**Timer Module design description.**

**Purpose:**

Timer module is intended to support operation of multiple simultaneous timers.

**Overview:**

Timer module uses one hardware timer.   
The hardware timer must be started before any new SW timer is created. Same HW timer is used to track time since system start-up.   
Timeout of SW timer is checked within HW Timer ISR. The user defined callback function is called in the context of HW timer ISR on timeout match.   
The hardware timer period is chosen so to match best resolution of SW timer and also to exclude interrupt overhead.

**Timer module user interface:**

User is able to start and stop HW timer without any parameters.

User is able to retrieve time in milliseconds since system start-up.

To add new timer user must specify timeout, timer mode, timeout resolution, pointer to callback function and pointer to data that should be passed to callback function.  
Timer mode could be ‘single’, ‘cyclic’ or ‘single-cyclic’. Timer of mode ‘single’ is freed on its timeout, so Timer module will add new timer on its place when requested. Timer of mode ‘single-cyclic’ is set to mode ‘cyclic’ after one-shot timed out. Timeout resolution is specified as seconds or milliseconds. Timeout must be greater than zero for specified timer mode.  
Timer module returns recently created timer Id or error code in case of no free timers and incorrect user timeouts.  
Timer starts counting immediately after it has been created.   
User is not able to delete/modify previously created timer.  
Timer module is reset on system power cycle and with HW Timer start/stop.