DESN2000 (Computer Engineering) 2024 T2
Lab sheet 4 (weeks 8 and 9)

Last edited: 15/07/2024

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

Task 1 (20%)

Write a programme that slowly increases the brightness of LEDs D1-D4 on the coast board and LD2 on the NUCELO board. Once 100% brightness is reached, slowly decrease the brightness back to 0% and repeat the process. You must use hardware PWM if the corresponding microcontroller pins have such hardware PWM support. For pins without such hardware PWM, you should implement the PWM signal using software.

Task 2 (20%)

Write a programme that takes a frequency value (integer) from the user through the keypad and then generates a sound wave of the requested frequency from the buzzer. The user input must be interactively displayed on the LCD. The '*' key should be used as a backspace. Pressing the '#' button should start generating the sound and will stop taking any keypresses. Pressing any of the SW1-SW4 buttons should stop the sound and take a new frequency input. Note that the sound that you generate should be in form of short beeps with sufficient silence between the delays, so that it is not that unpleasant (please do not do a continuous beep).

Task 3 (20%)

Write a program that lights up all LEDs on the board (D1-D20 on coaST and the LD2 on NUCLEO) whenever you cover any of the two LDRs on the coaST board. As long as an LDR is covered, the lights should be up.

Task 4 (20%)

Write a programme that illuminates the LEDs on the board (you can limit yourself to pins that support hardware PWM), based on the level of light intensity as detected by one of the LDRs of your choice. The level of brightness of the LEDs should be proportional to the level of darkness.

Task 5 (20%)

Write a programme that takes commands in the form of the following example from a user through the UART, executes them and sends the result back to the user through UART.

```
get LDRR24 5 3
```

The second argument can be LDRR24 or LDRR32. The third argument (value 5 in the example) is the number of readings required to take. The fourth argument (value 3 in the example) is the number of seconds between consecutive readings. This example requests that 5 readings be taken from the LDRR24 with 3-second intervals.

Based on the command provided by the user, the programme should take readings from the requested LDR and send the results back. You should start the relevant ADC only when a command has been issued. That is, you shouldn't be simply reading the ADC in a while loop.

Assume that the argument 3 and 4 are integers and the arguments are separated by spaces. Assume that the user doesn't make any errors in input commands. Send the readings as soon as they are taken, rather than waiting until all the readings are taken. Make sure to interactively show what the user is typing.

See the example below from the serial terminal:

```
get LDRR24 3 4
1032 1033 8000
get LDRR32 5 3
800 801 800 802 805
```

Hint: Scan UART character by character until the "enter key" press is detected.