DESN2000 (Computer Engineering) 2024 T2

Lab sheet 3 (weeks 5 and 7)

Last edited: 19/06/2024

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

Task 1 (20%)

Implement a programme that increments a 32-bit signed integer counter when SW1 button is pressed, decrements when SW2 is pressed, resets to 0 when SW3 is pressed and sets to the largest possible value when SW4 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 should be reset to 0 when the key '0' on the keypad is pressed and it should start when the blue button on the NUCLEO board is pressed.

Task 3 (20%)

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

Task 4 (20%)

Write a programme that uses hardware timer interrupts and external interrupts to measure the time between two consecutive button presses. You can use SW1 as the button. The time should be measured between the button release and the next press. 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 seconds 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 time out is reached, indicate it by lighting all LEDs (D1-D20 on coaST and the LD2 on NUCLEO) on the boards. The user should be able to clear the LEDs and start another round when the blue button on the NUCLEO board is pressed.