

# Respuestas Parcial TDII Prof. Alarcon -

## 09/07/2025

Alumno: Rios, Alexander Daniel      LEG: 153.172-4

### Enunciado

Se va a implementar en un sistema con FreeRTOS que tiene dos pulsadores que se activan al ponerlos a tierra (se recomienda utilizar los pull-ups de los pines) y un led. El funcionamiento es el siguiente:

1. El sistema monitorea el estado de ambos pulsadores y mide (con resolución de milisegundos) el tiempo que se encuentran puestos a tierra. Los pulsadores pueden estar presionados los dos a la vez y tiene que poder medirse el tiempo de ambos de manera correcta.
2. El led comienza parpadeando con un tiempo de encendido igual al tiempo de apagado de 50ms.
3. Al liberarse el primer pulsador que se puestos a tierra, se determina el tiempo de encendido y apagado del led. En el caso de que el tiempo de presión sea superior a 50 ms (si es menor se ignora) y si es mayor a 5000 ms el led encenderá y apagará 5000 ms.

Se pide:

- A. Determinar la cantidad de tareas de FreeRTOS necesarias para implementar el sistema. Indicar si hay tareas que pueden ser parametrizadas.
- B. Indicar claramente cómo realiza la comunicación entre las tareas evitando condiciones de competencia y esperas activas.
- C. Implementar el sistema.

Notas:

- Para LPC1769 se recomienda utilizar los pines 2.11 y 2.12 y el led de la placa (0.22). Utilizar la función Chip\_IOCON\_PinMuxSet( LPC\_IOCON, PUERTO, PIN, IOCON\_MODE\_PULLUP ); para configurar los pull-ups.
- Para STM32F103C8 se recuerda que el led es activo en bajo y está en el puerto C13. Utilizar el configurador de la herramienta para el hardware.
- La función TickType\_t xTaskGetTickCount( void ); mide el tiempo en ms desde que inició el sistema hasta que se la llama.

Respuesta:

- A. La cantidad de tareas de FreeRTOS necesarias para implementar el sistema son 3, de las cuales dos pueden ser parametrizadas.
  - **tarea\_led:** Esta tarea se encarga de setear el tiempo de parpadeo del led, el cual inicialmente es de 50ms encendido y 50ms apagado, por otro lado también se encarga de leer la cola de la cual extrae el nuevo tiempo de parpadeo del led.
  - **tarea\_pls:** Esta tarea permite leer el tiempo durante el cual es presionado un

determinado pulsador, una vez que este es liberado, el tiempo medido es utilizado para modificar el tiempo de parpadeo del led, esta tarea se comunica con la **tarea\_led** a través de una cola. Esta tarea está parametrizada para utilizar dos diferentes pulsadores y de esta forma obtenemos dos tareas que hacen lo mismo con sus respectivos pulsadores. Por otro lado, si el tiempo medido es inferior a 50ms no se envía nada por la cola y para cualquier otro caso el tiempo medido es enviado por cola teniendo en cuenta que si el tiempo es mayor a 5000ms el tiempo se trunca a 5000ms. Ambos pulsadores pueden ser presionados al mismo tiempo y al soltarse cualquier de estos se actualizará el tiempo de parpadeo del led, por lo que el led parpadeará finalmente al tiempo medido por el último pulsador liberado.

- B. La comunicación entre tareas se realiza con una cola de mensajes. Las tareas correspondientes a los pulsadores actúan como productores, ambas tareas escriben en la cola de mensajes, donde, dado el caso de que ambas quieran escribir en la cola al mismo tiempo, al ser de igual prioridad, lo hará primero la que lleve mayor tiempo bloqueada evitando condiciones de competencia. La tarea del led actúa como consumidor si esta tarea intenta leer la cola vacía bloquea la actualización del tiempo de parpadeo del led pero no la inversión del estado del led. Una vez que alguna tarea escribe en la cola de mensajes, la tarea led lee el contenido del primer mensaje y actualiza la demora del parpadeo del led.  
Para bloquear una tarea y evitar esperas activas en todo momento se llama a las APIs brindadas por el RTOS las cuales permiten que la correspondiente tarea no sea planificada por el scheduler.