SERIE DE EJERCICIOS 1
PARA PRACTICAR LA PROGRAMACION EN LENGUAJE ENSAMBLADOR

NOTA: es importante que el alumno tome conciencia de que la única manera de aprender a programar es practicando. Estos ejercicios promoverán considerablemente la capacidad de programación del alumno y por ende, sus conocimientos finales sobre el temario y los objetivos del curso.

Los ejercicios deben desarrollarse y posteriormente cargarse y probarse en el sistema Bolt 18F2550 ó algún sistema similar. Todos los ejercicios que lleven una secuencia de activación de los LEDS, tienen un retraso aproximado de 200 ms entre una activación y la siguiente.

1. Entradas: leer microswitches SW4..SW1. Salidas: Escribir el estado de los microswitches en B1...B4. Loop continuo.

2. Contador binario. Encender los leds B0...B7, mostrando los códigos del 0 al 255 en binario. Los leds encendidos deberán de apagarse durante 200 ms antes de mostrar el siguiente incremento.

3. Contador binario controlado por microswitch SW1. Realizar una cuenta binaria en los leds. Si SW1=0, la cuenta continúa. Si SW1=1, la cuenta se suspende.

4. Leer los microswitches. Si SW1=1, Encender los leds comenzando por B0, B2..y así sucesivamente hasta B6...Si SW1 =0, encender B1, luego B3 y así sucesivamente hasta B7. Repetir la rutina.

5. Leer los microswitches, SW4...SW1. clave de 4 bits (n= 0...15). Si n =2, activar el relevador, si n =7, desactivarlo. Repetir la rutina.

6. Ejercicio PING-PONG: si SW3=OFF, enciende led B0 con corrimientos a la derecha hasta llegar a B7. Si SW3=ON, enciende led B7 con corrimientos a la izquierda hasta llegar a B0.

7. Realice un programa en lenguaje ensamblador que haga la siguiente función:
   - Lee de los microswitches SW4..SW1, un número n con valores entre 0…F (en hexadecimal).
   - Si el valor es n=6, activa el relevador durante 1 segundo.
   - Convierte este número a su valor ASCII, el cual deberá mostrar en los leds.
   - Ejemplo: si SW4..SW1 es igual a 1101, en los leds LED7..LED0 deberá mostrarse:

     0100 0100 es decir, el número 0DH codificado en ASCII
   - Fin del programa. Para repetir el procedimiento, dar reset.

8. Entradas: clave de 3 bits (n= 0...7) en los microswitches. Salidas: LED B(n) encendido, los demás apagados. Fin del programa.

9. Entradas: clave de 3 bits (n= 0...7) en los microswitches. Salidas: LEDs B0 a B(n) encendidos los demás apagados.
10. Lee 3 microswitches (SW3...SW1), obtiene un valor de n=0...7. Enciende secuencialmente uno a uno, con retrasos de 200 ms. (call retro) <u>todos los leds</u> desde B0 hasta Bn, con los demás apagados. Al final, el programa entra en modo “sleep”.

11. Entradas: clave de 3 bits en los microswitches (n= 0...7). Salidas: realizar corrimientos con un led encendido de B0 a B7. Modificar la velocidad del corrimiento de acuerdo a la clave n, siendo n=0 la velocidad más lenta y n=7 la velocidad más rápida.

12. Después de reset, lee los 4 microswitches (SW4...SW1), y obtiene un valor de n=1...15 (en decimal). Si ***n es par***, realiza en los leds una cuenta binaria ascendente de 0 hasta n (con retrasos de 500 ms). Si ***n es impar***, realiza una cuenta descendente desde n hasta 0 (retrasos de 500 ms). Al final de la cuenta se ejecuta la instrucción ***sleep***. Para repetir el procedimiento, debe oprimirse el botón de reset.

13. Entradas: clave de 3 bits (n= 0...7) en los microswitches. Salidas: empezando con el LED B0, encendido, los demás apagados, hacer corrimientos hacia la izquierda. Al llegar al LED B(n), encendido, activar el relevador. Continuar con los corrimientos (al llegar a B7, seguir con B0). A la siguiente vuelta, desactivarlo y así sucesivamente. Retraso de 200 ms.

14. Generar retrasos variables, aproximados de 1 a 15 segundos dependiendo de la clave en los microswitches SW4..SW1 (n=0..15). Activar el relevador y desactivarlo al final del retraso. Flashear el led B0 cada segundo.

15. Mismo problema anterior, pero flashear RB7 cada segundo y escribir en los leds B3...B0 la cuenta descendente del valor leído en los microswitches al inicio del programa.

16. Después de reset, lee los 4 microswitches y forma un par de números: SW4 y SW3 (n=0..3). SW2 y SW1 (m=0..3). Muestra en los leds el valor de la multiplicación de (n*m) en binario. Si el usuario modifica la posición de los microswitches, el programa automáticamente modificará el resultado del valor de la multiplicación en los leds.

17. Programa que maneja 5 funciones según el valor en los microswitches. Al cambiar la clave, se modifica la función. Elija usted las claves en SW3…SW1 y asigne 5 funciones distintas: por ejemplo, activar y desactivar el relevador, cuenta binaria en los leds, corrimiento a la derecha ó izquierda en los leds. Al modificar la clave, debe darse reset al micro para cambiar la función.

18. A través de los microswitches SW4…SW1, leer 2 números A y B del 0 al 15. Mostrar en los LEDS la multiplicación de A x B, en binario, siguiendo la siguiente secuencia: al inicio del programa, muestra el número A en los leds. Retraso de 2 segundos (durante este retraso, el usuario modifica el valor de los microswitches). Lee el número B y lo muestra en los leds durante 2 segundos. Finalmente muestra el resultado A x B en los leds.

19. Manejo de tablas: utilizando tablas de datos almacenadas en la memoria de código (instrucción ***retlw***), desarrolle un convertidor de código binario (0…0xF) a código ASCII. El código binario será leído de los microswitches SW4…SW1. En los leds aparecerá el código ASCII correspondiente. Cada vez que se modifique el valor en los microswitches, se actualiza automáticamente el valor en los leds.

20. Realice un programa que almacene en localidades consecutivas de la **memoria de código** del microcontrolador una tabla con los siguientes 8 datos hexadecimales. Utilice la instrucción ***retlw***:

54 01 2E 1A 57 6A 70 55

Muestre en los leds, LED7..LED0 en código binario, en forma consecutiva, cada uno de estos valores durante 1 segundo. Finalmente, el programa deberá calcular y mostrar el valor mayor en los leds y ejecutar la instrucción ***sleep.***

21. Realice un programa que almacene en localidades de la **memoria de código** del microcontrolador ***una tabla*** mediante las instrucciones ***retlw***, con los siguientes 8 datos hexadecimales:

AA 20 3C 5E D9 16 0B 9F

Muestre en los leds, LED7..LED0 en código binario, en forma consecutiva, cada uno de estos valores durante 1 segundo. Si los 4 bits menos significativos del dato mostrado en los leds es igual al valor de los microswitches SW4..SW1 (n=0…F), los leds deberán permanecer mostrando ese mismo dato. Programa termina con instrucción sleep.

Si ninguno de los datos coincide con el valor de los microswitches, el programa mostrará los 8 datos de la tabla y termina.

22. Realice un programa que almacene en localidades consecutivas de la **memoria de datos** del microcontrolador una tabla con los siguientes 8 datos hexadecimales. Utilice direccionamiento indirecto con el ***registro FSR0:***

54 39 01 2E 1A 57 6A 70

Muestre en los leds, LED7..LED0 en código binario, en forma consecutiva, cada uno de estos valores durante 1 segundo. Al final, el programa deberá encontrar y mostrar el valor menor en los leds antes de ejecutar la instrucción ***sleep.***

23. Realice un programa que almacene en localidades de la **memoria de código** del microcontrolador ***una tabla*** mediante las instrucciones ***retlw***, con los siguientes 8 datos hexadecimales:

0A 20 0C 31 09 16 0B 1F

Muestre en los leds, LED7..LED0 en código binario, en forma consecutiva, cada uno de estos valores durante 1 segundo. Al final, el programa deberá calcular y mostrar el valor de la suma de los 8 valores en los leds, en binario. Fin del programa.