Description:

You are required to design a TMU (Timer Management Unit).
This module has a capability to calculate different timing slots and call different user functions using Callback mechanism.

Detailed Requirements

1. **Read System Requirement Specifications**
   1. Implement **tmu_init** function using the below table. This function will be used to initialize the corresponding hardware timer.

Function Name	tmu_init
Syntax	enu_system_status_t tmu_init (enu_tmu_timer_id_t enu_tmu_timer_id, ptr_func_tmu_callback_t ptr_func_tmu_callback);
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (in):	enu_tmu_timer_id : Enumeration value for a valid timer ID
Parameters (in):	ptr_func_tmu_callback : Pointer to the Callback Function to be called
Parameters (out):	None
Parameters (in, out):	None
Return:	TMU_STATUS_SUCCESS: In case of Successful Operation
	TMU_STATUS_INVALID_TIMER_ID: In case of Invalid Timer ID (Out of Supported Number of Timers)
	TMU_STATUS_TIMER_IN_USE: In case that the timer is in use with timer driver for another functionality (Example: PWM, Time measurements, etc..)
	TMU_STATUS_INVALID_CALLBACK: In case of Invalid Callback Pointer Address
	TMU_STATUS_INVALID_STATE: In case that the timer already in use in TMU Module

   2. Implement **tmu_deinit** function using the below table. This function will be used to uninitialized the corresponding hardware timer.

Function Name	tmu_deinit

Syntax	enu_system_status_t tmu_deinit (enu_tmu_timer_id_t enu_tmu_timer_id);
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in):	enu_tmu_timer_id : Enumeration Value for the corresponding Timer ID
Parameters (out):	None
Parameters (in,out):	None
Return:	TMU_STATUS_SUCCESS: In case of Successful Operation
	TMU_STATUS_INVALID_TIMER_ID: In case of Invalid Timer ID (Out of Supported Number of Timers)
	TMU_STATUS_INVALID_STATE: In case that the timer was not used previously || not in use currently in the TMU Module

3. Implement **tmu_subscribe_callback** that registers a specific callback function for a specific timer id. As this callback will be called when this timer ID expires (please create a table for this function as the previous functions).
4. Implement **tmu_start_timer** that starts a specific timer id with specific time for a specific mode of operation (please create a table for this function as the previous functions).
5. Implement **tmu_stop_timer** that stops a specific timer id (please create a table for this function as the previous functions).
6. Implement **tmu_reset_timer** that resets a specific timer ID with a specific time for a specific mode of operation. Reset Means stop the timer from its current operation then starts the timer again (please create a table for this function as the previous functions).
7. Implement **tmu_get_first_available_timer_index** that informs the user with the first valid timer index.
    1. **Example**: A microcontroller has 3 Timers (0,1,2). The TMU will use timer 0, timer driver uses timer 1. Then this function will return 2 to the user (please create a table for this function as the previous functions).
8. Implement **tmu_dispatcher** that runs in the super loop and call the user callback for the expired timer (please create a table for this function as the previous functions).

2. Module testing

    1. Implement an application that calls the TMU module and use 2 callbacks to toggle LED_0 (Every 3 Seconds) and LED_1 (Every 5 Seconds) for only one time.
    2. Repeat the point 1 for 3 times in your super loop.

3. **Prepare your design**
    1. Create a PDF file with the name **Timer Management Unit Design**
    2. The design document should contain the below fields
        1. Cover Page
        2. Table of content
        3. Project introduction
        4. High Level Design
            1. Layered architecture
            2. Modules Descriptions
            3. Drivers' documentation
            4. UML
            5. Sequence diagram
        5. Low Level Design
            1. Provide the flowchart for each function in each module
            2. Pre-compiling configurations for each module
            3. Linking configurations for each module

Delivery

1. Deliver the Design Document
2. English Video recording 5 minutes maximum discuss your design