DESN2000 (Computer Engineering) 2025 T2

Lab sheet 3 (weeks 5 and 7)

Last edited: 24/06/2025

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

Task 1 (20%)

Implement a programme that doubles a 16-bit signed integer counter when the SW1 button is pressed, subtracts 5 when the SW2 button is pressed, sets to 1 when the SW3 button is pressed and sets to the largest possible value when the SW4 button is pressed. The counter must be displayed on the LCD. For this task, you MUST use external interrupts for the buttons whenever possible.

Task 2 (20%)

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 3 (20%)

Write a program that blinks all LEDs on the board (D1-D20 on coaST and the LD2 on NUCLEO) at 0.5-second intervals (0.5 seconds on, then 0.5 seconds 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 4 (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 5 (20%)

Write an ARM assembly function that delays the number of milliseconds specified as an argument using busy waiting. Note that the delay should be very accurate, which means you should consider the clock frequency of the processor and the number of instructions in your function. Once the assembly delay function is implemented, call this function from a C programme (which you may use HAL) that takes a user input from the keypad (an integer) and waits for the requested number of seconds after the '#' key on the keypad is pressed. As soon as the timeout is reached, indicate it by lighting LEDs D5-D20 on coaST board. The user should be able to clear the LEDs and start another round when the SW1 button is pressed.