Lab 2: Onboard Communication

April 25th, 2022

Embedded devices typically employ dedicated sensors and actuators that require onboard communication with the connected microcontroller. You have already used the UART protocol (indirectly via `printf()`), which is one method to let the microcontroller communicate with other devices (in this case your computer). This lab focuses on the very common I^2C and SPI protocols, which we will use to set up a data logger for temperature and pressure values.

General Hints:

- As the programs become more complex over the labs, we strongly encourage you to structure your programs in classes / libraries. To fresh up on C++ programming, please find this compact guide on Writing-a-Library or refer to tutorial sites such as w3schools.
- The microcontroller allows you to stop and debug programs during execution (hit the *Debug Program* button right next to the *Run program* button). See also Debugging with Mbed Studio.

Task 2.1 : I^2C Sensor

Your first task is to establish an I^2C communication with the BMP280 digital air pressure sensor to read out pressure and temperature values stored in the sensor's data registers. Use the I2C API, the sensor's datasheet and the additional hints provided in GitLab to solve this task! The program to be created should contain the following steps:

1. Wake up the sensor from sleep mode.
2. Get the calibration data.
3. Get the raw sensor data.
4. Calculate the physical values using the calibration data and the provided compensation functions.
5. Display the temperature in degrees Celsius (degC) and the pressure in hectopascal (hPa) in the console.

Task 2.2 : Data Logger

The Lora board is connected to an SD card module, which can be accessed by the microcontroller via the SPI interface. To establish data logging in text file format on the SD card, we will use the SDBlockDevice and FATFileSystem APIs provided in Mbed OS. First, run the example provided in GitLab, which formats the SD card and creates a test file. Verify that the example worked properly by reading the SD card with your PC. Then write a program to fulfill the following tasks:

- Write the pressure and temperature values obtained in Task 2.1 with the elapsed time as comma separated values line-by-line to a file on the SD card. The log file should also contain the header: `Time (s), Pressure (hPa), Temperature (degC)`
- The measurements should be logged at a constant sampling rate, e.g., 1 Hz.
- A new log file should be created each time the board is powered up or reset. Always write to the newest file and make sure not to overwrite/erase existing files (bouncing)!
- When the blue user button is pressed, all log files should be deleted.

Hint: You may use the Time and InterruptIn APIs provided in Mbed OS to solve this task.

Task 2.3 : UART (Exam Bonus)

For the bonus task, the previous program(s) shall be modified/extended by using the UART protocol and implementing the following features:

- On program start, the program should wait for a (manual) user input via UART to set the current date and time of the real-time clock (RTC). The datetime should be input in one line in the following format `YYYY-MM-DD_hh-mm-ss` with year (YYYY), month (MM), day (DD), hour (hh), minute (mm), and second (ss).
- To automate the time setting and make it also more accurate, write a program in Python or MATLAB that reads the current datetime from the PC and sends it to the microcontroller.
- When recording the measurement data, write, instead of the elapsed time (in sec), the timestamp using a `hh:mm:ss.SS` (hours:minutes:seconds.milliseconds) format. When a new log file is created, the file name should contain the datetime of creation, e.g. `2022-04-14_14-40-15.log`.
- In addition to the user button, a new log file should be created when the command `"new log"` is sent via console / UART as a string. Also, all log files should be erased when the string `"erase all"` is sent.

Submission (Deadline: May, 6th 2022, 11:00 am)