporting fake data hinders scientific progress, harms people, and affects science's reputation. That is why **we seriously pursue plagiarism and cheating in the lab course, i.e., you fail.** If you do that repeatedly, the University may exclude you from further studies. You all know it, but once again: using someone else's texts, images, code, data, solutions, etc., without giving the source is plagiarism. Sources include books, lab course manuals or lecture notes, webpages, and your colleagues. Never make up data you didn't measure, and never use calculations or tools to eliminate unwanted data points without clearly mentioning them. If you get caught, you won't have saved any time!

Tools

The essential tool for report writing is the **text processor**. Mainly two possibilities exist, *Microsoft Word* (or its free pendants in *Open Office/LibreOffice*) and *LaTeX*. Word had a bad reputation, to some extent originating in the poor typesetting of equations in older versions and the possibility to write and format texts in an unstructured way. If you strictly use style sheets (called “Styles” or “Formatvorlagen”) for the formatting and resist all Word Art-like features, scientific writing is easily possible. The equation editor improved a lot in Office 2016, allowing some LaTeX code and related shortcuts to speed up writing equations. Students can access Microsoft Office 365 Pro Plus[6] for a small annual price (below 5 €). Microsoft Word runs on Windows and macOS, the free pendants on Linux as well. The alternative to Word is *LaTeX*, which features high-quality typesetting and facilitates the structuring of your document without the possibility of getting concerned about the final layout while writing. *Overleaf*[7] is a convenient online LaTeX environment that allows report writing in the cloud. The free plan is

[6] https://bildung365.de/
[7] https://www.overleaf.com

well-suited for a lab report or thesis. Generating PDF is easily possible in Word (save the document as PDF) or LaTeX (use pdflatex) without additional software. A warning, which is especially helpful for some LaTeX users: **Don't forget the content over the form!**

For **references** in a lab report, you could use manual formatting or use free (advertisement-financed) web services, e.g., *Cite This For Me*[8]. You may also use a professional reference manager, for larger documents such as a thesis highly recommended. There are several options, e.g., *Mendeley*[9] which offers a free version, or *Citavi*, for which a campus license exists[10].

Often **data processing** and **plotting** go hand in hand. *MATLAB* by MathWorks is a programming language and numeric computing environment well suited for signal processing, including plotting functionality. MATLAB is available for Windows, Linux, and macOS; there is a state license[11] providing students MATLAB for free. An alternative would be *Origin* by OriginLab, a Windows software dedicated to scientific plotting and data processing. Students can access OriginPro Campus license[12] for free. If you prefer plotting without writing code, Origin is a good choice. In contrast, Microsoft Excel is not well suited for scientific plotting (also, you might, with a lot of effort, extract similar-looking results). If you are into *Python* programming, libraries like *pandas* for data handling and *matplotlib* for plotting are also good choices to obtain publication-grade scientific plots efficiently. However, learning Python just for lab courses is overkill. In the forum, we provide support for MATLAB only.

Less relevant than the previously described tools but still a necessity is **drawing software**. Often a block diagram or a sketch explains relationships much better than lengthy text. While professional tools like *Adobe Illustrator* are out of reach for students' budgets, free tools like *draw.io* are a good compromise between capability and learning effort for a lab report. The latter also provides various shapes for flow diagrams and elements for electrical circuits. Draw.io is available as a desktop application[13] or cloud service[14] to draw directly in a browser.

---

8 https://www.citethisforme.com/
9 https://www.mendeley.com/
10 https://www.rz.uni-freiburg.de/en/services/procurement/software/citavi-en
11 https://www.rz.uni-freiburg.de/en/services/procurement/software/matlab-license
12 https://www.rz.uni-freiburg.de/en/services/procurement/software/info-originpro
13 https://github.com/jgraph/drawio-desktop/releases
14 https://app.diagrams.net

Module 0: Gyroscope and Acceleration Sensors ("onboarding")

Do you remember riding a merry-go-round as a child? The action got bigger the further you leaned out, and the faster you spun. In this module, we will quantify our childhood remembering by measuring the rotation speed and the acceleration at different distances from the center of rotation. Within this module, you install the software to program the Nicla Sense ME on your computer and make your first steps in programming the device. You can follow the tasks as we did on a merry-go-round or simply turn around yourself with the sensor in your hand, comparing bent and extended arms.

Figure 4: Merry-go-round, illustration dating back nearly two centuries. [4]

The primary intention of this module is to get you started with the sensor platform. We provide a sample report documenting our merry-go-round ride to illustrate our expectations for the other modules. Please note that the tasks are shorter than in modules 1 to 3 – so is the sample report.

Inertial Forces

Forces that appear when the frame of reference is non-inertial are *inertial forces*. Because of their dependency on the frame of reference, the term "fictitious force" is common, although the forces may have very real consequences (think of your ride on a merry-go-round).

In this module, we focus on a rotating frame of reference. The rotation velocity Ω represents the change in angle per time, usually expressed in rad s^{-1}. In general, it is a vector $\boldsymbol{\Omega}$ with its direction perpendicular to the plane of rotation (upward in counter-clockwise rotation). On a mass m with a radial distance r from the center of rotation, a *centrifugal force*

$$\boldsymbol{F} = m\omega^2\,\boldsymbol{r} \tag{0.1}$$

pulls the mass radially outward (think of a stone at a string).

Another inertial force in a rotating frame of reference is the *Coriolis force*

$$\boldsymbol{F} = -2m\,(\boldsymbol{\Omega} \times \boldsymbol{v}) \tag{0.2}$$

occurring when a mass moves relative to the rotating frame of reference with the velocity $\boldsymbol{v}$. Please note the vectorial notion with bold letters for the vectors. The Coriolis force causes, e.g., the low-pressure systems in the atmosphere to spin counter-clockwise in the northern and clockwise in the southern hemisphere. An important sensor example based on the Coriolis force is the gyroscope. Also, some flow sensor makes use of this force.

Sensors

The BHI260AP is a smart sensor comprising an *inertial measurement unit* (IMU) and a *microcontroller unit* (MCU). The MCU is a proprietary core called Fuser2, dedicated to sensor fusion and self-learning *artificial intelligence* (AI). Figure 5 illustrates the role of the BHI260 as a sensor hub combining the data from the internal and optional external sensors. See Figure 2 to understand the BHI260's role as a sensor hub on the Nicla Sense ME board. The BHI260 runs firmware based on Bosch Sensortec's BSX Sensor Fusion library.

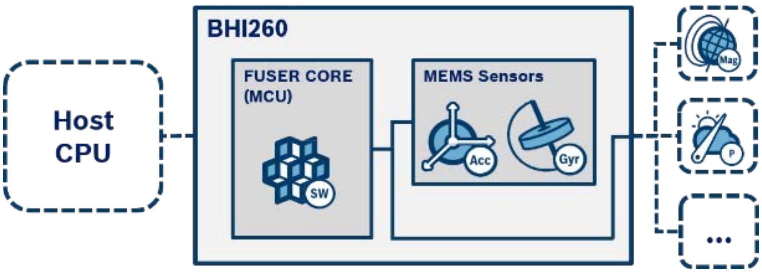


Figure 5: Block diagram of the BHI260 smart sensor with different sensors and a microcontroller core. [5]

The IMU consists of gyroscopes and accelerometers for all three axes – so, in total, a 6-axis IMU with six *degrees of freedom* (DoF). Other IMUs also comprise magnetic sensors for all three axes, which would be a 9-axis IMU. The combination of the different information from six or nine axes by a so-called *sensor fusion* algorithm provides spatial orientation. The BHI260 also allows sensor fusion with external sensors, such as the BMM150 magnetometer on the Nicla Sense ME board, resulting in a 9-axis arrangement. Beyond the spatial orientation, the self-learning AI allows, e.g., gesture or activity recognition. Within the lab course modules, we will not use such high-level features as the focus is on the sensor elements and principles, though you are encouraged to explore the systems AI features on your own.

Accelerometer

An accelerometer (acceleration sensor) measures the acceleration by observation of the force acting on a mass. Microfabricated accelerometers typically use the setup shown in Figure 6.

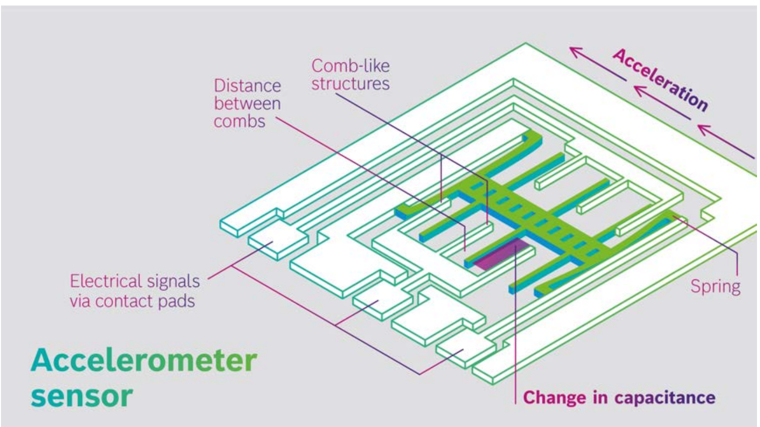


Figure 6: Typical microfabricated acceleration sensor with a moveable mass to detect the acceleration in one direction. Image: Bosch Sensortec.

Springs hold the movable mass (solid green), allowing movement along the longer axis of the mass. Comb-like structures allow the reading of the position of the mass. The mass movement causes the capacity between neighboring fingers to change (indicated by the purple plane). Practically, an electrostatic force generated from the same fingers helps to keep the mass steady by an electrical potential controlled according to the capacity values. The control signal used to keep the mass steady is the measured quantity resembling the acceleration. More details on the design and operation principle of microfabricated acceleration sensors will be provided in the lectures.

Gyroscope

The gyroscope (yaw-rate sensor) measures the rotation rate (yaw rate) by exploiting the Coriolis effect. Linear movement of a mass causes a Coriolis force perpendicular to the movement when the whole system rotates.

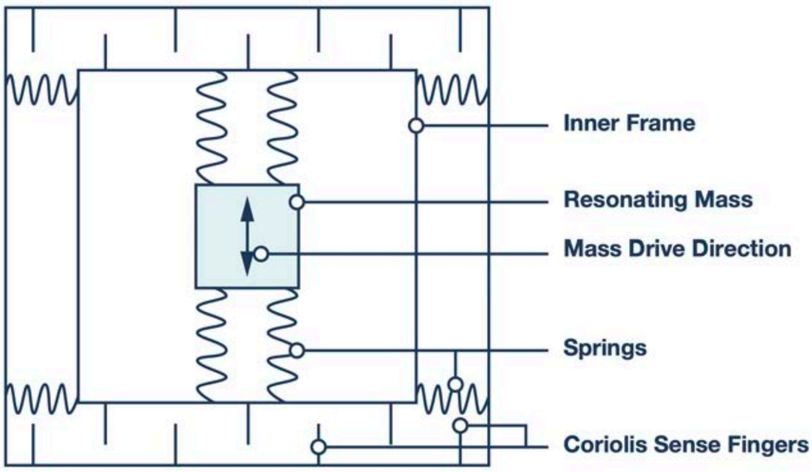


Figure 7: Typical microfabricated gyroscope (yaw-rate sensor) with an oscillating mass to detect the rotation. [6]

In a microfabricated realization of a gyroscope (Figure 7), the sensor mass, held by springs, oscillates in an inner frame (the velocity vector $\boldsymbol{v}$ is up and down in the drawing). This inner frame, held by springs, can move inside a fixed outer frame (left and right in the figure).

Rotation of the whole sensor (rotation rate vector $\boldsymbol{\Omega}$ into or out of the drawing plane) causes a Coriolis force according to the cross product in eq. 0.2. This force leads to a movement of the inner frame to the left or right in the drawing. The "Coriolis Sense Fingers" on the outer and inner frame form a capacitor. Measurement of the capacity provides the rotation rate.

Figure 8 illustrates the action of the Coriolis force for a rotation rate vector $\boldsymbol{\Omega}$ into the drawing plane. When the sensor mass moves up, the Coriolis force displaces the inner frame to the left. The displacement is onto the other side for the sensor mass moving down.

More details on the design and operation principle of microfabricated gyroscopes will be provided in the lectures.

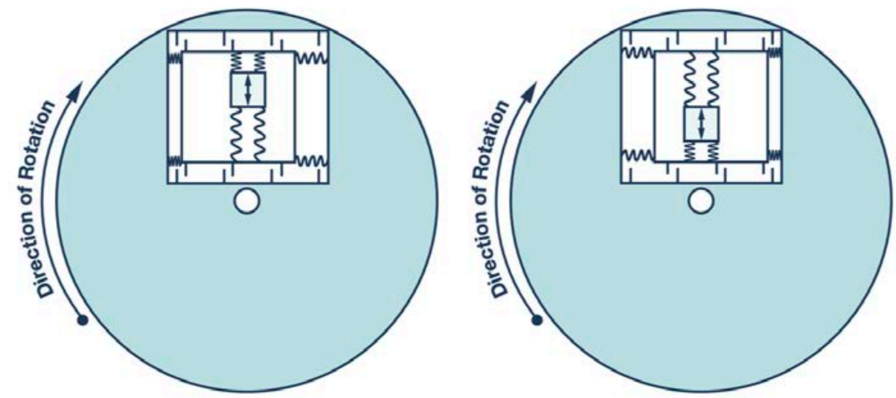


Figure 8: Illustration of the working principle for the microfabricated gyroscope from Figure 7. [6]

Sensors in the BHI260AP

The BHI260AP features acceleration sensors and gyroscopes for all three directions. Figure 9 shows the definition of the axes with the dot on the package as the reference. The positioning and orientation of the chip on the Nicla Sense ME board are clarified in Figure 10.

In the lab course, you should use the virtual sensors `PASSTHROUGH`, which provides the values from the physical driver directly:

```
SensorXYZ accel(SENSOR_ID_ACC_PASS);
SensorXYZ gyro(SENSOR_ID_GYRO_PASS);
```

The acceleration sensor's default configuration is the 8g range. The sensitivity is **4096 LSB/g** (least significant bit divided by free-fall acceleration), according to table 117 in [5]. To obtain the acceleration, you must divide the raw value by 4096 and multiply by 9.81 m s^{-2}.

For the rotation rate, the sensitivity is **16.4 LSB (° s^{-1})$^{-1}$** (table 119 in [5]). Make sure not to mess up ° s^{-1} and rad s^{-1}.

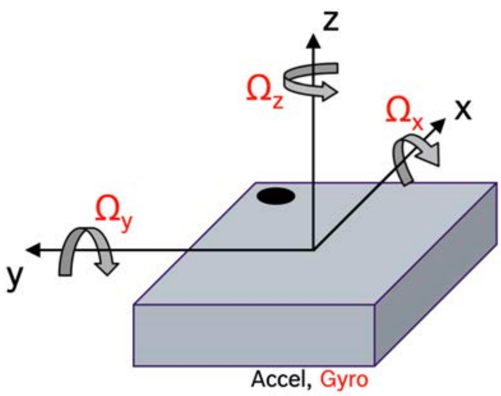


Figure 9: Orientation of the sensing axes with the dot on the BHI260AP package as the reference. [5]

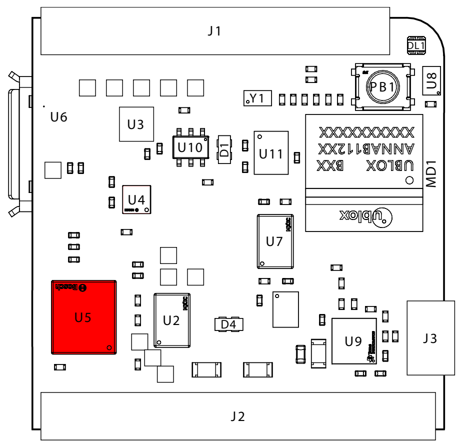


Figure 10: Top view of the Nicla Sense ME (USB port is on the bottom) with the BHI260AP marked in red. Adapted from [1].

Getting started

An excellent introduction and source of information beyond the first steps is the Arduino Nicla Sense ME Cheat Sheet[15]. Here you can follow the links to install the IDE, get the LED blinking and learn how to communicate with the sensor in standalone mode (the Nicla Sense ME board connected to a computer with a USB cable).

Always handle the Nicla Sense ME board with care. Strictly avoid contact with liquids! Although the assembled board is not very critical concerning electrostatic discharge (ESD), you should hold the board at its edges and not tap on the surface, reducing the risk of leakage currents due to contamination.

When you are familiar with microcontroller programming, the following steps are easy. If you are new, take your time, follow the steps and optionally browse further documentation on the Arduino programming environment.

1. Install the Arduino IDE and files for the Nicla Sense ME board as described in the Cheat Sheet or using the Quickstart Guide from the board's documentation[16]. On a modern computer, we recommend the Arduino IDE 2.0. If you have older hardware and face problems with the interface's performance, you could also use the Arduino IDE 1.8.
2. Try to get the LED blinking (section "RGB LED" in the Cheat Sheet). In contrast to other boards, you can even play around with color.
3. Follow the description in the Cheat Sheet's section on "Sensors" to get temperature readings available on the serial port of your USB connection. You need to install the libraries `Arduino_BHY2` and `ArduinoBLE`. Use the Arduino IDE's built-in serial console or any other serial terminal software to retrieve the data.

Once your setup is up and running, you are ready to start with the first tasks.

Tasks

Gyroscope

1. Write a program that measures every 100 ms the angular velocity in all three directions, and prints the values continuously on the serial interface.
   a. Ensure steady conditions for your sensor and record 1,000 gyroscope readings. Can you observe any drift? Repeat the measurements to ensure representative conditions.
   b. Calculate for each direction the mean value $\overline{\Omega}$. What is the offset in all three directions (sensor reading at rest)? Do all three directions behave the same? What could be a reason for the behavior?
   c. Show your data with subtracted mean value $\Omega - \overline{\Omega}$ for all three directions in an appropriate histogram. How are the values distributed? Do all three directions behave the same? What could be a reason for the behavior?

---

[15] https://docs.arduino.cc/tutorials/nicla-sense-me/cheat-sheet
[16] https://docs.arduino.cc/hardware/nicla-sense-me

Merry-go-round measurements

While the sample report contains data from a real merry-go-round, you, for the sake of onboarding, can hold the sensor (and preferably your computer) in your hand and turn around yourself with bent and extended arms. There is not even need for a complete turn.

2. Write a program that measures the rotation rate in the z-direction and the acceleration in the x- and y-direction every 100 ms and prints the values continuously on the serial interface. Additionally, include the time output since the controller start as a timestamp for each measurement.

    a. Place the Nicla Sense ME on a merry-go-round at the outer edge of the turning part (or use your extended arm). Position the x-direction towards the center. Measure the distance from the board (ideally the acceleration sensor) to the center of rotation.

    b. Spin the merry-go-round (or turn around yourself) while measuring and let the rotation decay.

    c. Plot the decaying rotation rate over time. What mathematical relationship does the decay follow? Use an appropriate fit function to describe the decay.

    d. Place the sensor board closer to the center (or use your bent arm). Position the x-direction towards the center and measure again the distance from the center of rotation.

    e. Spin the merry-go-round (or turn around yourself) while measuring.

    f. What were the maximal rotation rates in your rides from 2c and 2d? Consider also the noise in your evaluation.

    g. What were the acceleration values in radial and tangential direction at the moments of maximal rotation rates in your rides from 2c and 2d? Discuss your results and compare your values of the radial acceleration with the acceleration based on the centrifugal force (eq. 0.1).

Module 1: Acceleration and Pressure Sensors

Acceleration sensors and navigation applications go hand in hand, while pressure sensors are probably not your first thought. At least, if you are not an experienced mountaineer or paraglider pilot, for whom still today, altitude measurements based on barometric sensors complement the navigation with the *Global Positioning System* (GPS). In this module, you use a high-resolution pressure sensor to characterize an elevator ride and compare the data with readings from the acceleration sensor.

Barometric Formula

The atmospheric pressure depends on temperature and the weather conditions but also the altitude. For a period short enough to neglect the influence of weather change, the pressure changes at 250 m altitude approximately by 11.3 Pa m^{-1} (or 10 Pa m^{-1} if you prefer a rule of thumb).

You might remember a better description from your physics class, the barometric formula. For small changes in altitude, the influence of the temperature change with height is negligible (isothermal atmosphere), and the pressure p relates to the height above sea level h exponentially

$$p = p_0 \exp\left(-\frac{gMh}{RT}\right), \tag{1.1}$$

where p_0 is the reference pressure, and T is the absolute temperature. The constants are standard gravity $g = 9.81$ m s^{-2}, the molar mass of air $M = 0.02896$ kg mol^{-1}, and the universal gas constant $R = 8.314$ J mol^{-1} K^{-1}.

Sensors

This module uses both the accelerometer from the BHI260AP and the pressure sensor. See Module 0 for the description of the acceleration sensor.

Pressure Sensor

Microfabricated pressure sensors are based on a membrane deflection caused by a pressure difference between the two sides of the membrane. In a differential pressure sensor, both sides of the membrane are accessible from the outside, unlike an absolute pressure sensor (barometric pressure sensor) with the membrane sealing a cavity with a known reference pressure. The BMP390 is an absolute pressure sensor that matches the atmospheric pressure range even at high altitudes.

Figure 11 shows the typical setup of a microfabricated pressure sensor. A higher pressure above the silicon membrane causes a deflection which piezoresistive elements connected in a bridge circuit detect as a voltage change. A critical issue of such a semiconductor pressure sensor is its temperature dependency. That is why more accurate sensors include a temperature sensor for compensation with digital signal processing.

More details on the design and operation principle of microfabricated pressure sensors will be provided in the lectures.

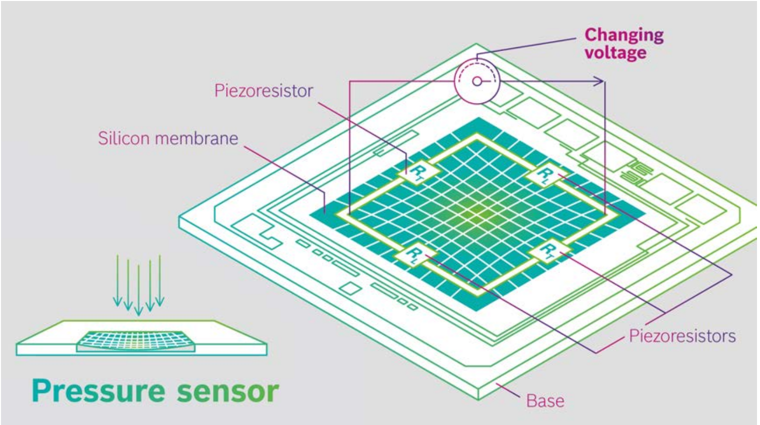


Figure 11: Typical microfabricated pressure sensor with piezoresistive elements measuring the deflection of a membrane. In an absolute pressure sensor, the membrane seals a cavity with the reference pressure. Image: Bosch Sensortec.

BMP390

The BMP390 combines a *micro electro mechanical system* (MEMS) sensor element with an *application-specific circuit* (ASIC). The ASIC features the analog front-end and a 24-bit *analog-digital converter* (ADC) together with logic for compensation purposes (see Figure 12 for the block diagram). The sensor features a relative accuracy of ±3 Pa, which translates into a relative accuracy better than ±0.3 m in a height measurement (see eq. 1.1).

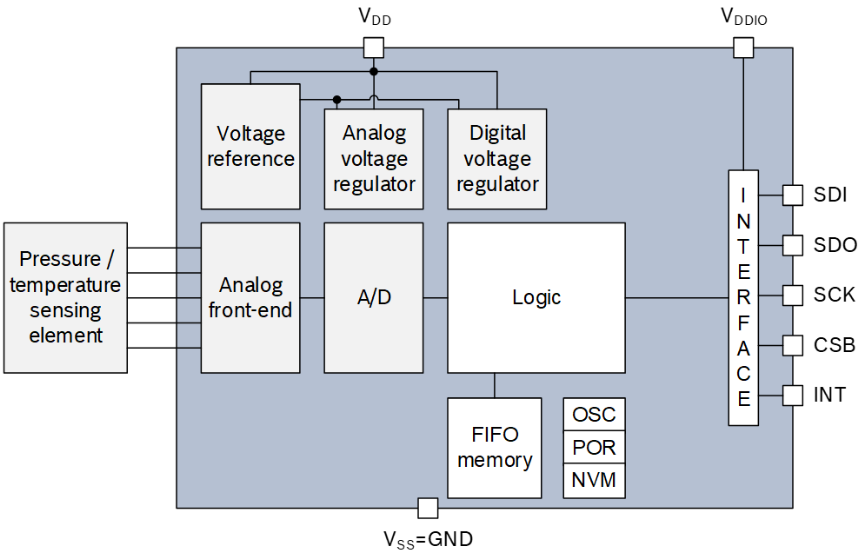


Figure 12: Block diagram of the BMP390. [7]

The metal case of the sensor has a tiny hole to allow the pressure exchange (Figure 13 left). Can you see it with bare eyes on your Nicla Sense ME board? The teardown of the BMP390 (Figure 13 right) shows two separate dies: the MEMS sensor element with the membrane placed on the ASIC and connected by five bond wires. With a closer look at the membrane, you can even guess the location of the piezoresistive elements.

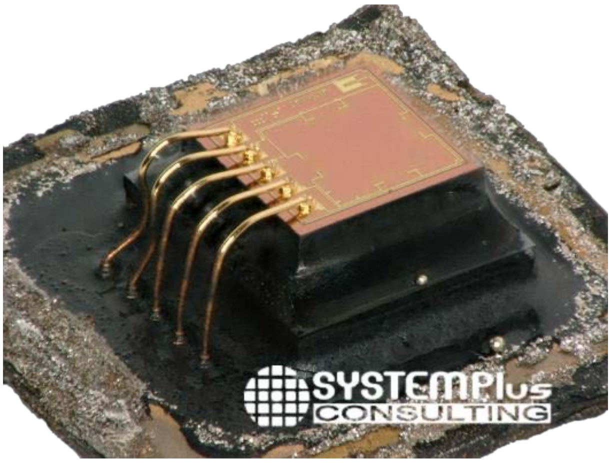


Figure 13: BMP390 in its housing with the tiny opening at the top for pressure exchange (left) [6] and the sensor with the cap removed (right) [8].

In the lab course, you should use the virtual sensors `SENSOR_ID_BARO`, which provides the values from in the BMP390:

```
SensorXYZ baro(SENSOR_ID_BARO);
```

The obtained values are the pressure in hPa (100 Pascal) to match the traditionally used unit millibar (mbar).

Tasks

Pressure Sensor

1. Write a program that measures every 100 ms the pressure and prints the values continuously on the serial interface.
   a. Ensure steady conditions for your sensor and record 1,000 pressure readings. Can you observe any drift? Repeat the measurements to ensure representative conditions.
   b. Convert the raw readings to pressure. Calculate the mean value $\bar{p}$. What is the offset (sensor reading at rest)? Is this value expected?
   c. Show your data with subtracted mean value $p - \bar{p}$ in an appropriate histogram. How are the values distributed? How do your results fit to the datasheet?

Acceleration Sensor

2. Write a program that measures every 100 ms the acceleration in all three directions and prints the values continuously on the serial interface.
   a. Ensure steady conditions for your sensor and record 1,000 acceleration readings. Can you observe any drift? Repeat the measurements to ensure representative conditions.
   b. Calculate for each direction the mean value $\bar{a}$. What is the offset in all three directions (sensor reading at rest)? Do all three directions behave the same? What could be a reason for the behavior?

c. Show your data with subtracted mean value $a - \bar{a}$ for all three directions in an appropriate histogram. How are the values distributed? Do all three directions behave the same? What could be a reason for the behavior?

Elevator Measurement

3. Write a program that measures the pressure and the acceleration in z-direction every 100 ms and prints the values continuously on the serial interface. Additionally, include the time output since the controller start as a timestamp for each measurement.

   a. Place your setup in an elevator (e.g., the elevator in lecture building 101). Make sure that your board is parallel to the floor and does not move during the ride. Start your measurement before the elevator moves. Ride the elevator from the ground floor to another floor and back. Compare different the different directions (upwards/downwards). Please ensure not to disturb other people.

   b. Evaluate the pressure data using the barometric formula (eq. 1.1) to describe the height of your rides in a diagram (height vs. time).

   c. Integrate the acceleration sensor data to show the velocity of your rides in a diagram (velocity vs. time).

   d. Integrate the acceleration sensor twice to show the height of your rides in a diagram (height vs. time) together with readings from the pressure sensor. Discuss which approach suits better if you want to track the height during your elevator rides.

Module 2: Magnetic Sensors

In contrast to the previous modules with acceleration, rotation, and pressure, we focus here on a quantity, the magnetic field intensity, which humans hardly can sense. While some animals can orient themselves according to the earth's magnetic field, we need technical help. For a long time, people have been using a pivoted magnet in the form of a needle to figure out the north-south axis. In the Chinese Han Dynasty, more than 2000 years ago, the compass was one of the Four Great Inventions. In Europe, the compass arrived in the 12th century and enhanced astronomic navigation. The measurement of the north direction became an essential tool for medieval seamen and arguable was one of the critical factors in European exploration and later, unfortunately also, lead to the colonialization of large parts of the world.

We will use the magnetic field sensor of the Nicla Sense ME board to characterize the earth's magnetic field (Figure 14) and evaluate how well we can find the magnetic north pole using the sensor's raw values. As we are all used to relying on the Global Positioning System (GPS) or even navigation based on GPS fused with cellular data on our smartphones, this module may inspire you to explore the traditional way.

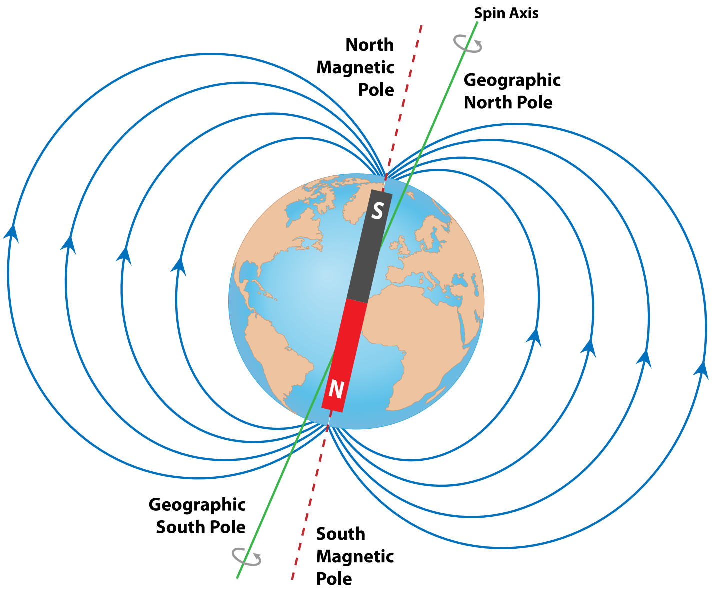


Figure 14: The earth's magnetic field forming the magnetic poles has an axis deviating slightly from the spin axis. The north magnetic pole (to which the north pole of a pivoted magnetic needle points) is a magnetic south pole.

Earth's Magnetic Field

Although we cannot sense it, we all depend on the earth's magnetic field vitally. Without its shielding properties, we would be exposed to solar winds (streams of charged particles emitted by the sun). Polar lights are an appealing side effect of this shielding effect. While one might think of the earth as a huge permanent magnet (as suggested by the illustration in Figure 14), a closer look reveals its impossibility. At a depth of 20 to 30 km already, the temperature is above iron's *Curie temperature* (the temperature at which a material loses its permanent magnetic properties). The commonly accepted *dynamo theory* assumes movement in the earth's liquid inner core, inducing currents in the solid outer core, causing the earth's magnetic field. The inner core's flow, including its direction, depends on many factors only loosely connected to the earth's rotation. Accordingly, the magnetic north-south axis deviates from the earth's rotation axis. Because of inhomogeneous flow conditions, regional anomalies exist.

In Germany, the *declination*, the deviation of the compass needle from magnetic north is only up to 3°. Other regions are more critical, having made medieval seamen sometimes struggle. That is, for example, why we are still not sure at which position Christopher Columbus landed in America. Additionally, ores in the outer core influence the magnetic field locally. For example, in northern Sweden, in Kiruna, there are places where the compass needle would point to the south.

Figure 15: Wandering of the north magnetic pole within the last centuries.

The exact position of the *north magnetic pole* (the point at which a compass needle would rotate freely) also changes with time. Figure 15 shows the trace since 1590. In this figure, you also find a point labeled the *north geomagnetic pole*. This point refers to an ideal dipole model fitting the Earth's magnetic field. More advanced models account for regional anomalies too. The International Geomagnetic Reference Field (IGRF) and the World Magnetic Model (WMM) are two commonly used magnetic earth models[17] to describe the field at a specific location and date. For reference data needed in the lab course, both models are fine.

Quantitative Description

The earth's magnetic field is described at specific locations by the horizontal, vertical, and total field intensity. The different parameters relate to sensor readings by simple vector calculations:

- Horizontal magnetic field intensity $B_{\text{hor}} = \sqrt{B_x^2 + B_y^2}$
- Vertical magnetic field intensity $B_{\text{vert}} = B_z$
- Total magnetic field intensity $B = \sqrt{B_x^2 + B_y^2 + B_z^2}$.

The coordinate system assumes the z-axis to be perpendicular to the ground.

---

[17] You can use an online tool (e.g., https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml) or a mobile phone app to get the values for your location.

Sensors

Hall Sensor

The most widely known magnetometer is the Hall sensor. The principle exploits the *Hall effect*, the formation of a voltage perpendicular to a current flow when charge carriers get deflected by a magnetic field. The conductor in a Hall sensor typically has the geometry of a thin plate, either with longer dimensions along the direction of current flow or, more commonly nowadays, a quadratic footprint.

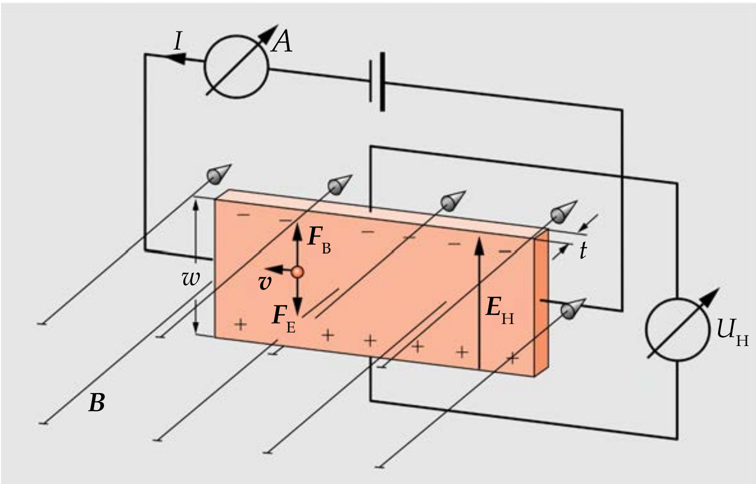


Figure 16: Hall effect: the magnetic field (intensity vector $\boldsymbol{B}$) causes deflection of charge carriers leading to a Hall voltage U_H across the width of the conductor. Adapted from [9].

Figure 16 illustrates the principle of the Hall effect. A magnetic field (intensity vector $\boldsymbol{B}$) points orthogonally to a conductor where charge carriers (current I, velocity vector $\boldsymbol{v}$) get deflected. An electric field (the *Hall field* with intensity vector $\boldsymbol{E}_{\mathrm{H}}$) establishes, causing an electric force ($\boldsymbol{F}_{\mathrm{E}}$) balancing the *Lorentz force* ($\boldsymbol{F}_{\mathrm{L}}$):

$$e\boldsymbol{E}_{\mathrm{H}} = -e\,\boldsymbol{v} \times \boldsymbol{B} \tag{2.1}$$

e is the elementary charge. The Hall voltage across the width w of the conductor relates to the charge carrier's velocity and the orthogonal component of the magnetic field intensity:

$$U_H = -wvB \tag{2.2}$$

Expressing the Hall voltage in terms of current I using the charge carrier density n makes clear why thin Hall plates (thickness t) are advantageous:

$$U_H = -\frac{IB}{ent} \tag{2.3}$$

More details on the Hall sensors and their technological realization in silicon will be provided in the lectures.

FlipCore Element

The FlipCore technology is a proprietary technology by Bosch Sensortec. The principle relates to a *fluxgate sensor* in which a sender coil drives a core between its saturation magnetization, and a receiver coil wound on the same core reads the induced signal. An external magnetic field (the field intensity to be measured) superimposes and causes a signal detected as 2nd harmonic. More details on the fluxgate sensor will be provided in the lectures.

The FlipCore element simplifies the arrangement using a thin-film core of an iron-nickel alloy, typically a stack of few separated layers with a thickness of a few ten nm only, containing a single magnetic domain each. Upon periodic excitation, the magnetization flips between its saturations. An external magnetic field superimposing the periodical excitation causes a delay at the receiving coil, allowing for much simpler time measurement in comparison to the detection principle in the fluxgate sensor. Compared to a Hall sensor, the FlipCore principle features reduced power consumption and lower noise.

BMM150

The BMM150 contains two FlipCore elements for the in-plane magnetic field components and one Hall sensor for the vertical component. The Flip Core elements and the Hall sensor have their own circuits, which all use the same ADC (Figure 17).

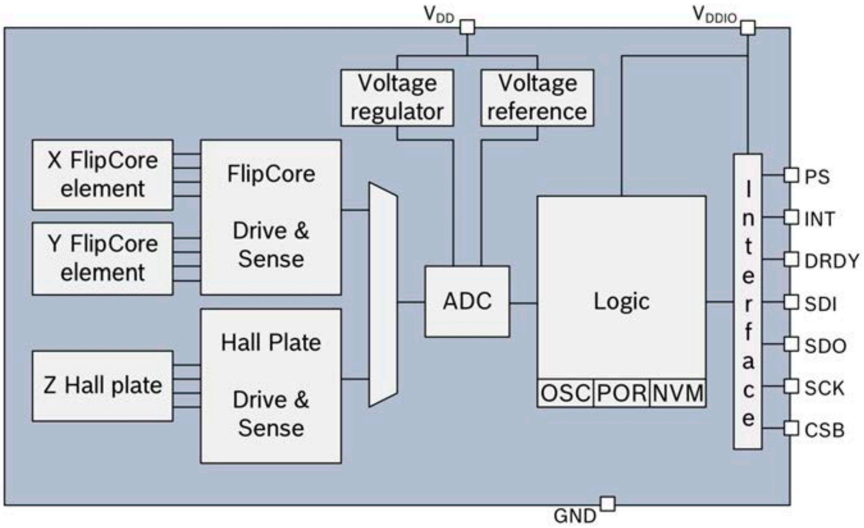


Figure 17: Block diagram of the BMM150. [10]

The inner life of the comparable BMC050 reveals the two separate FlipCore elements for the in-plane components integrated next to the stack of dies with the Hall plate below the main circuit (Figure 18).

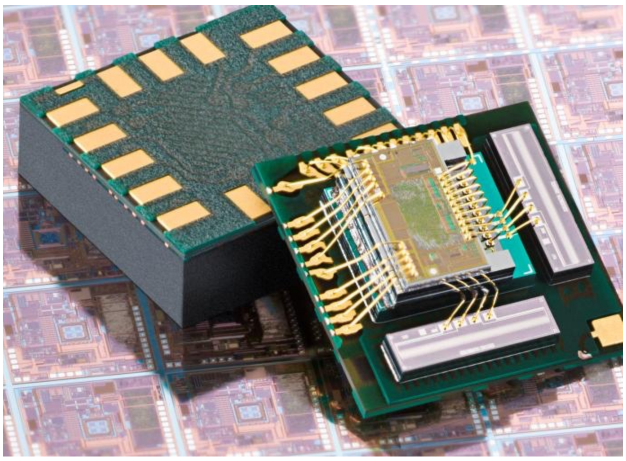


Figure 18: View into the BMC050, a comparable magnetic sensor from Bosch. [11]

While we focus in the lab course on unprocessed values from the sensor, the designer implementing a sensor in an application would prefer compensated data taking advantage of the manufacturer's knowledge of the device details. A typical application requiring sophisticated compensation is a magnetometer installed in a mobile phone. Here, sensor-specific properties play a role, but also the environment, like the field from the magnet of a loudspeaker with orders of magnitude higher strength than the magnetic field of interest, namely the earth magnetic field, on which it is superposed. Most critically, changing environmental conditions can be challenging to handle. For example, a temperature change of just 10 K could cause a difference in the speaker magnet's field intensity in the order of the entire earth's magnetic field intensity. Thus, sophisticated algorithms aim to distinguish desired effects from the environment's contribution.

In the lab course, you should use the virtual sensor `PASSTHROUGH`, which provides the values from the physical driver directly:

```
SensorXYZ magn(SENSOR_ID_MAG_PASS);
```

The sensitivity of the returned magnetic field intensity is **16 LSB μT^{-1}** [10].

Figure 19 shows the definition of the axes with the dot on the package as the reference. The positioning of the chip on the Nicla Sense ME board gets clear in Figure 20.

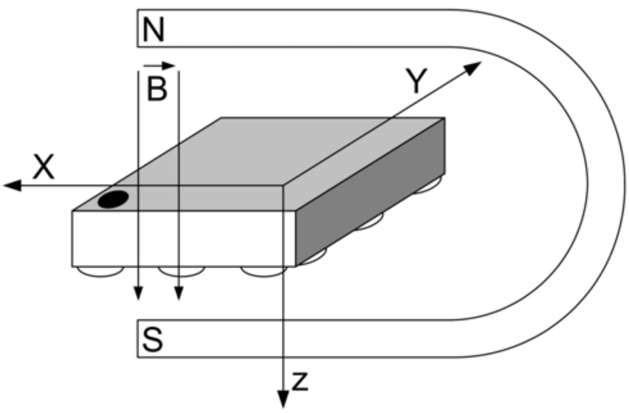


Figure 19: Orientation of the sensing axes with the dot on the BMM150 package as reference. [10]

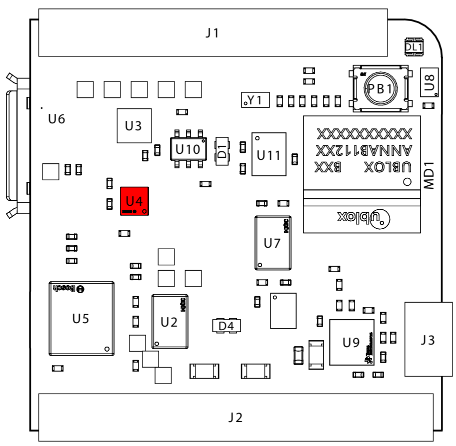


Figure 20: Top view of the Nicla Sense ME (USB port is on bottom) with the BMM150 marked in red. Adapted from [1].

Tasks

Magnetic Sensor

1. Write a program that measures the magnetic field intensity in all three directions (B_x, B_y, B_z) every 100 ms and prints the value on the serial interface.
    a. Ensure steady conditions for your sensor and record 1,000 magnetic field readings. Can you observe any drift? Repeat the measurements to ensure representative conditions.
    b. Calculate for each direction the mean value $\bar{B}$ and show your data with subtracted mean value $B - \bar{B}$ in an appropriate histogram. How are the values distributed?

Are all three directions behaving the same? Do you observe any difference between the FlipCore elements and the Hall sensor?

2. Evaluate the offset of all three axes:
    a. Ensure steady conditions for your sensor and record 100 magnetic field readings. Take the mean value in each direction.
    b. Flip your board 180° around the x and y axes and repeat the measurement for offset correction. Ensure that there is no change in positioning of the board or any external factors affecting the magnetic field. Take the mean value in each direction.
    c. Compare your results from 2a and 2b for each direction. What is the sensor's offset? Does this match the range specified in the datasheet? Are all three directions behaving the same? Do you observe any difference between the FlipCore elements and the Hall sensor?

Earth's Magnetic Field

3. Use your program from the first task to measure the earth's magnetic field at your place. Consider in your evaluation the offsets obtained in task 2.
    a. Find a place outdoor where you can perform an undisturbed measurement. Avoid metals that might influence the magnetic field locally. For each direction, use the mean value of the acquired data and report it together with the standard deviation (e.g., $(42 \pm 3)\ \mu\mathrm{T}$).
4. Evaluate the horizontal magnetic field intensity B_{hor}, the vertical component B_{vert}, and the total intensity B.
    a. Compare your findings with the expected value according to a geomagnetic model for your location (longitude, latitude) and date.
    b. Look up the specifications of the BMM150 and discuss your results regarding offset, resolution, and accuracy.
5. Place your sensor board aligned with a building or straight street for which you know the north direction. You could measure the north direction from a satellite image, e.g., using Google Earth Pro[18] or similar software. On the campus, you find a north-south street (from the corner behind the stone garden next to building 101, along buildings 051 and 052, towards building 105). Avoid metals that might influence the magnetic field locally.
    a. Measure B_x and B_y. Use trigonometric calculation to obtain the angle with respect to the north direction.
    b. How good does your measured direction match the expected north direction? Discuss your results. Are your results accurate enough to relate possible deviations to the declination?

[18] https://www.google.com/earth/about/versions/#download-pro

Module 3: Gas Sensors

The description for Module 3 will be provided during the semester.

Recommended Literature

- Arduino Nicla Sense ME Cheat Sheet: https://docs.arduino.cc/tutorials/nicla-sense-me/cheat-sheet

Depending on your preferred language:

- Meschede, Gerthsen: „Gerthsen Physik", 2015, Springer Spektrum (Campus-Lizenz: http://www.redi-bw.de/start/unifr/EBooks-springer/10.1007/978-3-662-45977-5)
- Halliday, Resnick, Walker: "Fundamentals of Physics", 2010, Wiley

References

[1] Arduino Nicla Sense ME Product Reference Manual, ABX00050, 03-Aug-2022. https://docs.arduino.cc/resources/datasheets/ABX00050-datasheet.pdf

[2] R. F. Keeling, C. D. Keeling, 2017, Atmospheric Monthly In Situ CO_2 Data - Mauna Loa Observatory, Hawaii (Archive 2022-06-01). In Scripps CO_2 Program Data. UC San Diego Library Digital Collections. https://doi.org/10.6075/J08W3BHW

[3] D. Lüthi, M. Le Floch, B. Bereiter, T. Blunier, J.-M. Barnola, U. Siegenthaler, D. Raynaud, J. Jouzel, H. Fischer, K. Kawamura, and T.F. Stocker, "High-resolution carbon dioxide concentration record 650,000-800,000 years before present", *Nature*, vol. 453, pp. 379-382, 2008. https://doi.org/10.1038/nature06949

[4] "Das Carussel" in "Kinderspiele in 24 illum. Kupfern: Fibel und Rechenbuch für kleine Kinder". Nürnberg, 1828. Available: https://pictura.bbf.dipf.de/viewer/image/30733_89350651-53ec-4421-a21b-1561384935de/1/ [Accessed: 12-Aug-2022].

[5] BHI260AP Datasheet, Bosch Sensortec, BST-BHI260AP-DS000-02, rev. 1.1, 15-Apr-2021. https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bhi260ap-ds000.pdf

[6] J. Wattson, 2016, "MEMS Gyroscope Provides Precision Inertial Sensing in Harsh, High Temperature Environments", Analog Devices. https://www.analog.com/media/en/technical-documentation/tech-articles/MEMS-Gyroscope-Provides-Precision-Inertial-Sensing-in-Harsh-High-Temps.pdf [Accessed: 14-Oct-2022].

[7] BMP390 Datasheet, Bosch Sensortec, BST-BMP390-DS002-07, rev. 1.7, Mar-2021. https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp390-ds002.pdf

[8] https://s3.i-micronews.com/uploads/2017/11/Yole_Bosch_Sensortec_BMP380_barometric_pressure_sensor_flyer_SP17358.pdf [Accessed: 18-Aug-2022].

[9] Meschede, Gerthsen: „Gerthsen Physik", 2015, Springer Spektrum

[10] BMM150 Datasheet, Bosch Sensortec, BST-BMM150-DS001-05, rev. 1.4, Apr-2020. https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmm150-ds001.pdf

[11] S. Finkbeiner, "MEMS for automotive and consumer electronics," *Proceedings of the ESSCIRC*, 2013, pp. 9-14, https://10.1109/ESSCIRC.2013.6649059