Practical 9

Attached is a template. It has the startup code I discussed in class such that it will copy the contents of the .data section LMA from flash into the .data section VMA in RAM and zeroing the .bss section thereby initialising all statically allocated variables. It also has the sections defined in the linker script and and the symbols being created there. Please check out the startup files and the linker script so that you know what's going on in them.

In main.c is a global array. Please don't change the name of this array or move it.
The marker may change the values and length of this array.

You're strongly encouraged to write a library file as we did in assembly to assist with interfacing with the peripherals. This will be useful to you in the prac exam.
Hence, you'll now be allowed to submit multiple source files and header files.

Part 1: (1)
When the micro powers on, it should display element 0 of the array on the LEDs.

Part 2: (3)
If SW0 is held down, the LEDs should be made to go to the next element in the array every every 0.5 seconds. After showing the last element it should cycle back to the first.
If the switch is not held, the LEDs should stay on whatever element they were showing last.
You must use a TIM6 to implement your timing.
Suggestion: TIM6 is always generating a 0.5 second interrupt. In the ISR you check the switch. It's okay if when you hold the switch down the first transition is faster than 0.5 seconds. (though this can be fixed!)

Part 3: (3)
When SW1 is pressed, the LEDs should immediately change to show the next element in the array. You will need to debounce noisy edges. 20 ms of debounce should be enough. You can use a small delay loop for debouncing.
Both the timer ISR and the main code will need to know which pattern is currently being displayed so that either of them can go to the next element. Hence, it seems appropriate to use a global variable to store the currently displayed index. (there are more elegant ways of doing it though…)

Part 4: (2)
When SW2 is held, display a value on the LEDs proportional to the voltage outputted by whichever potentiometer is outputting a lower voltage. If the lower output voltage is 0, the LEDs should display 0. If the lower output voltage is 0xFF, the LEDs should display 0xFF (or thereabout). Linear in between.
When SW2 is released the LEDs should go back to displaying whatever pattern in the array was shown before SW2 was held down.