



Laboratorio N° 5
Introducción a RTOS

Fecha de evaluación: Viernes 18/10/19

Objetivo: El objetivo de este laboratorio consiste en introducir a los alumnos en un sistema operativo de tiempo real como FreeRTOS.

Desarrollo: El laboratorio deberá realizarse en comisiones de no más de 2 alumnos. Al finalizar la evaluación, se deberá comprimir y enviarse por mail respetando el siguiente formato: *LaboratorioX-ApellidosComision.zip*.

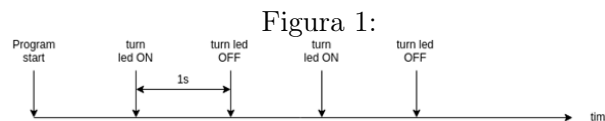
Descripción del hardware

El Hardware a utilizar se compone de:

- una placa Arduino Uno [1] con un microcontrolador ATmega328P [2].
- led provisto por la cátedra.
- protoboard y cables para realizar el conexionado.

Actividad 1: Introducción a FreeRTOS

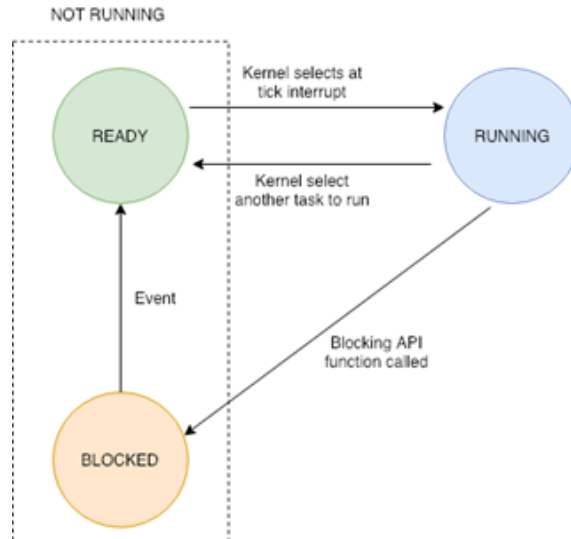
1. Escribir un programa con el kernel FreeRTOS [3] (Atmel ó framework de Arduino) para hacer que un LED parpadee con una frecuencia de 1 Hz, como indica la Figura 1. Para la implementación del parpadeo, utilice dos tareas, una que encienda el LED, y otra que lo apague. Analice los posibles estados de una tarea en FreeRtos (figura 2)



Actividad 2: Productores y consumidores en FreeRTOS

Implementar tres tareas con el kernel de FreeRTOS [3] (Atmel o framework de Arduino), dos productores y un consumidor. Los dos productores escriben datos (`uint32_t`) en un vector(array) compartido de cinco posiciones mientras que el consumidor los lee y los imprime a través del puerto serie. El primer productor lo hace cada 50 ms y el segundo productor lo hace cada 100 ms. Por su lado, el consumidor lee un dato del vector cada 75 ms. Las tareas de escritura tienen la misma

Figura 2: Posibles estados de una tarea



prioridad de ejecución mientras que la tarea de lectura tiene una prioridad superior. Recordar que antes de escribir o leer del vector es necesario comprobar que hay espacio suficiente y que no está vacío. Para ello es necesario utilizar una variable adicional que indique el tamaño actual. En caso que no haya espacio suficiente o no haya datos, la tarea se debe bloquear hasta que quede una posición libre o haya un dato disponible. Se pide:

1. Implementar el sistema descrito anteriormente utilizando solo mecanismos de exclusión mutua entre procesos ¿Cuántos de dichos elementos son necesarios para garantizar el correcto funcionamiento del sistema en cuanto a concurrencia? ¿Que elementos hay que proteger?
2. ¿Qué diferencia hace configurar un RTOS (como FreeRTOS) en modo apropiativo o hacerlo en modo cooperativo? O sea ¿cuál es la diferencia entre estos dos modos?

Referencias

- [1] Arduino Uno WebSite. <https://arduino.cc/en/Main/arduinoBoardUno>.
- [2] Atmel AVR ATmega48A/48PA/88A/88PA/168A/168PA/328/328P Data Sheet.
- [3] FreeRTOS. <https://www.freertos.org/>.