RTOS In A Nutsel

short line

# **Challenge**

# Sprint Number (3)

## User story (1):

Use an RTOS project to define two tasks with the following specifications.

1. **Task A:** Responsible for monitoring push button 1 state and sending data to LCD.
2. **Task B:** Responsible for monitoring push button 2 state and sending data to LCD.
3. **Task C:** Responsible for writing data on LCD.

**Task C** shall read a queue continously and print the content of the data if found, The data shall contain two informations:

1. Any text with maximim size 16 character.
2. Which button sent this text (Button ID).

If button 1 is pressed then **Task A** shall write to the queue (Any text + Its ID), And if button 2 is pressed then **Task B** shall write to the queue (Any text + Its ID). If button ID belongs to button 1 then the corresponding text will be displayed on row 1 of the LCD, And If button ID belongs to button 2 then the corresponding text will be displayed on row 2 of the LCD. Any text displayed on the LCD shall be displayed for 500ms and cleared.

## User story (2) – Final Project:

Use an RTOS project to design and implement a chatting application between two micro-controllers with the following requirements:

1. The two micro-controllers shall communicate over a UART channel (115200 Kbps) full duplex.
2. Each micro-controller will contain 1 keypad, 2 LEDs, 1 LCD, and 1 push button.
3. LCD shall display the received message in the lower row. The received message will be displayed when received for two seconds and then cleared.
4. LCD shall display the sender input from keypad in the upper row. When sender press the push button the message displayed in the upper row is sent to the other micro-controller through UART channel and the message is cleared from the sender LCD.
5. LED 1 is turned ON for 500ms when a new message is received.
6. LED 2 is turned ON for 200ms when a new message is sent.

Assume any missing configurations (Ex. Preemptive or non-preemptive – Periodicity – Priority). Ensure that the system responds on time with high performance.

## Group members number (1, 2 or 4):

* User story (1) require 1 person only.
* User story (2) – Final Project requires 2 persons.

## SW/HW input environment:

TivaC - Development board.

## Test (If exists):

NA

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

1. All applications implemented in this sprint shall use two type of tasks. An **init task** that comes only once to initialize all required interfaces. A **cyclic task** that comes on a fixed period to execute a certain logic.
2. In this sprint LEDs, LCDs, keypad and push buttons are considered individual and independent objects that should have its own task to handle its logic (ex. A task for LCD – A task for LED – A task for another LED – Etc...). A single task shall not handle two objects of the same category.
3. Driver for LCDs, keypad and push buttons shall completely be developed by the student and is allowed only to use the GPIO, UART drivers from the TivaWare library.
4. In this sprint configure RTOS **configTICK\_RATE\_HZ** to 1000 HZ.
5. Use only one or more of these methods (Semaphores – Mutex – Event flags) to achieve tasks synchronization and resource sharing.
6. Use only one or more of these methods (Semaphores – Mutex – Event flags) to achieve tasks synchronization and resource sharing.
7. Use only one or more of these methods (Messages Queues – Mailbox) to achieve tasks inter-communication.