

DESN2000 (Computer Engineering) 2026 T2

Lab sheet 3 (weeks 5 and 7)

Last edited: 13/06/2026

Make reasonable assumptions if not explicitly stated and state such assumptions to your demonstrator when getting marked.

Task 1 (15%)

Implement a milliseconds counter (stopwatch) using hardware timers and timer interrupts. Display the milliseconds counter on the LCD. The stopwatch starts/stops when the blue button on the NUCLEO board is pressed. The stopwatch should be reset when SW1 is pressed. For buttons, you should use external interrupts too.

Task 2 (15%)

Write a programme that blinks all LEDs on the board (D1-D20 on coaST and the LD2 on NUCLEO) at 1-second intervals (1 second on, then 1 second off and so on). You must use hardware timer interrupts for this task. When button SW1 is pressed, the blinking frequency should halve. When SW2 is pressed, the blinking frequency should double. When you press the button multiple times, it should halve/double the current rate each time. For buttons, you should use external interrupts.

Task 3 (20%)

Write a programme that uses hardware timer interrupts and external interrupts to measure the time between any consecutive button presses. You should use SW1, SW2 and SW3 as the buttons. The time should be measured between a button release and the next press. The consecutive button presses can be from different buttons. The measured time should be displayed on the LCD. You can assume the maximum time between two presses is one hour. Note that you should repeatedly do the measurement and update the LCD for an indefinite number of button presses.

Task 4 (30%)

Write a programme that implements a simple two-way chat between the development board and a user on a PC, communicating through the UART. When the PC user sends a message, it is displayed on the top line of the LCD. As the board user composes a message on the keypad it should also be shown on the bottom line of the LCD and remain there when sent until the board user begins typing a new message. In the UART terminal, every message sent should appear with an indicator of who sent it (e.g. "User 1:" for the PC user and "User 2:" for the board user), so the terminal shows the full conversation transcript.

The keypad on the board has the keys 0–9, A, B, C, D, * and #. Since the keypad does not have a key for every letter, you must decide how to map the available keys to characters so the user can compose a message. You can assume the maximum length of a message is 16 characters, containing only spaces and the letters A–Z (you need only implement either upper or lower case). How you map the keys is your choice — for example, pressing a digit key multiple times to cycle through its letters (like an old mobile phone), or using the A, B, C and D keys to select a letter within a group, or any other scheme you can justify. State your chosen mapping and assumptions to your demonstrator. Reserve at least one key for sending the message. You must also handle a backspace, and what happens when a message reaches 16 characters.

You do not need to display the other party's message until it is sent, but the LCD's bottom line must show the keypad message as it is being written. The top line shows only the most recently received message from the PC.

See the example below (PC user = User 1, board user = User 2):

1. PC user types "hello" and enters message

LCD:

Top Row	hello
Bottom Row	

Terminal:

User 1: hello

2. Board user replies "hi" from the keypad:

Top Row	hello
Bottom Row	hi

Terminal:

User 1: hello

User 2: hi

3. Board user types “how are u” and enters message

Top Row	hello
Bottom Row	how are u

Terminal:

User 1: hello

User 2: hi

User 2: how are u

4. PC user types “i am good” and enters message

Top Row	i am good
Bottom Row	how are u

Terminal:

User 1: hello

User 2: hi

User 2: how are u

User 1: i am good

Task 5 (20%)

Without using HAL functions, write a programme that increments a counter when SW1 is pressed and decrements when SW2 is pressed. The counter is only 4-bit and should be displayed on LEDs D4-D1. For this task, you are NOT allowed to use HAL functions. Instead, you should be directly accessing relevant memory addresses in C. What you have to write in C directly would be the following:

- Everything inside the *MX_GPIO_Init(void)*
- Infinite while loop inside the main function

Hints: You can first write a programme using HAL, and then replace each HAL function with direct C code, one by one. This way, it would be easier to debug.