Practical 9

In this prac we will implement a variable speed, bi-directional binary counter in C.
It will be remarkably similar to prac6, exacpt that we will use a delay loop rather than a timer, seeing as we don't know how to do ISRs in C yet.

Although not enforced, I strongly encourage you to make use of the register definitions provided in ST's header file. I've provided you with a template project structure.
Take a look at the makefile and note that for the main.o rule I've told GCC to look inside the Libraries folder for files which are #included. Check out the *stm32f0xx.h* file - that's where the struct and struct pointer definitions are.

Part 1: (2)
The value on the LEDs should be made to increment by 1 every 0.5 seconds.

Part 2: (2)
In the event that SW3 is held down, the LEDs should decrement by 1 rather than incrementing.

Part 3: (3)
In the event that SW2 is held down, the length of the delay should be modulated by POT1.
When the pot is outputting 0 V, the delay should last 0.1 seconds.
When the pot is outputting 3.3 V, the delay should last 0.5 seconds.
Linear in between.
The suggested way to implement this is that whenever the delay routine is called, the pot value is read and this is used to determine the number of loop iterations which the delay routine performs.

Useful questions:
How many loop iterations do I need to get 0.1 s delay?
How many additional loop iterations do I need to get 0.5 s?
What value must I multiply my maximum ADC result value by in order to provide these additional iterations?
I advise trying with 10-bit mode - it seems to produce the easiest numbers to work with.

Bonus: (1)
Instead of the LEDs resetting to a fixed value when the micro power cycles, the LEDs should remember what value they displayed when power was removed and return to counting from that same value when power is applied again.

Marked out of: 7
Available marks: 8