

## E.- Gestión de colas en FreeRTOS

- **Objetivo:**
  - Uso del IDE (edición, compilación y depuración de programas)
  - Uso de GPIO & FreeRTOS (manejo de Salidas y de Entradas Digitales en Aplicaciones)
  - **Modificar/Documentar** lo que se solicita en c/items
- **Referencias:**
  - [FreeRTOS, API Reference, FreeRTOS Documentation: Mastering the FreeRTOS Real Time Kernel - a Hands On Tutorial Guide, FreeRTOS V10.0.0 Reference Manual, Book companion source code](#)
  - [CMSIS-RTOS Documentation, CMSIS-RTOS API Version 1](#)
  - [FreeRTOS on STM32 CMSIS\\_OS API \(CMSIS\\_V1\)Archivo](#)

E.1.- Tome del Campus los proyectos correspondientes a los siguientes ejemplos de [Mastering the FreeRTOS Real Time Kernel - a Hands On Tutorial Guide](#):

[freertos\\_book\\_Example020](#) - Re-writing vPrintString() to use a semaphore  
[freertos\\_book\\_Example030](#) - Synchronizing tasks

### Se solicita:

- **Documentar** mediante un diagrama temporal la distribución del tiempo de CPU entre tareas, Kernel, Interrupciones (buscar imagen en: [Mastering the FreeRTOS Real Time Kernel - a Hands On Tutorial Guide](#)), detallar qué ocurre en cada cambio de contexto
- **Documentar** observaciones
- **Documentar** el valor de **time slice** de FreeRTOS, dónde y cómo modificarlo ([FreeRTOSConfig.h](#))
- **Documentar** el efecto de **modificar time slice** sobre la **ejecución** de **tareas** (probar con 1000mS/100mS/10mS/1mS)
- **Documentar** el **criterio** a aplicar para la **elección** del **valor** de **time slice** para una aplicación

E.2.- Tome del Campus el proyecto:

[freertos\\_app\\_Example5\\_6](#) - Práctica Obligatoria (5 de 6)

### Se solicita:

- **Documentar** mediante un diagrama temporal la distribución del tiempo de CPU entre tareas, Kernel, Interrupciones, detallar qué ocurre en cada cambio de contexto
- **Documentar** observaciones

E.3.- Tome del Campus el proyecto:

[freertos\\_app\\_Example5\\_6](#) - **Práctica Obligatoria (5 de 6)**

### Se solicita:

- **Modificar** [app.c](#), [task\\_Button.c](#), [task\\_Led.c](#) y [app\\_Resources.h](#) para que:
  - la función **vTaskButton** use un semáforo binario para controlar la función **vTaskLed** (**vSemaphoreCreateBinary()** / **xSemaphoreTake()** / **xSemaphoreGive()**)
  - dicho semáforo debe reemplazar a la cola de mensajes (**ledFlag**), para que las tareas (**TaskButton** y **TaskLed**) sigan haciendo exactamente lo mismo que hacen en el proyecto original
- **Documentar** observaciones
- **Subir al Campus:** La carpeta **App comprimida** (archivo del tipo ".zip" o ".rar"), **nombrar:** **RTOS I - PO 5\_6 - Apellidos\_Nombres.zip**