Practicum 2: Timers

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

Source code for the HAL library is located inside the Drivers folder (Project/Drivers/STM32H7xx_HAL_Driver/Src/*.c)

Task A: Blink an LED with Timer Module

In this task, you are required to toggle the LED on the EliteBoard periodically using a timer. The clock frequency of the timer module should be configured according to the requirements given below.

Requirements:

- ✓ Use **PJ7 (SEGDP)** as digital output (LED)
- ✓ **Timer 2** module should be used for blinking the LED
- ✓ T_{out} should be **500 ms.**

Task B: Vary the Brightness of an LED

The user should be able to control the brightness of the LED by using two push buttons provided on the board. These buttons are connected to the pins PJ12 and PJ13 and labelled as BTN1 and BTN2 respectively. BTN1 button should be used to increase the brightness of the LED and BTN2 is used to decrease the brightness. The range of brightness should be between 0% and 100% and an individual step should be 10%. That means, increasing the brightness by a single step indicates a 10% increase in brightness and vice versa. The brightness should be controlled by adjusting the PWM signal which can be generated using a timer module. The configuration of the timer module and the PWM signal should be done according to the below requirements.

Requirements:

- ✓ Use **PJ7 (SEGDP)** as digital output
- ✓ Use **PJ12 (BTN1)** and **PJ13 (BTN2)** for digital inputs
- ✓ **Frequency** of the PWM signal should be **500 Hz**
- ✓ Use **channel 2** of **TIM8** for generating PWM signal (**PJ6**)
- ✓ Max brightness = **100%**, Min brightness = **0%**, Step size = **10%**

Task C: Count Pulses of a Blinking LED

In this task, you are required to generate a pulse counter with Timer 3. A clock (Tout = 1 sec approx.) signal shall be generated from Timer 2, which shall be used as an internal trigger for Timer 3. Timer 3 shall count repeatedly from 0 to 9. The current count of the Timer 3 should be displayed on the Seven Segment Display of the Eliteboard.

Requirements:

- ✓ Use **the 7-segment display** for digital output
- ✓ **Timer 2** should provide an internal trigger**; T_{out}** should be **1 s.**
- ✓ **Timer 3** should be used to count (**0-9**) pulses