

# **Sistemas Basados en Microprocesador**

## **Bloque 2 – Práctica 6**

**Sistema operativo en tiempo real.**  
**Gestión de un joystick utilizando**  
**CMSIS-RTOS2**

## **Introducción y objetivos de la práctica.**

El objetivo de esta práctica es desarrollar un driver software para manejar el joystick disponible en la tarjeta *mbed application board*. Para ello se hará uso de los recursos básicos que ofrece el sistema operativo RTX5 utilizado desde la API CMSIS-RTOS2: threads, timers virtuales, señalización entre threads, etc. Se manejarán los cinco gestos disponibles en el joystick de la *mbed app board* y como en prácticas anteriores, se busca disponer de una serie de funciones de librería en ficheros .c y .h para poder utilizarla en futuros proyectos. Por tanto, es muy importante depurar y obtener un “driver” libre de errores que permitan al usuario interactuar con el microcontrolador.

Al igual que prácticas anteriores debe leer la documentación que se indica en el siguiente apartado

## **Documentación**

Además de los documentos empleados en las prácticas anteriores debe disponer de los siguientes:

- CMSIS-RTOS2 Documentation:

<https://www.keil.com/pack/doc/CMSIS/RTOS2/html/index.html>

## **Ejercicio 1**

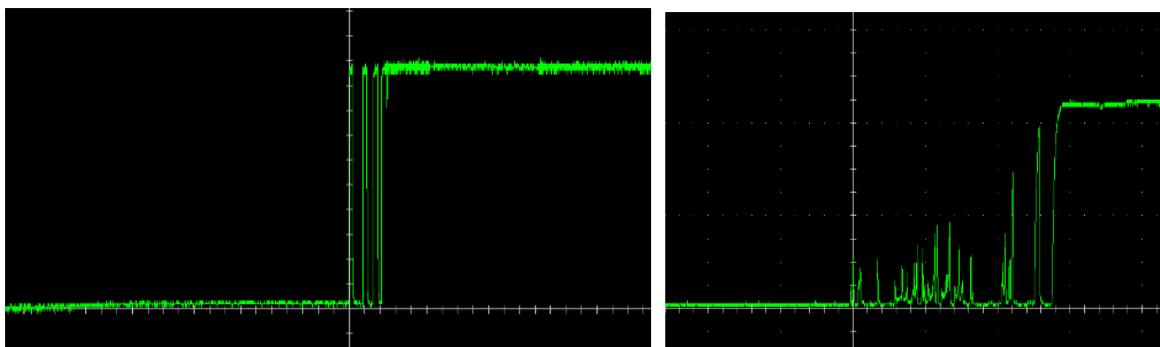
Utilizando la tarjeta NUCLEO-144 STM32F429ZI desarrolle un proyecto que responda a los siguientes requisitos:

- Se utilizará el **gesto RIGHT** para generar interrupciones externas.
- Se utilizarán los **LEDs rojo (LD3) y verde (LD1)** para mostrar el estado del sistema.
- **En el arranque del sistema se encenderá el led verde y se arrancará una cuenta de tres segundos. Pasados esos tres segundos sin que se pulse nuevamente el gesto RIGHT el sistema apagará el led verde y encenderá el led rojo.**
- **Si se pulsa el gesto RIGHT mientras la cuenta de tres segundos está activa, se reiniciará un nuevo periodo de cuenta de tres segundos.**
- Una vez que el sistema se bloquea (LED rojo encendido), no se procesarán peticiones de reinicio de la cuenta con el gesto RIGHT, solo un reset hace que el sistema vuelva a la situación inicial.

En este ejercicio la temporización debe ser obtenida mediante timers virtuales.

## **Ejercicio 2**

El joystick incluye cinco pulsadores encapsulados en un mismo dispositivo. Es conveniente recordar que los pulsadores presentan el conocido efecto de los rebotes, debido al choque de las láminas metálicas con los que están contruidos. En la siguientes figuras se muestran dos ejemplos de este fenómeno.



A la vista de la figuras se puede observar cómo transcurre un tiempo hasta que el valor de la pulsación es estable. El número de rebotes y el tiempo que transcurre entre ellos dependen del tipo y estado del pulsador. Asimismo, una persona necesita un tiempo mínimo para actuar sobre un pulsador.

Teniendo estas consideraciones presentes, se va a seguir la siguiente estrategia para gestionar el *joystick*.

- Cada vez que se acciona cualquier pulsador (o gesto) del *joystick*, esto debe generar en nuestro sistema una interrupción.
- Una vez detectada la interrupción se enviará una señal a un *thread* que estará esperando la llegada de una señal procedente de la *callback* del manejador de interrupciones del *joystick*. Una vez recibida la señal se arrancará un *timer one-shot* con una duración de 50 ms. De esta manera, cada vez que se produzca un flanco (rebote) en el *joystick*, se generará una nueva interrupción y el *timer* se reiniciará de nuevo. Una vez que venza la temporización se puede dar por finalizado el proceso de pulsación y se puede leer el estado de las cinco líneas del *joystick* y determinar qué gesto de este se ha accionado.
- Una vez que se haya leído el *joystick*, incrementa una variable por cada gesto pulsado, de forma que se pueda llevar la cuenta del número de gestos accionados en el *joystick*. Asimismo, represente en los leds de la tarjeta la tecla pulsada codificada de la siguiente forma.

Gesto	Valor
Arriba	1
Derecha	2
Abajo	3
Izquierda	4
Centro	5

Para la representación del código del gesto seleccionado en los LEDs debe considerar que LD1 es el LSB.

Conecte el *joystick* de la tarjeta de aplicaciones (*mbed app board*) a la tarjeta NUCLEO-144 STM32F429ZI siguiendo la siguiente tabla:

Gesto <i>joystick</i>	Puerto/Pin tarjeta Núcleo	Pin <i>mbed app board</i>
Arriba	Puerto B / Pin 10	15
Derecha	Puerto B / Pin 11	16
Abajo	Puerto E / Pin 12	12
Izquierda	Puerto E / Pin 14	13

Centro	Puerto E / Pin 15	14
--------	-------------------	----

Implemente un *driver* para la gestión del joystick, El driver estará formado por un módulo software que se denominará Thjoy.c que implementará la estrategia antes indicada. El módulo estará formado por un *thread*, un *timer* virtual, así como las señales necesarias para la correcta sincronización.

### **Ejercicio 3**

Partiendo del ejercicio anterior, realice un nuevo proyecto.

Este proyecto encolará las pulsaciones del *joystick*, una vez eliminados los rebotes, en una cola de mensajes de forma que el valor pueda ser leído posteriormente por otro módulo que se suscriba a dicha cola. La información a enviar a la cola será el valor del *joystick* leído como un conjunto de 5 bits.

Deberá añadir al proyecto anterior la cola de mensajes, así como la definición del mensaje de la misma.

Para probar la nueva funcionalidad añada un *thread* que lea dicha cola y represente en el LCD el gesto pulsado.

### **Ejercicio 4 (Opcional)**

Utilizando la tarjeta NUCLEO-144 STM32F429ZI y los recursos necesarios de CMSIS-RTOS2 desarrolle un proyecto que implemente un cronógrafo cuya funcionalidad se describe a continuación:

1. Tendrá cuenta descendente con minutos y segundos
2. El cronógrafo descontará por defecto desde 2 minutos (02:00)
3. Durante el primer minuto de la cuenta descendente, se hará parpadear el LED1 durante un tiempo de 2 segundos (a una frecuencia de 4Hz) cada vez que el número presente en los dígitos de los segundos sea múltiplo de 4.
4. Cuando llegue a cero se hará un barrido de los LEDs (LED1->LED2->LED3) con un tiempo de activación entre LEDs consecutivos de 250 ms. Cuando se termine el barrido, se cambiará el sentido del barrido (LED3->LED2->LED1).
5. El sistema permanecerá en este bucle de barridos indefinidamente.

Utilizando los recursos del sistema operativo para gestionar el control del tiempo y las funciones de representación en el LCD, se debe mostrar la siguiente información en la zona inferior del LCD:

Crono-> mm:ss

en donde **mm** son los minutos y **ss** los segundos de tiempo restante hasta 00:00, incluyendo los ceros no significativos.