

# Practicum 3: Sensors and Communication

## General Instructions

- Use the template (provided in Moodle) to create new projects for each task.
- Ensure that all **calculations** along with **formulas** and their **relevant parameter** necessary for the tasks are documented within the report. Make sure to list all the **values** you used for the parameters. Provide an explanation if **customized parameters** and values are chosen.
- Code has to be **clearly structured** and follow **common coding guidelines**:
  - ✓ Use **meaningful variable names** that reflect their purpose or content. Avoid single-letter variable names except for loop counters (e.g., i, j, k).
  - ✓ **Add comments** to explain complex logic, algorithms, or any part of the code that might not be immediately clear to others (or yourself in the future).
  - ✓ Use **constants** (#define or const) to define constants instead of hardcoding values. This improves code readability and makes it easier to update values later.
  - ✓ Break your code into **smaller, logical functions** to improve readability, maintainability, and reusability. Each function should ideally perform a single, well-defined task.
  - ✓ **Avoid using "magic numbers"** (hardcoded numeric constants) in your code. Instead, define them as named constants with descriptive names.
- Please ensure to **submit a report** as part of this exercise. Kindly adhere to the **guidelines** outlined in the provided **template** when creating the report.
- If you encounter any issues regarding the tasks, please consult the **FAQ section** in **Moodle**. If you cannot find the answer you're looking for, feel free to post your queries in the **forum** or contact us directly via email.

### Useful tips:

- For this exercise, you can use any of the following serial monitors: putty serial client, Arduino serial monitor, HyperTerminal, serial monitor extension from VS Code.
- Check the application notes as well as the data sheet for necessary register values and formulas.

For a detailed description of these functions, please refer to the function definitions in the HAL library documentation ([https://www.st.com/resource/en/user\\_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf)).

Source code for the library is located inside the Drives folder  
(Project/Drivers/STM32H7xx\_HAL\_Driver/Src/\*.c)

## Task A: Read Data from the LIS3MDL Magnetic Sensor

In this task, you are required to extend the program from the I<sup>2</sup>C tutorial and read the X and Y-axis magnetic data from the LIS3MDL magnetic sensor. The addresses of the respective registers can be found in the datasheet and application note. After reading the three-axis magnetic data, it must be displayed on the serial monitor of your choice. To do this, you are required to use the UART4 interface

of the STM32 MCU with the configuration specified in the UART tutorial. The read values must be displayed in the following format:

**X: xxxx Gauss, Y: yyyy Gauss, Z: zzzz Gauss**

### **Requirements:**

- ✓ Use the **I2C2 interface** to read data from the digital sensor
- ✓ Use the **UART4 interface** to send magnetic values to the serial monitor
- ✓ Follow the display format mentioned above
- ✓ Implement minimal error handling

## **Task B: Read Humidity Values from the HTS221 Sensor**

In this task, you are required to create an application that reads the raw humidity data from the HTS221 sensor's output registers using the I2C2 interface. The raw data from the register must be converted to obtain relative humidity values according to the guidelines provided in the datasheet and application note. The relative humidity should be displayed on the serial monitor of your choice. To do this, you are required to use the UART4 interface of the STM32 MCU with the configuration specified in the UART tutorial. Each value must be displayed in a new line on the serial monitor. The device and required register addresses can be found in the datasheet and application note.

### **Requirements:**

- ✓ Use the **I2C2 interface** to read data from the digital sensor
- ✓ Use the **UART4 interface** to send humidity values to the serial monitor
- ✓ Set **CTRL\_REG1 (20h)** to **0x81** (see description in the datasheet)
- ✓ Implement minimal error handling

## **Task C: Read Temperature Values from the HTS221 Sensor**

In this task, you are required to create an application that reads the raw temperature data from HTS221 output registers using the I2C2 interface. The raw data must be converted using linear interpolation to obtain the temperature values. The device and register addresses can be found in the datasheet and the application note. After obtaining temperature values, each value must be displayed on the serial monitor of your choice. To do this, you are required to use the UART4 interface of the STM32 MCU with the configuration specified in the UART tutorial. Each value must be displayed in a new line on the serial monitor.

### **Requirements:**

- ✓ Use the **I2C2 interface** to read data from the digital sensor
- ✓ Use the **UART4 interface** to send temperature values to the serial monitor
- ✓ Set **CTRL\_REG1 (20h)** to **0x81** (see description in the datasheet)
- ✓ Implement minimal error handling

## Task D: Read Temperature & Pressure Values from the LPS22HH Sensor

In this task, you are required to create an application that reads the raw pressure data and the internal temperature of the sensor chip using the I2C2 interface. The raw data must be converted to obtain the pressure and temperature values according to the instructions provided in the datasheet and the application note. The device and register addresses can be found in the datasheet. After converting raw data, the values must be displayed on the serial monitor of your choice. Each newly obtained value must be displayed in a new line on the serial monitor and should be displayed in the following format:

**Pressure: xxxx hPa, Temperature: yyyy °C**

### Requirements:

- ✓ Use the **I2C2 interface** to read data from the digital sensor
- ✓ Use the **UART4 interface** to send temperature and pressure values to the serial monitor
- ✓ Follow the display format mentioned above
- ✓ Implement minimal error handling