GUÍA DE EJERCICIOS N° 3
PUERTOS I/O E INTERRUPCIONES

Objetivo: Familiarizarse con los registros de los periféricos de manejo de puertos I/O y Systick. Familiarizarse con el uso de interrupciones.

1. Aprendiendo a leer el Reference Manual del K64
   Contestar las siguientes preguntas, indicando en cuál sección se encuentra la respuesta:
   - ¿Cuál es la dirección absoluta donde se encuentra el registro PCR del pin PTA12?
   - ¿Cuál bit del registro PCR corresponde al *Interrupt Status Flag*?
   - Luego del *reset* los pines del puerto B: ¿cómo tienen configurada su dirección (entrada o salida)? ¿y cómo tienen configurado el slew-rate (activado o no)? ¿y cómo tienen configurado el *pull* (desactivado, activado *pullup* o activado *pulldown*)?

2. Aprendiendo a leer el SDK
   - Analizar la estructura `Port_Type`. ¿En cuál archivo se encuentra declarado? ¿Por qué tiene campos reservados?

3. MyBlink
   - Basado en `Blink`, crear un proyecto nuevo llamado `MyBlink`.
   - Quitar el archivo `gpio.o` y escribir su propio archivo `gpio.c`, de tal forma que se implementen las funciones (servicios) de `gpio.h`.
   - Verificar el correcto funcionamiento de `gpio.c` en los proyectos `Pul2Switch` y `Baliza`.

4. SysTick
   - Escribir su propio archivo `SysTick.c`, de tal forma que se implementen las funciones (servicios) de `SysTick.h`.
   - Modificar el proyecto `MyBlink` para que el retardo se implemente con el SysTick en lugar de un ciclo. Analizar cómo hacer para que el retardo no sea bloqueante. Verificar el correcto funcionamiento con un osciloscopio.
   - Agregar un pin de testeo (*tespoint* o TP) que se encienda mientras se ejecuta la interrupción. Medir el tiempo que dura la ISR y cuánto representa porcentualmente.

5. Baliza_SysTick
   - Basado en los proyectos anteriores, crear el proyecto `Baliza_SysTick` que implemente el programa de la baliza utilizando `SysTick` como base de tiempo. Utilizar el pulsador SW3 y realizar un muestreo periódico de su estado para detectar eventos.

6. Interrupciones de puerto
   - Escribir un nuevo archivo `gpio.c`, de tal forma que se implementen las funciones (servicios) de manejo de interrupciones del nuevo `gpio.h`.
   - Escribir un programa de prueba (*testbench*) que verifique el correcto funcionamiento de la nueva implementación de `gpio`.
   - Agregar un TP que se encienda mientras se ejecuta la interrupción. Medir el tiempo que dura la ISR.

7. Baliza_IRQ
    - Basado en los proyectos anteriores, crear el proyecto `Baliza_IRQ` que implemente el programa de la baliza utilizando el nuevo `gpio` para leer el pulsador. Utilizar el pulsador SW2 y detectar evento de presionado por interrupción.