

Laboratorio 4

FreeRTOS

Objetivo

Utilizar el sistema operativo en tiempo real **FreeRTOS** para trabajar con una arquitectura de software de **multitasking** tomando como base los requerimientos del laboratorio 3.

Evaluación y metodología de trabajo

El docente asistirá a los estudiantes realizando sugerencias, devoluciones y aclarando conceptos cada vez que se necesite y sea pertinente.

Se evaluará el trabajo de cada grupo durante el laboratorio, y se considerarán para dicha evaluación los siguientes criterios:

- A. Organización del grupo, involucramiento de cada uno de los integrantes.
- B. Completitud y correctitud de las soluciones a las tareas indicadas en el laboratorio en el tiempo de clase.
- C. Utilización de buenas prácticas de programación.

Detalles

Se tomará como punto de partida el laboratorio 3, pero se deberá modificar teniendo en cuenta las siguientes tareas:

- Una tarea (TASK A) será encargada de hacer parpadear el LED rojo (LED A).
- Una tarea (TASK B) será encargada de gestionar el USB/UART, y recibir y procesar los comandos recibidos.

Al comando de encendido/apagado de un LED RGB se le debe agregar un parámetro de tiempo (en segundos), que será el tiempo en el que efectivamente se ejecutará el comando (es decir, la cantidad de segundos que pasarán entre que se pasa el comando por USB y efectivamente se ejecuta). Por cada uno de estos comandos, se deberá crear un timer one-shot que tomará como timerID un puntero a un bloque de memoria creada dinámicamente con la información del número de LED y color a modificar.

El encendido/apagado efectivo de cada LED RGB se hará desde otra tarea que se describe a continuación (TASK C) y que sacará los comandos desde un queue.

Si el parámetro de tiempo es cero, TASK B colocará directamente el comando en el queue. Si el parámetro de tiempo es distinto de cero, TASK B creará un timer, y será el callback del timer que pondrá el comando en el queue.

Documentación timers:

<https://www.freertos.org/FreeRTOS-Software-Timer-API-Functions.html>

- Una tarea (TASK C) que recibirá los comandos de LEDs RGB a modificar a través del queue.

Tener en cuenta que la información del último LED RGB modificado será utilizada por más de una tarea.

Para completar correctamente este laboratorio, se deberá por tanto poner en funcionamiento FreeRTOS, utilizando timers, memoria dinámica, queues y semáforos.