**Hook Functions**

**Idle Hook Function**

The idle task can optionally call an application defined hook (or callback) function - the idle hook. The idle task runs at the very lowest priority, so such an idle hook function will only get executed when there are no tasks of higher priority that are able to run. This makes the idle hook function an ideal place to put the processor into a low power state - providing an automatic power saving whenever there is no processing to be performed.

The idle hook will only get called if configUSE\_IDLE\_HOOK is set to 1 within FreeRTOSConfig.h. When this is set the application must provide the hook function with the following prototype:

**void vApplicationIdleHook( void ); In task.c**

The idle hook is called repeatedly as long as the idle task is running. It is paramount that the idle hook function does not call any API functions that could cause it to block. Also, if the application makes use of the vTaskDelete() API function then the idle task hook must be allowed to periodically return (this is because the idle task is responsible for cleaning up the resources that were allocated by the RTOS kernel to the task that has been deleted).

Other changes done in config.h file

**Every 100ms the task1 prints the message.**

**Whenever the button is pressed LED toogles**

**In between ideal task rund that does the task of cleaning TCB is any task is deleted.**