



### **Práctica 3: Programación en C de Microcontroladores**

Para la realización de estos ejercicios el alumno deberá tener a mano y consultar el siguiente material relacionado con el Kit de desarrollo:

- Guía de inicio rápido del kit FRDM-KL46Z
- Guía del usuario de OPENSDA para FRDM
- Manual del usuario del kit FRDM-KL46Z
- Manual de referencia de la familia de microcontroladores KL46 de Freescale
- Esquemático (circuito) del FRDM-KL46Z

Antes de comenzar a realizar los ejercicios, el alumno debe:

- 1) Ver esquemático del kit e identificar pines y puertos conectados con los leds y pulsadores
- 2) Ver manual de referencia del MC, particularmente:
  - Module Multiplexer (PINOUT - Pin Control Register)
  - System Clock Gating Control
  - General-Purpose Input/Output (GPIO)
- 3) Cargar la aplicación DEBUG-APP\_Pemicro\_vXXX.SDA como aplicación a ejecutar por OpenSDA.
- 4) Correr Eclipse/Codewarrior y familiarizarse con el entorno

1) Hacer parpadear permanentemente el led rojo del kit.

Ensayar dos métodos: uno utilizando GPIOx\_PSOR/GPIOx\_PCOR y otro usando GPIOx\_PTOR.

El tiempo de encendido/apagado deberá estar determinado por una función `void retardo (int veces)` a la cual llamará el programa principal según sea necesario.

Este ejercicio debe realizarse mediante asignación directa de valores numéricos a los registros y luego utilizando las macros disponibles.

2) Hacer parpadear en forma alternada los leds verde y rojo del kit.

3) Hacer parpadear en forma alternada los leds verde y rojo del kit sólo mientras el pulsador SW1 permanezca pulsado. En caso contrario deben permanecer apagados.

4) Crear sendas funciones C que hagan las operaciones lógicas NOT, AND, OR y XOR tomando como entradas/argumentos SW1 y SW2. Probarlas con algún código atacando los leds con su producido en el `main()`. (Por ejemplo, codificando las expresiones lógicas: `ledRojo=(SW1 or (not SW2))` , `ledVerde=SW1 and SW2`, etc.)

5) Modificar el ejercicio 3 para que sólo se produzca el parpadeo cuando ambos pulsadores estén accionados (SW1 y SW3)

6) En los ejercicios anteriores se pierde mucho tiempo durante los retardos. Implementarlos con un esquema que permita liberar el CPU para otras tareas mientras el retardo transcurre.

**Para los ejercicios que siguen, partir de la realización previa de la o las Maquinas de Estado (MEF) necesarias para representar y resolver cada problema planteado.**

7) Hacer un programa en el cual un pulsado de SW1 encienda el led rojo (quedando encendido al soltar SW1) y un pulsado de SW3 lo apague.

Luego, verifique qué hace su MEF y programa con ambos pulsadores accionados simultáneamente.

**8)** Hacer un programa similar al anterior en el cual el led se encienda y apague utilizando sólo un pulsador.

Analice: ¿por qué en este kit no se producen efectos de “rebotes” en los pulsadores?

**9)** Modificar el programa anterior para hacerlo apto para cualquier circunstancia en la que existieran rebotes.

**10)** Hacer la/las MEF para el ejercicio 3 y considerando el ejercicio 6). Luego, hacer el programa para dicha MEF y verificar el funcionamiento.

**11)** Hacer un programa que, arrancando con los leds apagados, pasen a parpadear alternativamente al pulsar SW1 y vayan aumentando la velocidad de parpadeo con cada pulsada de SW1 y disminuyendo la velocidad con cada pulsada de SW2. Defina la magnitud de los incrementos/decrementos a gusto.

**12)** Realizar el problema 11 pero aumentando la velocidad mientras se mantenga pulsado SW1 y disminuyéndola mientras se mantenga pulsado SW2.

**13)** Realizar un programa que memorice una secuencia de 10 pulsadas, de SW1 y/o SW2, y la repita simulada con los leds verde y rojo al terminar el ingreso.

**14)** Realice el programa anterior sin utilizar arreglos (implemente una FIFO con punteros) y sin límite de pulsadas (sólo limitada por la memoria del microcontrolador). La repetición debe comenzar luego de 5 segundos (aproximadamente) de no accionar ningún pulsador.

**15)** Realizar un programa que memorice una secuencia de tiempos entre pulsadas de SW1 y, al pulsar SW2 repita dicha secuencia temporal usando el led rojo.

**16)** Realice el ejercicio anterior pero ahora recordando y reproduciendo también el tiempo en que el pulsador permaneció accionado.

**17)** Realizar un programa tal que se vaya generando al azar una secuencia de encendido de los leds verde y rojo, permitiendo al usuario la repetición de la secuencia mediante SW1 y SW3. Cada vez que el usuario repite la secuencia correctamente, el programa debe mostrar la secuencia con los leds y agregar un paso más. El juego termina cuando el usuario se equivoca, evento que debe mostrarse haciendo parpadear en forma distintiva el led rojo.