

## Lab Objective

To demonstrate the synchronisation of FreeRTOS tasks with interrupt handlers using: -

- Semaphores
- Event Group Bits

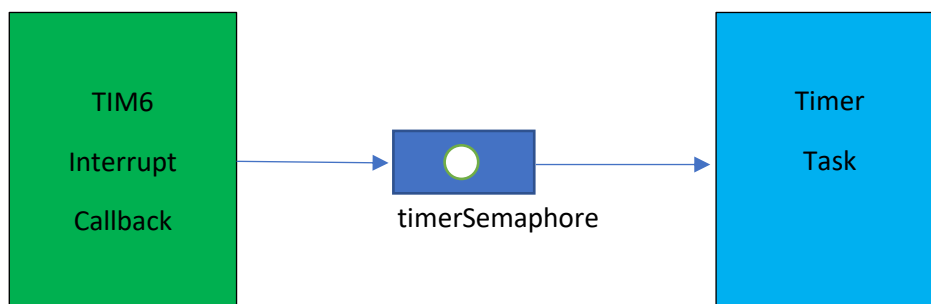
```
xSemaphoreGiveFromISR
(
    SemaphoreHandle_t xSemaphore,
    signed BaseType_t *pxHigherPriorityTaskWoken
)

BaseType_t xEventGroupSetBitsFromISR(
    EventGroupHandle_t xEventGroup,
    const EventBits_t uxBitsToSet,
    BaseType_t *pxHigherPriorityTaskWoken );
```

### Part A: Synchronising a Timer Interrupt Handler and a Task using a Semaphore

Create a semaphore and a single task. Configure the TIM6 timer to generate an interrupt request every second. The TIM6 interrupt callback function should give a semaphore. The timer task should take the semaphore, increment a time variable and print it.

See overleaf for the timer configuration. More information on timer configuration and timer API is available in the padlet topic.

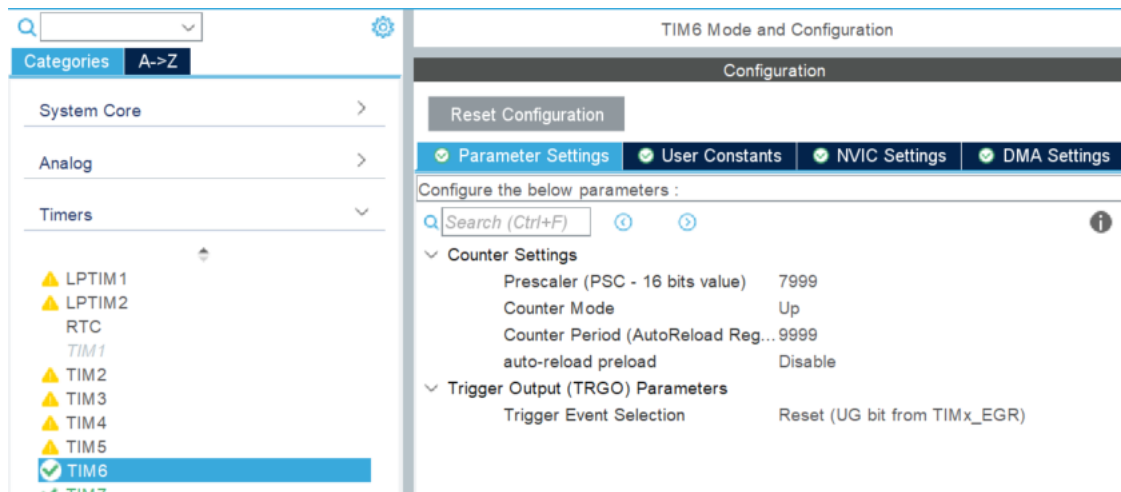


```
COM4
Starting Timer Task
Timer: 01
Timer: 02
Timer: 03
Timer: 04
Timer: 05
```

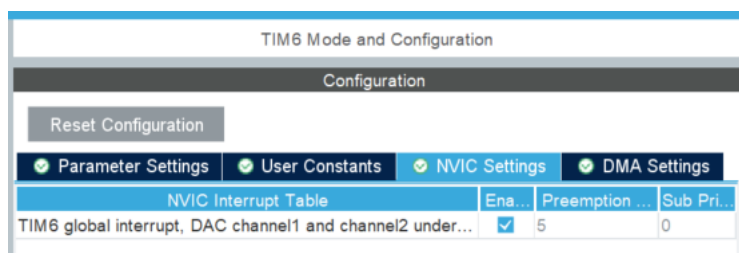
**Timer 6 Configuration**

Overflow time = (Prescaler + 1) \* (Auto-reload + 1)/Timer Clock

Overflow Time = 8000 \* 10000/80000000 = 1 second



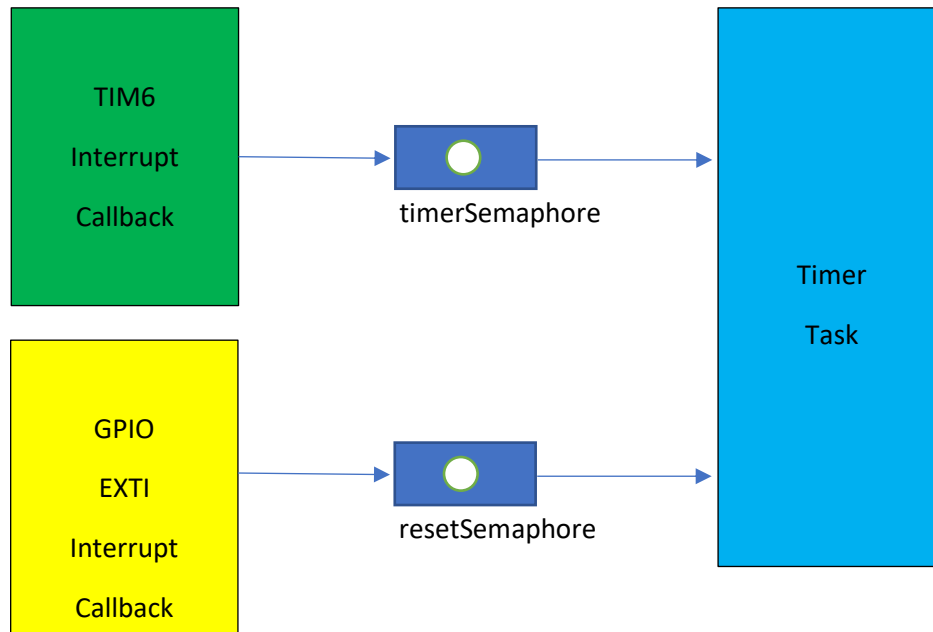
Enable the TIM6 interrupt.



**Part B: Synchronising a GPIO Interrupt Handler and a Task using a Semaphore**

Add a second semaphore. Configure the GPIO button switch pin to generate an interrupt request every time the switch is pressed. The GPIO EXTI interrupt callback function should give a semaphore. The timer task should take the semaphore and reset the time variable to 0.

See the GPIO pin interrupt padlet for more information on configuring and programming GPIO interrupts.



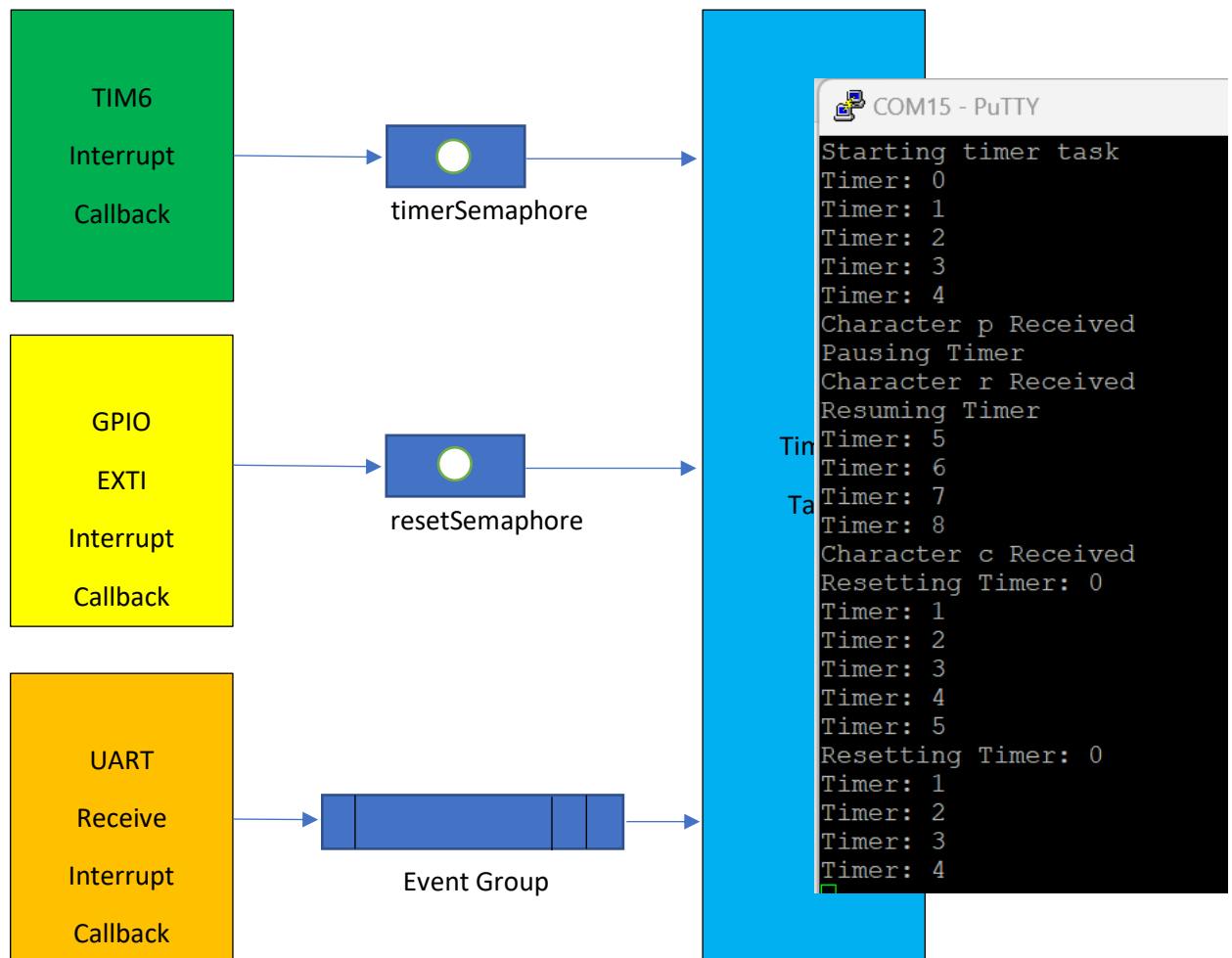
```
COM5 - PuTTY
Starting timer task
Timer: 0
Timer: 1
Timer: 2
Timer: 3
Resetting Timer: 0
Timer: 1
Timer: 2
Timer: 3
Timer: 4
Timer: 5
Timer: 6
```

**Part C: Synchronising a UART Interrupt Handler and a Task using an Event Group**

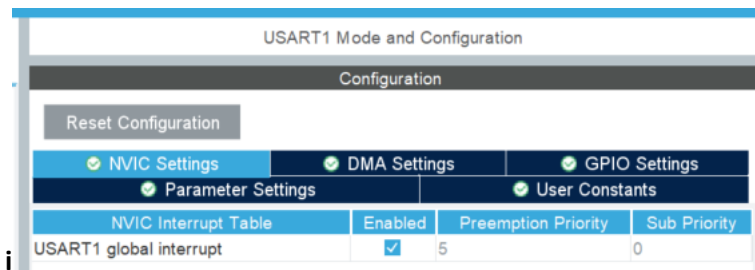
Add an Event Group. Configure the UART1 peripheral to generate an interrupt request every time a character is entered into the serial terminal. The UART interrupt handler sets different event group bits depending on the character typed. The timer task should wait for the event group bits to be set and take the appropriate action as shown below.

See overleaf for the UART configuration. More information on configuring the UART and programming the UART interrupt callback function is available on the UART interrupt configuration padlet.

Serial Character	Timer Task Action
'p'	Pause Timer
'r'	Resume Timer
'c'	Clear timer to 0



Enable the UART interrupt.



Change the FreeRTOS Config and Include parameters as shown below. This is required to set event group bits from an interrupt handler.

