Third year final project - 2024

You are required to implement a LED blinking project with a watchdog supervision capability. The LED blinking capability is handled through two software components. The first one is the LEDMgr and the second one is the GPIO.
The GPIO shall provide two functions. The GPIO_Init to initialize GPIO configuration for the PIN that used to control the LED. The GPIO_Write to write a specific value (0 or 1) to that pin.
The LEDMgr component shall provide two functions. LED_Init to initialize the LED component internal variables. The LED_Manage to manage the LED blinking actions using the GPIO_Write function.
The LED_Mange shall be called from a supper loop every 10ms and shall manage the LED blinking periodicity to be 500ms for each stage. You can use the standard Delay function to manage the timing.

The Watchdog management capability is handled through two different components the WDGDrv and the WDGM.
The WDGDrv shall implement a complete driver for the window watchdog peripheral in your STM32 microcontroller. The driver shall provide two functions, WDGDrv_Init.
The WDGDrv_Init shall configure the watchdog driver support the following features:

- **Set the Maximum timeout value to 64ms.**
- Activate the watchdog.

You should configure a timer with ISR called every 50ms:
In the timer ISR, you shall check the following conditions. If both of them are satisfied, it shall refresh the watchdog timer otherwise it will leave it to reset. The conditions are:

1. WDGM_MainFunction is not stuck.
2. The WDGM State set by the WDGM_MainFunction is OK.

The WDGM state shall be known by checking the return of the function WDGM_PovideSuppervisionStatus.

The WDGM shall provide supervision for the availability of the LEDM software entity. It shall provide three functions, WDGM_Init, WDGM_MainFunction, WDGM_PovideSuppervisionStatus and WDGM_AlivenessIndication.
The WDGM_Init shall initialize the WDGM_Inernal variables. The WDGM_AlivenessIndication shall be called from the LEDM_Manage function to detect its call at the correct timing. The WDGM_ProvideSuppervisionStatus shall provide the Status of the LEDM entity to the WDGDrv. The WDGM_MainFunction shall check the number of calls of the LEDM_MainFunction within a 100ms period. If the number of calls is between 8 and 12 then the status is OK. otherwise, the status is not OK. The WDGM_MainFunction shall be called periodically every 20ms.

You are required to provide the following:

1- Source code of the different components you
provide. 2- simulation file for your test.
3- a screen recording for the simulation of the following scenarios.

A. Positive scenario that checks the periodicity of the LED Blinking, Call of the LEDM_Manage, Call of the WDGM_MainFunction and refreshment of the WDGDrv.
    - You can provide the timing evidence by using test pins toggle on the oscilloscope.
B. Negative scenario that comments the call of the WDGM_MainFunction and checks that the watchdog reset occurred after 50ms.
C. Negative scenario that comments the call of the WDGM_AlivenessIndication from the LEDM_Manage while the WDGM_MainFunction is executed periodically and checks that the watchdog reset occurs after 100ms.
D. Negative scenario that changes the periodicity of the call of the LEDM_MainFunction to be every 5ms and checks that watchdog reset occurs after 100ms.

The header files of the components are attached, you are not allowed to change the APIs prototype or add new APIs.

You are not allowed to use any helper functions unless you are not able to access registers in c code. you can only use wdt_reset() and ISR() macros from avrlib.