

Laboratorio N° 5

Introducción a RTOS

Actividad 1)

Se realizó una actividad, en la cual se hacía titilar a una frecuencia de 1Hz un LED a través del **FreeRTOS**. Se consiguió el objetivo definiendo dos tareas, **TaskBlinkOn** y **TaskBlinkOff**, que alternan entre dos estados, **READY**, **RUNNING** y **BLOCKED**.

El kernel se encarga de seleccionar en base a las prioridades estipuladas en las definiciones de las tareas, por lo que la transición de las tareas entre los estados **READY** y **RUNNING** es transparente para nosotros. Con el código desarrollado, la única función que hace pasar a una tarea al estado **BLOCKED** es *vTaskDelay(TicksToDelay)*, la cual manda la tarea que llama a la función a le estado de bloqueo por *TicksToDelay* ticks.

Se realizó una llamada inicial a *vTaskDelay(FRECUENCIA/2)* en la tarea *TaskBlinkOff* para poder sincronizar el prendido y apagado del led.

Luego en cada tarea se escribió un bucle infinito el cual prende o apaga el led, según la tarea y se llama a *vTaskDelay(FRECUENCIA)*.

Actividad 2)

Se realizó una actividad, en la cual se hacían titilar dos LEDS a distintas frecuencias, donde a uno de ellos se le puede cambiar la frecuencia ingresando un número por teclado.

Para esto, se definieron dos timers con dos funciones de callback asociadas, *callbackTimer1* y *callbackTimer2*. También se definió una tarea *SerialReader*, esta leerá caracteres ingresados por consola y llamará a la función *xTimerChangePeriod* para cambiar la frecuencia del titileo de uno de los LED.

Ventajas de usar timers de software

- Se abarantan costos porque no necesito hardware extra (en relación a usar timers de hardware).
- Se usan menos recursos en relación a definir una tarea para cada acción que se ejecute periódicamente.