**Challenge Requirements Document**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Wave** |  |  | **Trainer Name** | **Hossam Adel** |
| **Week Number** |  |  | **Challenge Name** | **Timer Management Unit** |
| **Duration** |  |  | **Type** | **Project** |

**Team Size**

2

**SW/HW environment**:

Timer driver based on the ISR and Dev Board.

**Restrictions (peripherals, configurations, what to use and not to use):**

We going to Design the software using whatever we had in our HW to get the highest throughput.

**Challenge Requirements**

**AGILE\_REQ1:**

Implement TMU ( timer management unit), Module has a capability to calculate different timing slots and call different user functions using Callback mechanism.

**AGILE\_REQ2:**

Implement TMU\_Init by following the next table :

|  |  |
| --- | --- |
| Function Name | TMU\_Init |
| Syntax | EnmTMUError\_t TMU\_Init (const TMU\_ConfigType \* ConfigPtr ) |
| Sync/Async | Synchronous |
| Reentrancy | Non Reentrant |
| Parameters (in): | ConfigPtr Pointer to a selected configuration structure |
| Parameters (out): | None |
| Parameters (inOut): | None |
| Return: | EnmTMUError\_t | one of predefine enumeration number |

**AGILE\_REQ3:**

The function TMU\_Init shall initialized the hardware timer module.

**AGILE\_REQ4:**

Implement TMU\_DeInit by following the next table :

|  |  |
| --- | --- |
| Function Name | TMU\_DeInit |
| Syntax | EnmTMUError\_t TMU\_DeInit ( void ) |
| Sync/Async | Synchronous |
| Reentrancy | Non Reentrant |
| Parameters (in): | None |
| Parameters (out): | None |
| Parameters (inOut): | None |
| Return: | EnmTMUError\_t | one of predefine enumeration number |

**AGILE\_REQ5:**

The function TMU\_DeInit shall uninitialized the hardware timer module.

**AGILE\_REQ6:**

Implement TMU\_Start\_Timer by following the next table :

<Part of my training let the Sprinters imaging how to implement the function from scratch>

Hint: this table need to be filled be the Sprinters.

|  |  |
| --- | --- |
| Function Name | TMU\_Start\_Timer |
| Syntax | EnmTMUError\_t TMU\_Start\_Timer (TMU\_Cfg\_start \* TMUStartPtr) |
| Sync/Async | Synchronous |
| Reentrancy | Non Reentrant |
| Parameters (in): | ConfigPtr Pointer to a selected TMUStartPtr structure |
| Parameters (out): | None |
| Parameters (inOut): | None |
| Return: | EnmTMUError\_t | one of predefine enumeration number |

**AGILE\_REQ7:**

The function TMU\_Start\_Timer shall start specific software timer object that supposed to be added in internal list and define the callback function that supposed to get called form the dispatcher function [REQ10].

**AGILE\_REQ8:**

Implement TMU\_Stop\_Timer by following the next table :

<Part of my training let the Sprinters imaging how to implement the function from scratch>

Hint: This table need to be filled be the Sprinters.

|  |  |
| --- | --- |
| Function Name | TMU\_Stop\_Timer |
| Syntax |  |
| Sync/Async |  |
| Reentrancy |  |
| Parameters (in): |  |
| Parameters (out): |  |
| Parameters (inOut): |  |
| Return: |  |

**AGILE\_REQ9:**

The function TMU\_Stop\_Timer shall Stop the specific software timer object and remove it from the TMU internal list.

**AGILE\_REQ10:**

Implement TMU\_MainFunction by following the next table :

<Part of my training let the Sprinters imaging how to implement the function from scratch>

Hint: This table need to be filled be the Sprinters.

|  |  |
| --- | --- |
| Function Name | TMU\_Dispatch |
| Syntax |  |
| Sync/Async |  |
| Reentrancy |  |
| Parameters (in): |  |
| Parameters (out): |  |
| Parameters (inOut): |  |
| Return: |  |

**AGILE\_REQ11:**

The function TMU\_Dispatch shall Run the basic logic code to calculate the timing and call the user function depends on the timeout event for each software timer object.

This function part of the super loop code.