

Lab Objective

To demonstrate FreeRTOS task management using the following features: -

- GPIO interfacing with LEDs, Switches
- Task synchronisation using Semaphores
- UART interfacing

Reference: FreeRTOS Reference Manual Chapter

☐ [Semaphore / Mutexes](#)

[xSemaphoreCreateBinary\(\)](#)

[xSemaphoreCreateBinaryStatic\(\)](#)

[vSemaphoreCreateBinary\(\)](#)

[xSemaphoreCreateCounting\(\)](#)

[xSemaphoreCreateCountingStatic\(\)](#)

[xSemaphoreCreateMutex\(\)](#)

[xSemaphoreCreateMutexStatic\(\)](#)

[xSemaphoreCreateRecursiveMutex\(\)](#)

[xSemaphoreCreateRecursiveMutexStatic\(\)](#)

[vSemaphoreDelete\(\)](#)

[xSemaphoreGetMutexHolder\(\)](#)

[uxSemaphoreGetCount\(\)](#)

[xSemaphoreTake\(\)](#)

[xSemaphoreTakeFromISR\(\)](#)

[xSemaphoreTakeRecursive\(\)](#)

[xSemaphoreGive\(\)](#)

[xSemaphoreGiveRecursive\(\)](#)

[xSemaphoreGiveFromISR\(\)](#)

Part A: Countdown Timer

Create a project with a single task that counts down every second.

- The timer starts counting down from 30 seconds on start-up.
- LED2 toggles every second when the time is 10 seconds or less.
- The timer stops when it reaches 0 seconds.

```
COM4
Starting Display Task
Time: 30
Time: 29
Time: 28
Time: 27
Time: 26
```

Part B: Using a Semaphore to Synchronise 2 Tasks

Create a second task called buttonTask. This task should read the switch and send a semaphore to the countdown task when the switch is pressed. On taking the semaphore, the countdown task should re-start the timer at its initial count.

```
Starting Display Task
Time: 30
Starting Button Task
Time: 29
Time: 28
Time: 27
Time: 26
Time: 25
Button Pressed, Giving Semaphore
Semaphore Taken, Resetting Timer
Time: 30
Time: 29
Time: 28
Time: 27
```

Name	Priority (B...	Start of S...	Top of St...	State	Event Ob...	Stack Usa...	Run Time...
Button Task	5/5	0x20001...	0x20001...	RUNNING		312B / 8...	>99%
Count Task	6/6	0x20001...	0x20001...	DELAYED		512B / 8...	<1%
IDLE	0/0	0x20000...	0x20000...	READY		44B / 51...	0%
Tmr Svc	6/6	0x20000...	0x20000...	BLOCKED	TmrQ	168B / 1...	<1%

Part C: UART Data Input

Add a third task to interface to the UART. The user types the character 'p' into the serial terminal to pause or restart the timer countdown. The UART task should then send a semaphore. The countdown task should take the semaphore and pause or restart the countdown timer.

See the UART Receive Polling padlet for more information on the UART receive API functions.

```
Starting Display Task
Time: 30
Starting UART Task
Starting Button Task
Time: 29
Time: 28
Character Received: p
UART Semaphore Taken

Pausing Timer
Time: 27
Time: 27
Time: 27
Time: 27
Character Received: p
UART Semaphore Taken

Restarting Timer
Time: 27
Time: 26
Time: 25
Time: 24
Time: 23
Time: 22
Button Pressed, Giving Semaphore
Button Semaphore Taken, Resetting Timer

Time: 30
Time: 29
Time: 28
Time: 27
```

